



Aula 03

Prof: Henrique Augusto Maltauro

Desenvolvendo Algoritmos

Validação de Algoritmos

Validação de Algoritmo

Uma vez construído um algoritmo, nós precisamos validar a sua lógica, e até mesmo se surgir uma falha no algoritmo, nós precisamos encontrar uma maneira de identificar essa falha.

E para isso nós utilizamos o teste de mesa.

Validação de Algoritmo

O teste de mesa é um processo manual que é utilizado para validar a lógica de um determinado algoritmo.

Ele é utilizado principalmente em algoritmos quando a linguagem utilizada não possui nenhuma ferramenta automatizada de depuração.

Validação de Algoritmo

Como as linguagens de programação costumam possuir tais ferramentas, é mais comum utilizá-las do que fazer o teste de mesa propriamente dito, embora exista linguagens de programação que não possuem tais ferramentas, ou a estrutura da aplicação dificulta o seu uso.

Validação de Algoritmo

Não há uma forma definida para a elaboração de um teste de mesa, pois dependerá muito do que pretende verificar no algoritmo e do seu nível de entendimento.

No geral, você deverá criar no papel uma tabela com todas as variáveis do programa e executar passo a passo seu código, anotando sempre os valores das variáveis.

Validação de Algoritmo

Assim você será capaz de identificar se os valores condizem com o esperado ou localizar a exata linha de código onde o valor da variável passa a ficar errado.

Contudo, nós podemos definir uma série de passos para nos auxiliar na execução do teste de mesa.

Validação de Algoritmo

1. Elaborar uma tabela com as colunas de **passo**, **linha**, **condição**, **entrada**, **saída** e uma coluna para **cada variável** envolvida
2. Executar os passos previstos no algoritmo
3. Verificar se os resultados obtidos são coerentes com os previstos
4. Encerrar o teste após um número razoável de resultados corretos obtidos


```
1  ALGORITMO
2  VAR      NUMERO      NUMÉRICO      =      0
3  VAR      RESULTADO   NUMÉRICO      =      1
4  LEIA(NUMERO)
5  SE      (NUMERO      <      0)      ENTÃO
6  ESCREVA("O  NÚMERO  NÃO  PODE  SER  NEGATIVO")
7  SENÃO
8  ENQUANTO      (NUMERO      >      0)      FAÇA
9  RESULTADO      =      RESULTADO      *      NUMERO
10 NUMERO      =      NUMERO      -      1
11 FIM_ENQUANTO
12 ESCREVA("O  FATORIAL  VALE", RESULTADO)
13 FIM_SE
14 FIM_ALGORITMO
```

Validação de Algoritmo

Passo	Linha	Condição	Entrada	Saída	NUMERO	RESULTADO
Início	1	-	-	-	-	-
1	2	-	-	-	0	-
2	3	-	-	-	0	1

Validação de Algoritmo

Passo	Linha	Condição	Entrada	Saída	NUMERO	RESULTADO
Início	1	-	-	-	-	-
1	2	-	-	-	0	-
2	3	-	-	-	0	1
3	4	-	3	-	3	1
4	5	$3 < 0$ (false)	-	-	3	1
5	8	$3 > 0$ (true)	-	-	3	1
6	9	-	-	-	3	3

Validação de Algoritmo

Passo	Linha	Condição	Entrada	Saída	NUMERO	RESULTADO
7	10	-	-	-	2	3
8	8	$2 > 0$ (true)	-	-	2	3
9	9	-	-	-	2	6
10	10	-	-	-	1	6
11	8	$1 > 0$ (true)	-	-	1	6
12	9	-	-	-	1	6
13	10	-	-	-	0	6

Validação de Algoritmo

Passo	Linha	Condição	Entrada	Saída	NUMERO	RESULTADO
14	8	$0 > 0$ (false)	-	-	0	6
15	12	-	-	0 FATORIAL VALE 6	0	6
FIM	14	-	-	-	-	-

Exercício

Exercício

Utilizando a estrutura de tabela abaixo, faça o teste de mesa dos algoritmos a seguir.

Passo	Linha	Condição	Entrada	Saída	VARIÁVEL

```
1  ALGORITMO
2  VAR          N1          NUMÉRICO
3  ESCREVA( "INFORME      UM      NÚMERO:      " )
4  LEIA(N1)
5  SE      (N1      MOD      2      =      1)      ENTÃO
6  ESCREVA( "ESSE      NÚMERO      É      IMPAR" )
7
8  ESCREVA( "ESSE      NÚMERO      É      PAR" )
9
10 FIM_ALGORITMO      FIM_SE
```



```
1  ALGORITMO
2  VAR REAJUSTE NUMÉRICO = "17.5"
3  VAR SALARIO NUMÉRICO
4  VAR SALARIO_REAJUSTADO NUMÉRICO
5  ESCREVA("INFORME O SEU SALÁRIO: ")
6  LEIA(SALARIO)
7  SALARIO_REAJUSTADO = SALARIO * (100 / REAJUSTE)
8  ESCREVA("SEU NOVO SALÁRIO: ", SALARIO_REAJUSTADO)
9  FIM_ALGORITMO
```

```
1  ALGORITMO
2  VAR N1, N2, N3, N4, SOMA, MEDIA NUMÉRICO
3  LEIA(N1)
4  LEIA(N2)
5  LEIA(N3)
6  LEIA(N4)
7  SOMA = N1 + N2 + N3 + N4
8  MEDIA = SOMA / 4
9  SE (MEDIA >= 7) ENTÃO
10  ESCREVA("Aluno Aprovado")
11  SENÃO
12  ESCREVA("Aluno Reprovado")
13  FIM_SE
14  FIM_ALGORITMO
```