

# Aula 3 – Oficina de Programação

## Algoritmos - Parte 2

Profa. Elaine Faria

UFU - 2017

# Estruturas de Seleção

- Uma estrutura de seleção permite a escolha de um grupo de ações (bloco) a ser executado quando determinadas condições são ou não satisfeitas.
- Tipos de Estruturas de Seleção
  - Seleção Simples
  - Seleção Composta
  - Seleção Encadeada
  - Múltipla escolha

# Seleção simples

- Seleção simples: se...entao...fimse

**se** (<condição>) **entao**

instrução 1

instrução 2

instrução 3

...

instrução n

**fimse**

# se...entao...fimse - Exemplo

- Faça um algoritmo que calcule a média aritmética entre quatro notas bimestrais fornecidas por um aluno. O aluno será aprovado se a média for maior ou igual a 7.

**var**

n1, n2, n3, n4, MA: real

**inicio**

**leia**(n1, n2, n3, n4)

MA ← (n1+n2+n3+n4)/4

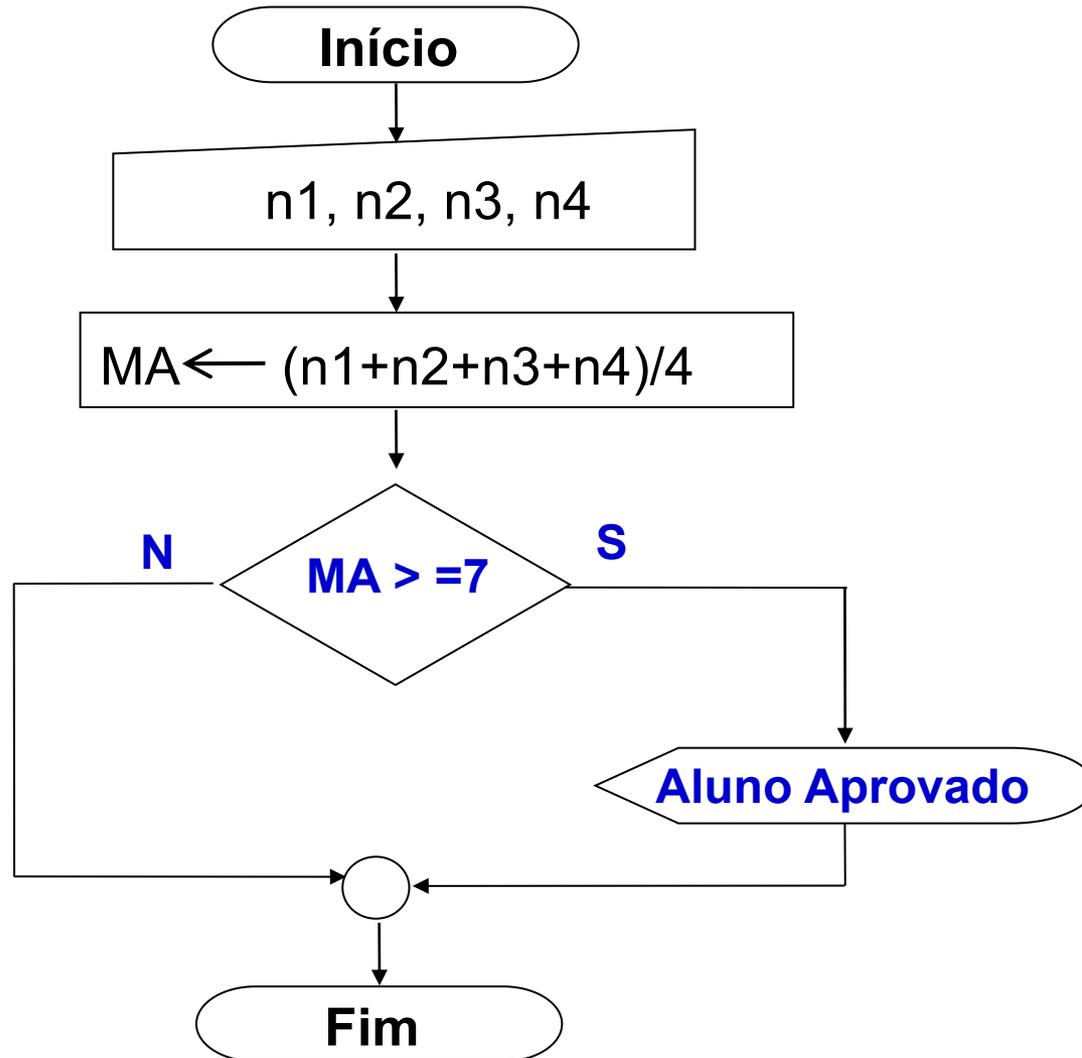
**se** (MA >= 7) **entao**

**escreva**("Aluno Aprovado")

**fimse**

**fimalgoritmo**

# Fluxograma



# Operadores relacionais

- Usados para comparar dois valores de mesmo tipo

<b>Operador</b>	<b>Função</b>
=	Igual a
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a
<>	Diferente de

- O resultado é sempre um valor lógico (Verdadeiro ou Falso)

- Faça um algoritmo que leia dois valores numéricos, efetue a multiplicação entre eles e apresente o resultado na tela apenas se o valor for diferente de 10.”

```
var  
    num1, num2, result: inteiro  
inicio  
    escreva("Digite dois números: ")  
    leia(num1, num2)  
    result ← num1 * num2  
  
    se (result <> 10) entao  
        escreva("O valor é: ", result)  
  
    fimse  
fimalgoritmo
```

# Seleção Composta

- Usado quando tivermos situações em que duas alternativas dependem de uma mesma condição, onde uma condição é verdadeira e a outra condição é falsa.

```
se (<condição>) entao
```

```
    C1 // instrução executada se a condição for  
        verdadeira
```

```
senao
```

```
    C2 // instrução executada se a condição for falsa
```

```
fimse
```

# se...entao...senao...fimse

**var**

n1, n2, n3, n4, MA: real

**inicio**

**leia**(n1, n2, n3, n4)

MA  $\leftarrow$  (n1+n2+n3+n4)/4

**se** (MA  $\geq$  7) **entao**

**escreva**("Aluno Aprovado")

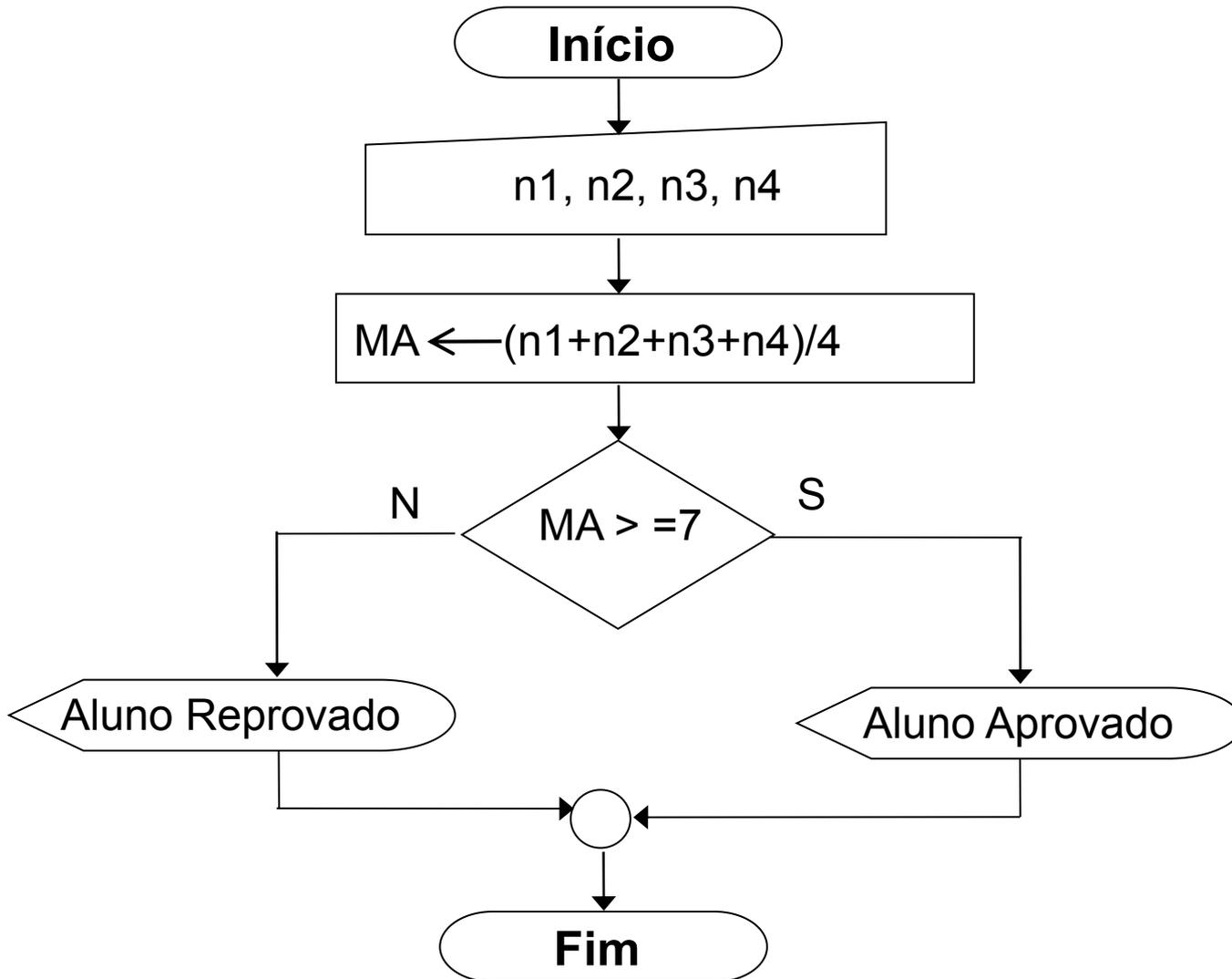
**senao**

**escreva**("Aluno Reprovado")

**fimse**

**fimalgoritmo**

# Fluxograma



- Faça um algoritmo que leia dois valores numéricos e efetue a soma.
  - Caso o valor somado seja maior ou igual a 10, some 5 ao resultado e apresente-o na tela.
  - Caso o valor somado seja menor que 10, subtraia 7 do resultado e imprima-o na tela.”

**algoritmo** “Calcula”

**var**

num1, num2, soma, result: real

**inicio**

**leia**(num1, num2)

soma ← num1 + num2

**se** (soma ≥ 10) **entao**

result ← soma + 5

**senao**

result ← soma - 7

**fimse**

**escreva**(“Resultado: ”, result)

**fimalgoritmo**

# Seleção Encadeada

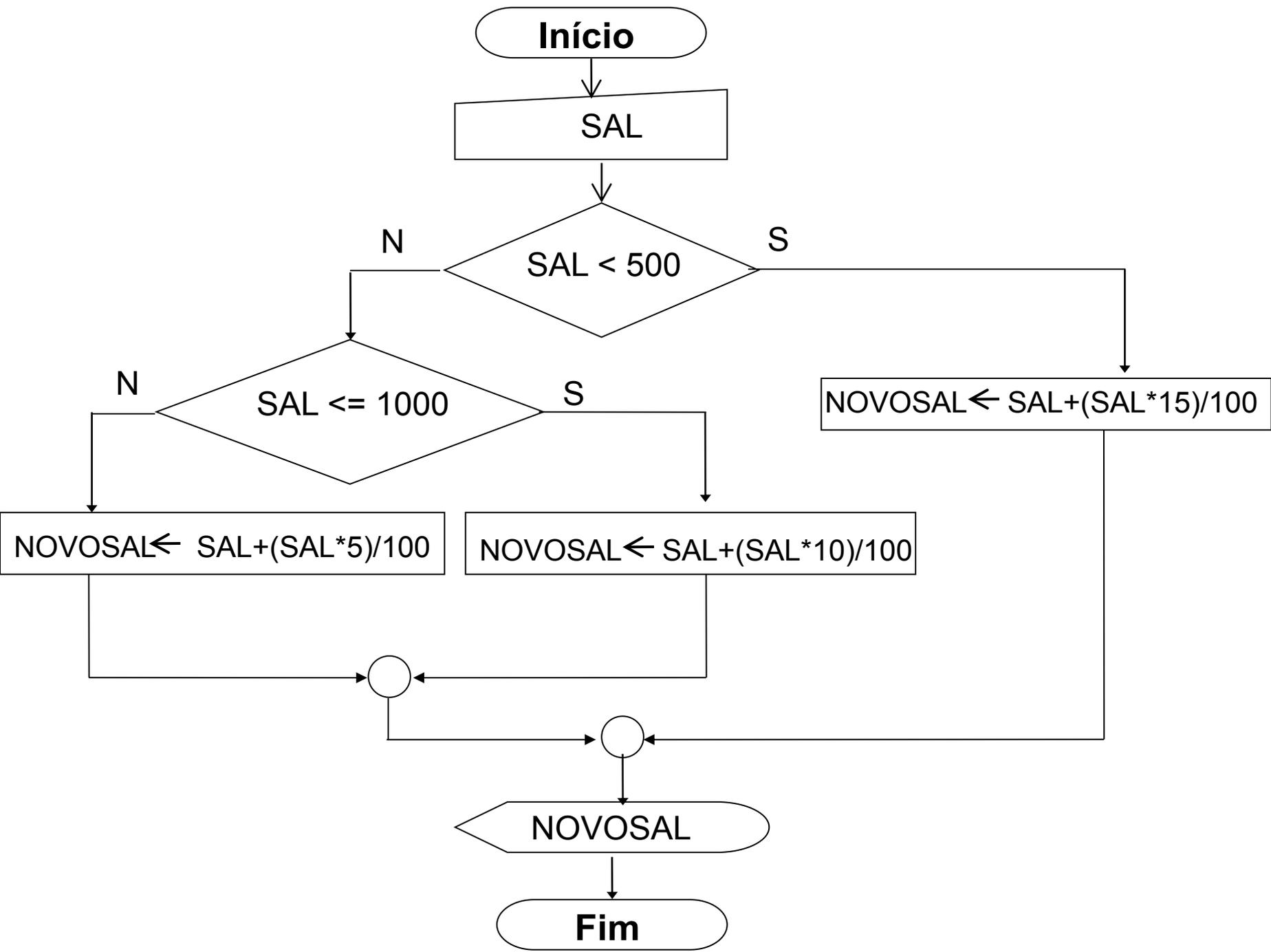
- É o agrupamento de várias seleções

```
se (<condição 1>) entao  
    se (<condição 2>) entao  
        Comandos  
    senao  
        se (<condição 3>) entao  
            Comandos  
        fimse  
    fimse  
fimse
```

- Faça um algoritmo efetue o cálculo do reajuste de salário de um funcionário. Considere que o funcionário deverá receber um reajuste de 15% caso seu salário seja menor que 500. Se o salário for maior ou igual a 500, mas menor ou igual a 1000, seu reajuste será de 10%; caso seja ainda maior que 1000 o reajuste deverá ser de 5%”

O problema estabelece três condições para calcular o reajuste do salário:

- 1) Salário < 500, reajuste de 15%
- 2) Salário > = 500, mas <= 1000, reajuste de 10%
- 3) Salário > 1000, reajuste de 5%



**algoritmo** “salario”

**var**

salario, novoSalario: real

**inicio**

leia(salario)

**se** (salario < 500) **entao**

novoSalario ← salario+(salario\*15)/100

**senao**

**se** (salario <=1000) **entao**

novoSalario ← salario+(salario\*10)/100

**senao**

novoSalario ← salario+(salario\*5)/100

**fimse**

**fimse**

escreva(novoSalario)

**fimalgoritmo**

**algoritmo** “EstadoCivil”

**var**

ESTADOCIVIL: inteiro

**inicio**

**escreva**(“Escolha: 1(casado),2(Solteiro),3(Divorciado) : ”)

**leia**(ESTADOCIVIL)

**se** (ESTADOCIVIL = 1) **entao**

**escreva**(“Indivíduo casado”)

**senao**

**se**(ESTADOCIVIL = 2) **entao**

**escreva**(“Indivíduo solteiro”)

**senao**

**se**(ESTADOCIVIL = 3) **entao**

**escreva**(“Indivíduo divorciado”)

**senao**

**escreva**(“Estado civil inválido”)

**fimse**

**fimse**

**fimse**

**fimalgoritmo**

# escolha .... caso ... fimescolha

- Em algumas situações é necessário termos várias soluções ligadas a respostas diferentes
- O comando de seleção simples ou composta não é uma solução prática, porque obrigará o programador a escrever muitas linhas de programa, além de ter que criar vários comandos de alternativas compostas
- Solução: usar o comando de alternativa de múltipla escolha (escolha ... caso)

var

**ESTADOCIVIL: inteiro**

inicio

**escreva("Escolha: 1(casado),2(Solteiro),3(Divorciado): ")**  
**leia(ESTADOCIVIL)**

**escolha (ESTADOCIVIL)**

**caso 1**

**escreva("Individuo casado!")**

**caso 2**

**escreva("Individuo solteiro!")**

**caso 3**

**escreva("Indivíduo divorciado!")**

**outrocaso**

**escreva("Estado civil inválido")**

**fimescolha**

fimalgoritmo

**escolha .... caso ... fimescolha**

**escolha** (< expressão-de-seleção >)

**caso** < exp 1 > , < exp 2 > , ... , < exp n >  
< lista-de-comandos-1 >

**caso** < exp 1 > , < exp 2 > , ... , < exp n >  
< lista-de-comandos-2 >

**outrocaso**

< lista-de-comandos-3 >

**fimescolha**

# Exemplo

Um determinado clube de futebol pretende classificar seus atletas em categorias e para isto ele contratou um programador para criar um programa que executasse esta tarefa. Para isso o clube criou uma tabela que continha a faixa etária do atleta e sua categoria

<b>IDADE</b>	<b>CATEGORIA</b>
De 05 a 10	Infantil
De 11 a 15	Juvenil
De 16 a 20	Junior
De 21 a 25	Profissional

# Exemplo

```
algoritmo "CLASSIFICAÇÃO DE ATLETAS
```

```
var
```

```
    nome, categoria : caractere
```

```
    idade : inteiro
```

```
inicio
```

```
    escreva("Nome do Atleta: ")
```

```
    leia (nome)
```

```
    escreva("Idade do Atleta: ")
```

```
    leia (idade)
```

**escolha (idade)**

**caso** 5,6,7,8,9,10

categoria ← "Infantil"

**caso** 11,12,13,14,15

categoria ← "Juvenil"

**caso** 16,17,18,19,20

categoria ← "Junior"

**caso** 21,22,23,24,25

categoria ← "Profissional"

**outrocaso**

categoria ← "INVALIDO"

**fimescolha**

**escreva** ("Categoria: ", categoria)

**fimalgoritmo**

# Expressões lógicas

- A condição que segue o 'se' do comando de seleção pode formar uma expressão lógica.
- Uma expressão lógica pode ser formada pelos seguintes operadores lógicos:

<b>Operador</b>	<b>Função</b>
não	negação
e	conjunção
ou	disjunção

# Operador lógico: e

Tabela-verdade

<b>Condição 1</b>	<b>Condição 2</b>	<b>Resultado</b>
Falsa	Falsa	Falso
Falsa	Verdadeira	Falso
Verdadeira	Verdadeira	Verdadeiro
Verdadeira	Falsa	Falso

- Se fizer **sol** e eu tiver roupa de banho, eu vou para a praia. Quando eu vou para a praia?

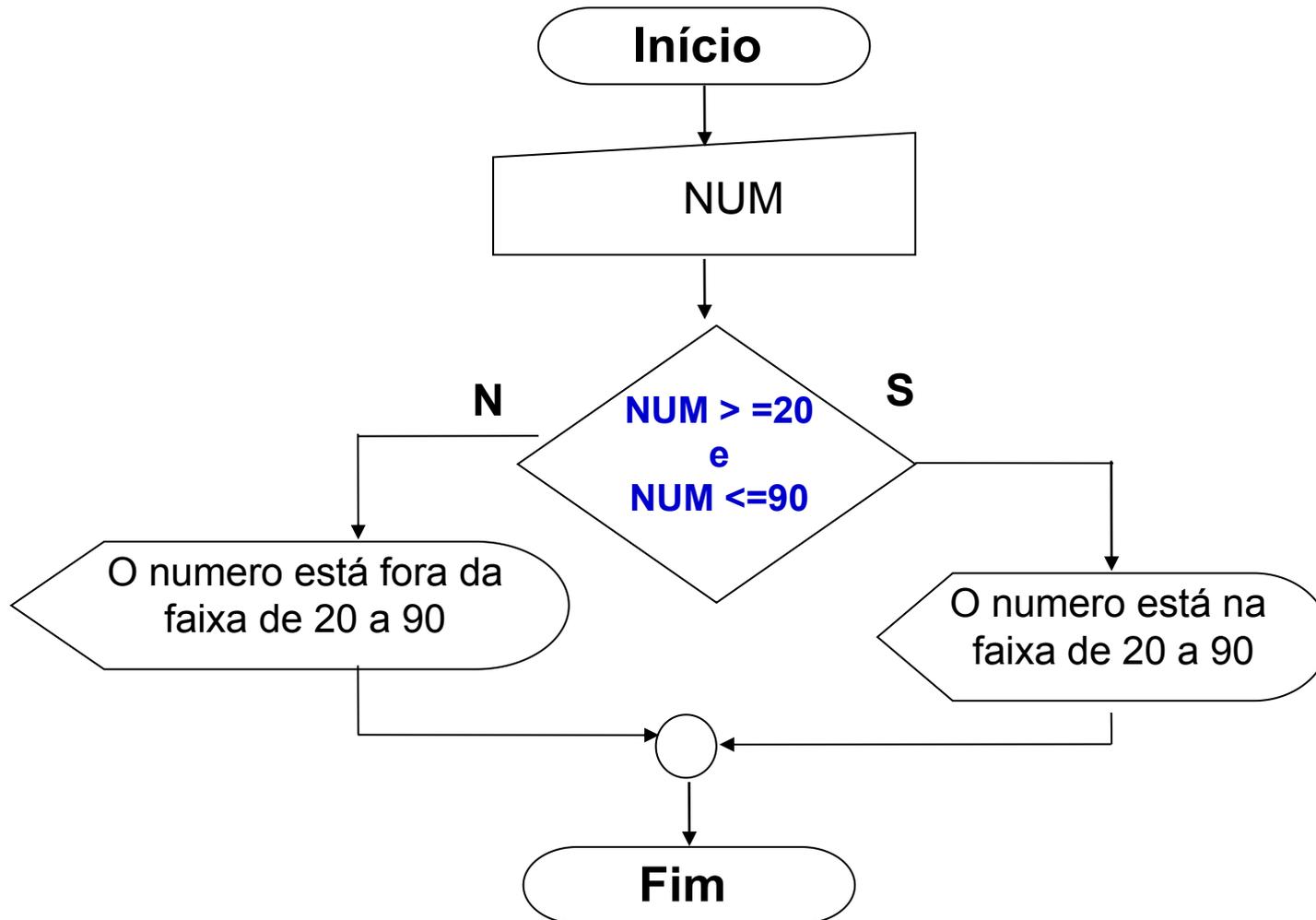
# Operador lógico: e

- Exemplo

```
var
    num: inteiro
inicio
    leia(NUM)

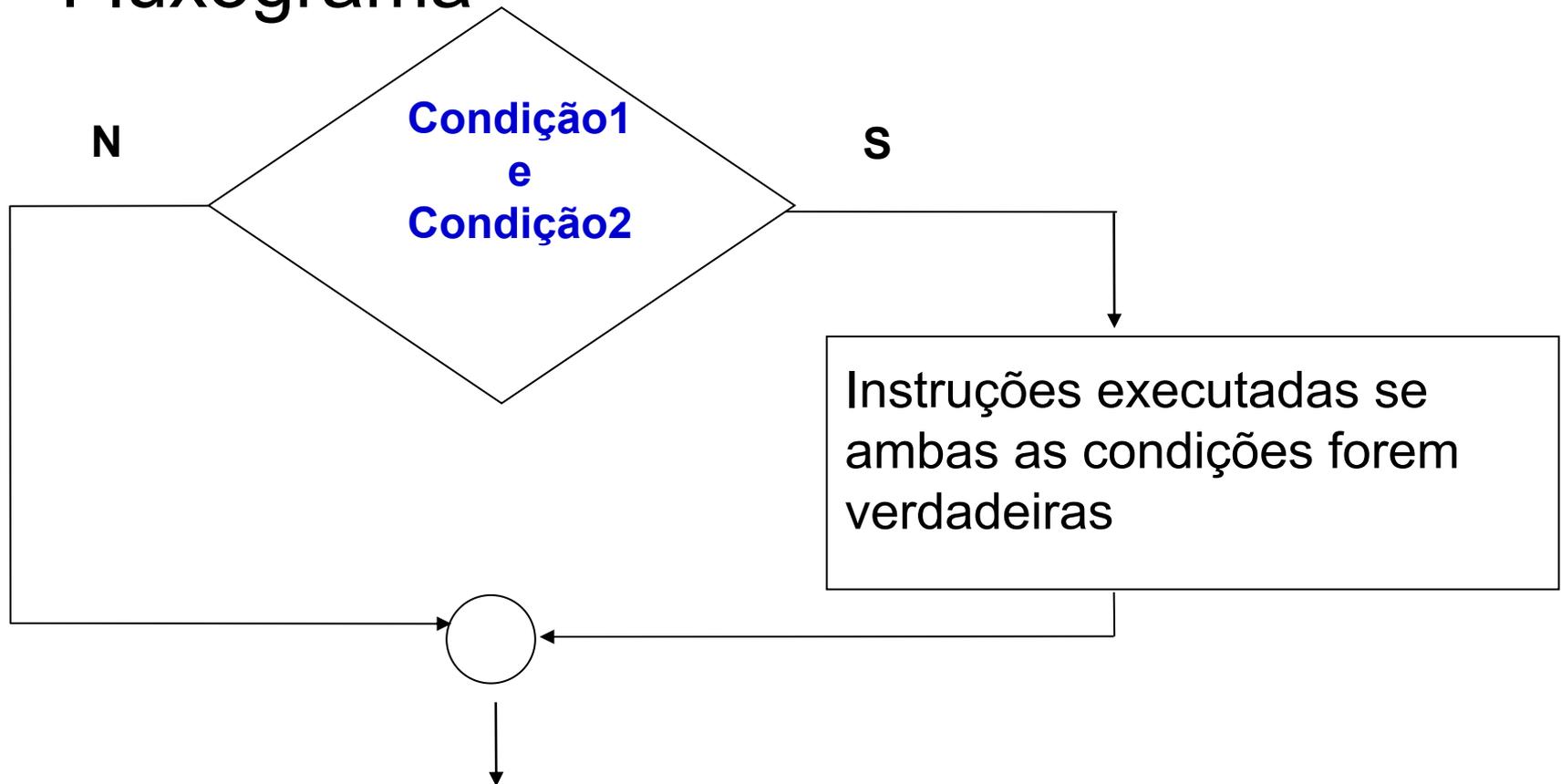
    se (NUM >= 20) e (NUM <=90) entao
        escreva("O número está na faixa de 20 a 90")
    senao
        escreva("O número está fora da faixa de 20 a 90")
    fimse
fimalgoritmo
```

# Fluxograma



# Operador lógico: e

- Fluxograma



# Operador lógico: e

- Português Estruturado

```
se (<condição1>) e (<condição2>) entao
```

```
    C1    // instrução executada se ambas condições  
          forem verdadeiras
```

```
senao
```

```
    C2    // instrução executada se pelo menos uma das  
          condições for falsa
```

```
fimse
```

# Operador lógico: ou

## Tabela-verdade

Condição 1	Condição 2	Resultado
Falsa	Falsa	Falso
Falsa	Verdadeira	Verdadeiro
Verdadeira	Verdadeira	Verdadeiro
Verdadeira	Falsa	Verdadeiro

- Se fizer **sol** **ou** eu tiver roupa de banho, eu vou para a praia.

Quando eu vou para a praia?

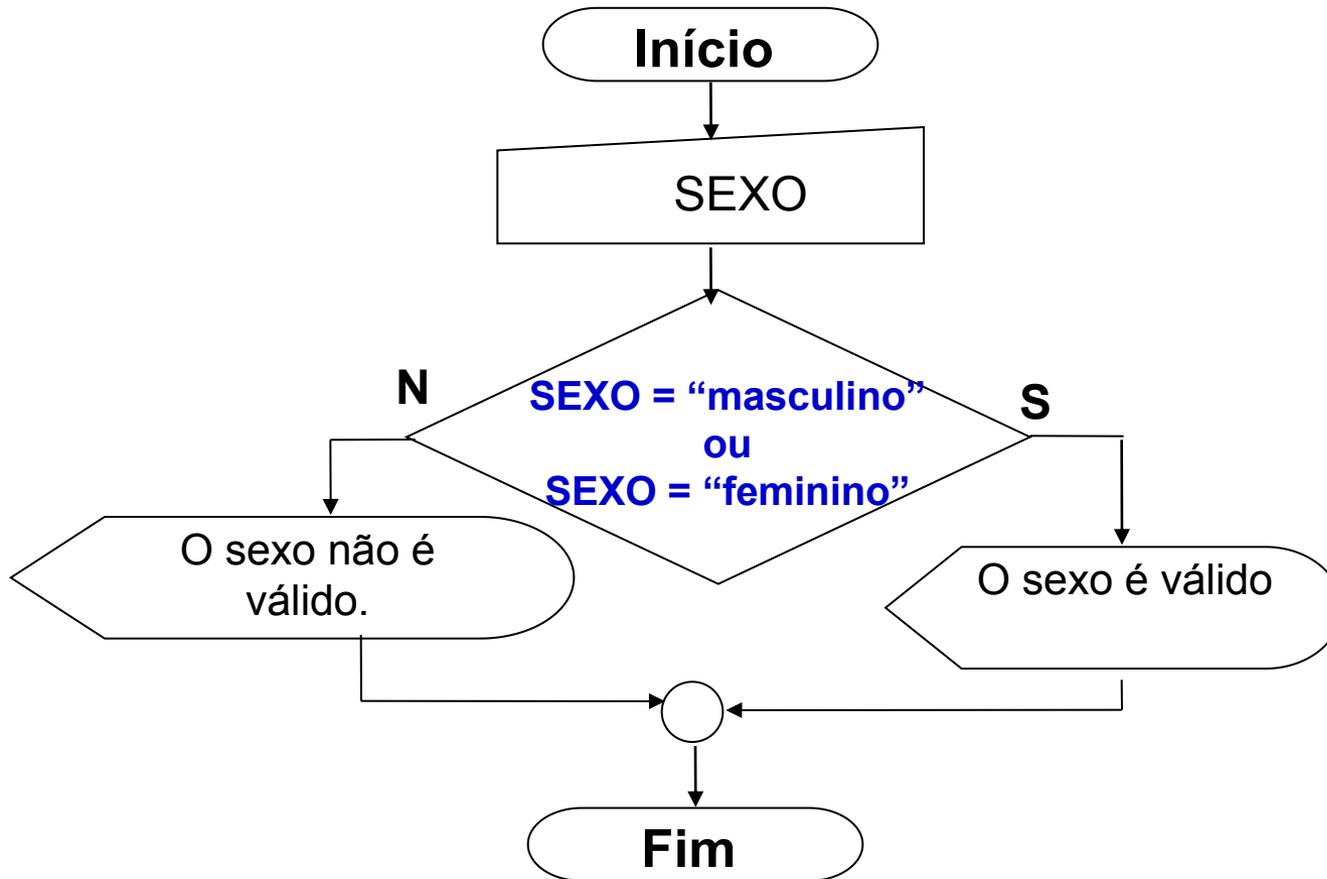
# Operador lógico: ou

- Exemplo

```
var
    sexo: caracter
inicio
    leia(sexo)

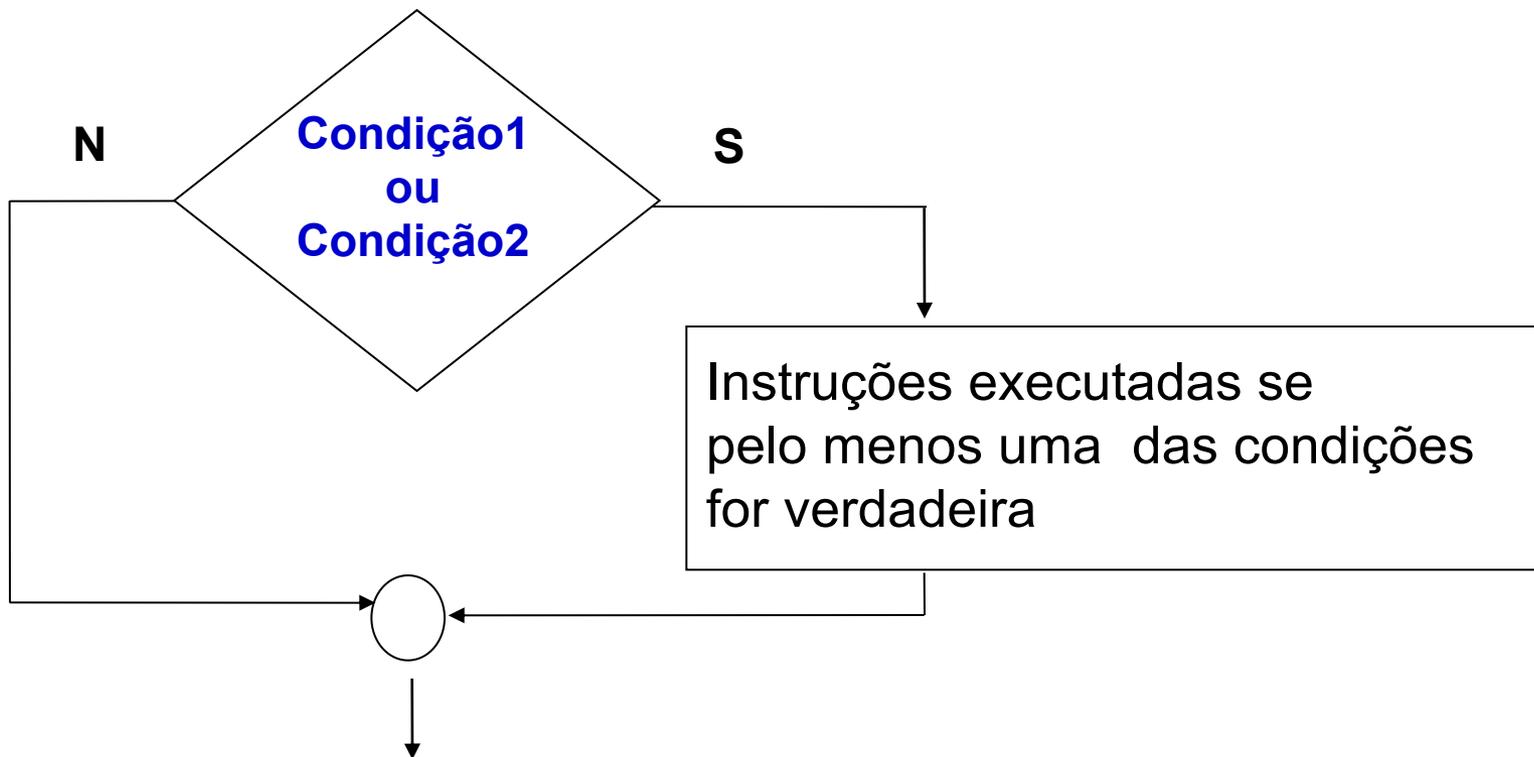
    se (sexo = "masculino") ou (sexo = "feminino") entao
        escreva("O sexo é válido")
    senao
        escreva("O sexo não é válido")
    fimse
fimalgoritmo
```

# Fluxograma



# Operador lógico: ou

- Fluxograma



# Operador lógico: ou

- Português Estruturado

**se** (<condição1>) ou (<condição2>) **então**

C1 // instrução executada se pelo menos uma  
condições for verdadeira

**senao**

C2 // instrução executada se ambas as condições  
forem falsas

**fimse**

# Operador lógico: não

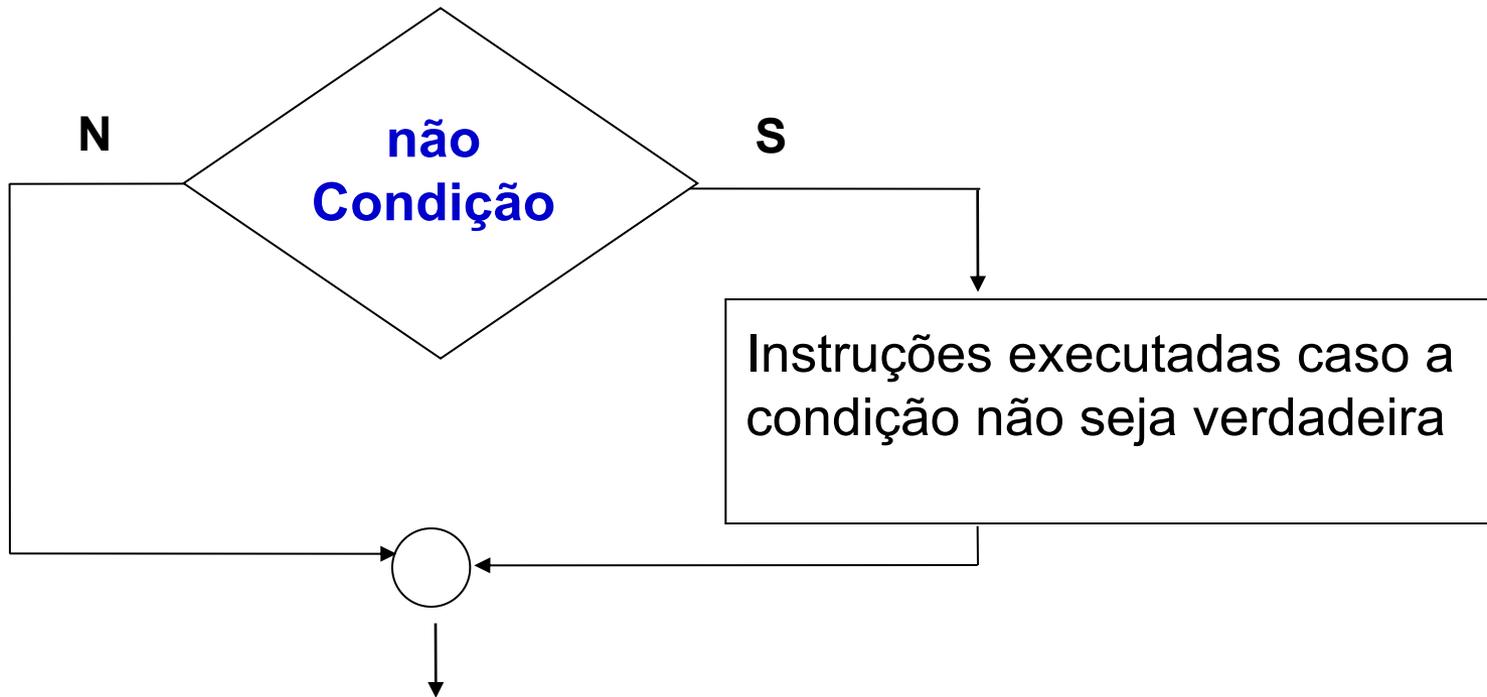
- O operador do tipo **não** é utilizado quando a necessidade de estabelecer a inversão do resultado lógico de uma determinada condição.
- Tabela-verdade

<b>Condição</b>	<b>Resultado</b>
Falsa	Verdadeira
Verdadeira	Falso

- Se **não(chuva)** eu vou para a praia.  
Quando eu vou para a praia?

# Operador lógico: não

- Fluxograma



# Operador lógico: não

- Português Estruturado

```
se nao(<condição>) entao
```

```
    C2 // instrução executada se a condição não for  
        verdadeira
```

```
fimse
```

# Operadores Lógicos

## ordem de precedência

<b>Prioridade</b>	<b>Operador</b>
1º	não
2º	e
3º	ou

# Exercícios

- 1) Resolva os problemas abaixo usando as 3 formas de representação de algoritmos: narrativa, fluxograma e português estruturado.
  - a) Faça um algoritmo que receba dois números e mostre qual deles é o maior
  - b) Faça um algoritmo que receba três números e mostre-os em ordem crescente
  - c) Faça um algoritmo que receba um número inteiro e verifique se este número é par ou ímpar

# Exercícios

2) Faça um algoritmo que receba a altura e o peso de uma pessoa. De acordo com a tabela a seguir, verifique e mostra qual a classificação dessa pessoa

Altura	Peso		
	Até 60	Entre 60 e 90 (Inclusive)	Acima de 90
Menor que 1,20	A	D	G
De 1,20 a 1,70	B	E	H
Maior que 1,70	C	F	I

# Exercícios

3) A nota final de um estudante é calculada a partir de três notas atribuídas respectivamente a um trabalho de laboratório, a uma avaliação semestral e a um exame final.

- se a média do aluno estiver entre 0,0 e 5,0, então seu conceito é E,

- se estiver entre 5,0 e 6,0, seu conceito é D,

- se estiver entre 6,0 e 7,0, seu conceito é C,

- se estiver entre 7,0 e 8,0, seu conceito é B,

- e se estiver entre 8,0 e 10,0, seu conceito é A.

Faça um algoritmo que receba as três notas, calcule e mostre a média e o conceito do aluno.

4) Reescreva o algoritmo ao lado usando Escolha ... caso

```
algoritmo "Opções de desconto"  
  OPCAO: inteiro  
inicio  
  escreva("Escolha a opcao de desconto(1, 2, 3 ou 4): ")  
  leia(OPCAO)  
  se (OPCAO = 1) entao  
    escreva("Dar desconto de 20%")  
  senao  
    se(opcao = 2) entao  
      escreva("Dar desconto de 15%")  
    senao  
      se (opcao = 3) entao  
        escreva("Dar desconto de 10%.")  
      senao  
        se (opcao = 4) entao  
          escreva("Dar desconto de 5%.")  
        senao  
          escreva("Não dar desconto.")  
      fimse  
    fimse  
  fimse  
fimse  
fimalgoritmo
```

# Testando o algoritmo

- Um algoritmo, depois de ser elaborado, pode (e deve) ser testado
- Utiliza-se um método conhecido como teste de mesa
  - É como uma simulação de todos os passos, entradas, comandos e instruções do algoritmo, a fim de saber se ele chega ao resultado a que se propõe e se a lógica está correta
  - Preenche-se uma tabela com valores para as variáveis e segue-se o fluxo de execução do algoritmo, simulando a execução de cada instrução
  - Se, para uma instrução executada, uma ou mais variáveis não ficaram com os valores esperados, há um erro na lógica do algoritmo

<b>Algoritmo</b>	<b>Teste de Mesa</b>			
<b>Algoritmo</b>				
<b>Var</b>				
a,b,c: REAL	a	b	c	Saída
<b>Inicio</b>	?	?	?	
a <- 5	5	?	?	
b <- 15	5	15	?	
c <- a+b	5	15	20	
<b>escreva</b> (c)	5	15	20	20
a <- 10	10	15	20	
b <- 25	10	25	20	
c <- a+b	10	25	35	
<b>escreva</b> (c)	10	25	35	35
a <- a-b	(10-25) = -15	25	35	
<b>escreva</b> (a)	-15	25	35	-15
a <- 0	0	25	35	
b <- 0	0	0	35	
c <- 0	0	0	0	
<b>Fimalgoritmo</b>				