




Sistema
FIRJAN  INFORMA, FORMA, TRANSFORMA.

Algoritmos e Estruturas de Dados



Fabício Curvello Gomes

Sistema
FIRJAN  INFORMA, FORMA, TRANSFORMA.



```
graph TD; A[Qual o caminho ?] --> B[Caminho 1]; A --> C[Caminho 2];
```

Programação Com Decisão

INFORMA, FORMA, TRANSFORMA.

Decisões, Condições e Operadores Relacionais

Condição – É uma obrigação que se impõe e se aceita.


Decisão – É o ato ou efeito de decidir, de tomar uma decisão.



O ato de tomar uma decisão está calcado no fato de haver uma condição.

Do ponto de vista computacional, condição é uma expressão booleana cujo resultado é um valor lógico **falso** ou **verdadeiro**.

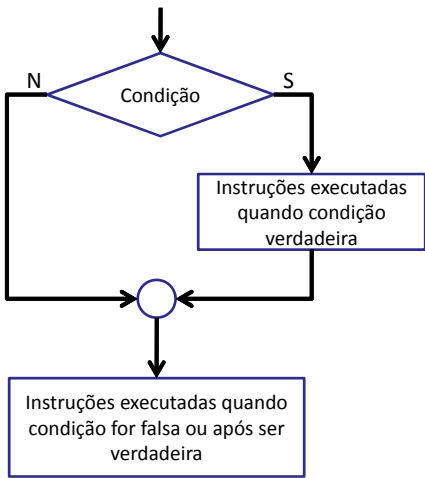
Para ter uma expressão booleana como condição, usa-se uma relação lógica entre dois elementos e um operador relacional.

Serão utilizados os operadores lógicos apresentados no capítulo 2.


3

INFORMA, FORMA, TRANSFORMA.

Desvio Condicional Simples

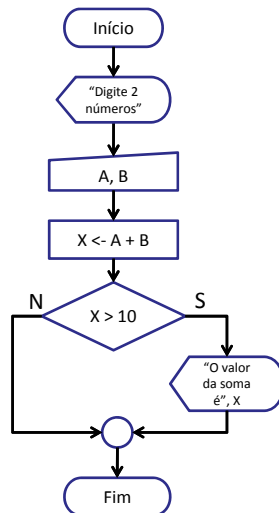


```
...  
se (<condição>) entao  
  <instruções para  
    condição verdadeira>  
fimse  
<instruções para condição  
falsa ou após ser  
verdadeira>
```

4

Desvio Condicional Simples (Cont.)

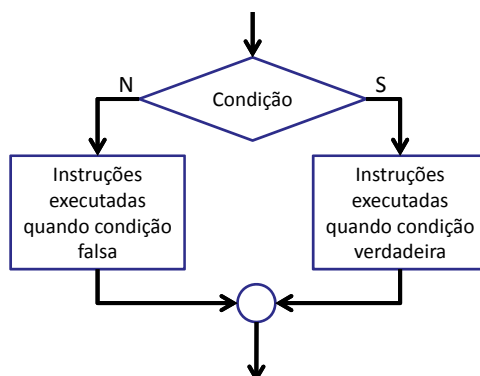
Exemplo: Algoritmo para ler dois valores numéricos, efetuar a adição e apresentar o resultado **caso** o valor seja maior que 10.



```
algoritmo "CONDSIMPLES"  
// Função : Ex de Desvio Cond Simples  
// Autor : Fabrício Curvello Gomes  
// Data : 8/9/2010  
// Seção de Declarações  
var  
    A, B, X: inteiro  
inicio  
// Seção de Comandos  
    Escreval ("Digite 2 números:")  
    Leia (A,B)  
    X <- A + B  
    se X > 10 entao  
        Escreval ("O valor da soma é",X)  
    fimse  
fimalgoritmo
```

5

Desvio Condicional Composto

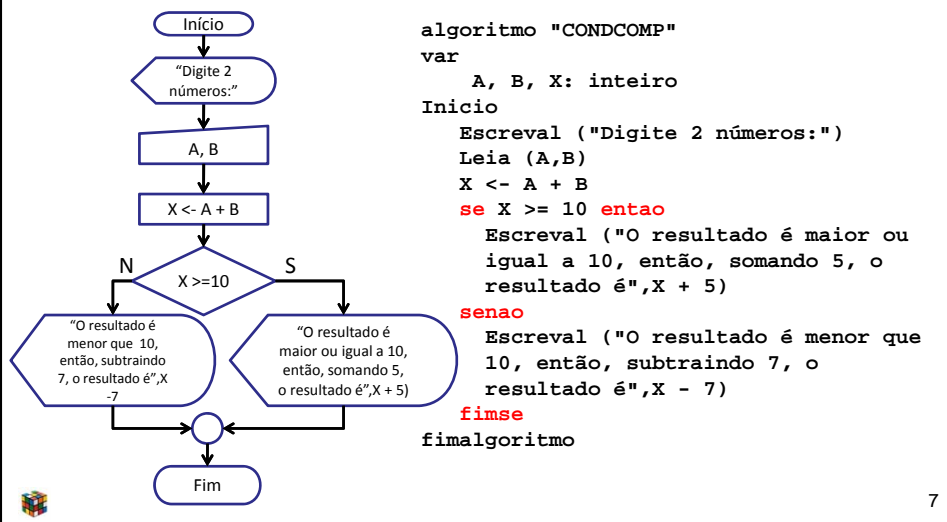


```
...  
se (<condição>) entao  
    <instruções para  
    condição verdadeira>  
senao  
    <instruções para  
    condição falsa>  
fimse
```

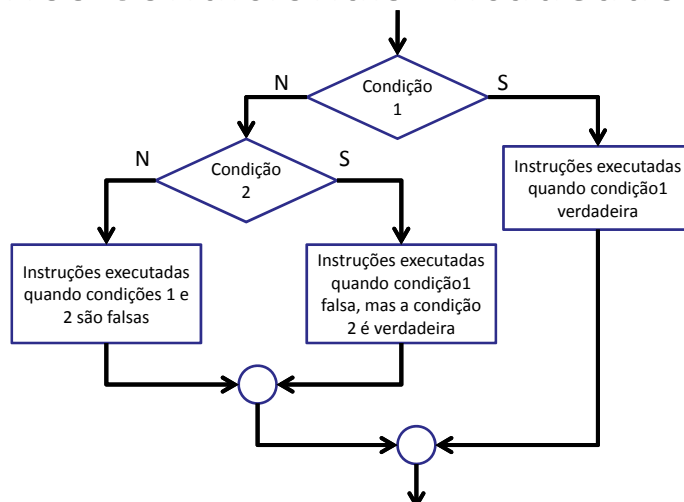
6

Desvio Condicional Composto (Cont.)

Exemplo: Algoritmo para ler dois valores numéricos, efetuar a adição. Caso a soma seja maior ou igual a 10, apresente o resultado somando 5. Caso a soma seja menor que 10, apresente o resultado subtraindo 7.

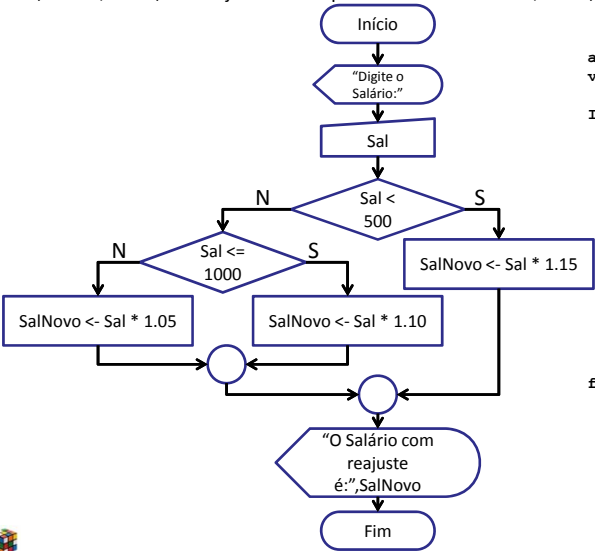


Desvios Condicionais Encadeados



Desvios Condicionais Encadeados (Cont.)

Exemplo: Elaborar um programa que calcule o reajuste de salário de um funcionário, sob as seguintes condições: Reajuste de 15% para salário menor que R\$ 500,00. Reajuste de 10% para salário entre R\$ 500,00 e R\$ 1000,00. Reajuste de 5% para salário acima de R\$ 1000,00.



```
algoritmo "CONDENC"
var
    Sal, SalNovo: real
Inicio
    Escreval ("Digite o Salário:")
    Leia (Sal)
    Se (Sal < 500) entao
        SalNovo <- Sal * 1.15
    Senao
        Se (Sal <= 1000) entao
            SalNovo <- Sal * 1.10
        Senao
            SalNovo <- Sal * 1.05
        fimse
    fimse
    Escreval ("O Salário com reajuste é:",SalNovo)
fimalgoritmo
```



Operadores Lógicos (Cont.)

Conforme já foi visto no Capítulo 2, segue novamente a tabela verdade dos operadores lógicos:

A	B	NÃO A	NÃO B	A E B	A OU B	A XOU B
VERDADEIRO	VERDADEIRO	FALSO	FALSO	VERDADEIRO	VERDADEIRO	FALSO
VERDADEIRO	FALSO	FALSO	VERDADEIRO	FALSO	VERDADEIRO	VERDADEIRO
FALSO	VERDADEIRO	VERDADEIRO	FALSO	FALSO	VERDADEIRO	VERDADEIRO
FALSO	FALSO	VERDADEIRO	VERDADEIRO	FALSO	FALSO	FALSO



Operador Lógico E

```
graph TD; Entry(( )) --> Cond{Condição1 e Condição2}; Cond -- S --> Instrucoes[Instruções executadas se ambas as condições forem verdadeiras]; Cond -- N --> Merge(( )); Instrucoes --> Merge; Merge --> Exit(( ))
```

...

```
se (<condição1 e  
condição2>) entao  
  <instruções executadas se  
  ambas forem verdadeiras>  
fimse
```

11

Operador Lógico E (Cont.)

Algoritmo que testa se um numero dado está na faixa de 20 a 90.

```
graph TD; Inicio([Início]) --> Input[/Digite o Número: /]; Input --> Num[Num]; Num --> Cond{Num >= 20 e Num <= 90}; Cond -- S --> OutputS[/O número está entre 20 e 90 /]; Cond -- N --> OutputN[/Número não está entre 20 e 90 /]; OutputS --> Merge(( )); OutputN --> Merge; Merge --> Fim([Fim])
```

```
algoritmo "Operador E"  
  // Função : Exemplo de Operador E  
  // Autor : Fabrício Curvello Gomes  
  // Data : 8/9/2010  
  // Seção de Declarações  
  var  
    Num: inteiro  
  inicio  
    // Seção de Comandos  
    Escreval ("Digite o Número:")  
    Leia (Num)  
    Se (Num >= 20) e (Num <= 90) entao  
      escreval ("Número está entre 20 e 90")  
    senao  
      escreval ("Número não está entre 20 e 90")  
    fimse  
  finalgoritmo
```

12

Operador Lógico OU

```
graph TD; Entry(( )) --> Cond{Condição1  
ou  
Condição2}; Cond -- N --> Join(( )); Cond -- S --> Instrucoes[Instruções executadas  
se pelo menos uma  
das condições for  
verdadeira]; Instrucoes --> Join; Join --> Exit(( ))
```

...

```
se (<condição1 ou  
condição2>) então  
<instruções executadas se  
pelo menos uma das  
condições for verdadeira>  
fimse
```

13

Operador Lógico OU (Cont.)

Algoritmo que testa a validade do sexo informado.

```
graph TD; Inicio([Início]) --> Input[/Informe seu sexo: /]; Input --> Sexo[Sexo]; Sexo --> Cond{Sexo = "masculino" ou  
Sexo = "feminino"}; Cond -- N --> Invalid[/O seu sexo é duvidoso /]; Cond -- S --> Valid[/O seu sexo é válido /]; Invalid --> Join(( )); Valid --> Join; Join --> Fim([Fim])
```

```
algoritmo "Operador OU"  
// Função : Exemplo de Operador OU  
// Autor : Fabrício Curvello Gomes  
// Data : 8/9/2010  
// Seção de Declarações  
var  
    Sexo: caracter  
inicio  
// Seção de Comandos  
Escreval ("Informe seu sexo:")  
Leia (Sexo)  
Se (Sexo ="masculino") ou (Sexo  
="feminino") então  
    escreval ("O seu sexo é válido")  
senao  
    escreval("O seu sexo é duvidoso")  
fimse  
finalgoritmo
```

14

Operador Lógico NÃO

```
graph TD
    Entry(( )) --> Cond{Não Condição}
    Cond -- N --> Join(( ))
    Cond -- S --> Box[Instruções executadas caso a condição não seja verdadeira]
    Box --> Join
    Join --> Exit(( ))
```

...

```
se nao(<condição>) entao
    <Instruções executadas caso a condição não seja verdadeira>
fimse
```

15

Operador Lógico NÃO (Cont.)

Observe o que este algoritmo faz:

```
graph TD
    Inicio([Início]) --> Input[/Informe 3 números:/]
    Input --> Var[A, B, X]
    Var --> Cond{Não (X > 5)}
    Cond -- N --> Calc1[C <- (A - B) * X]
    Cond -- S --> Calc2[C <- (A + B) * X]
    Calc1 --> Join(( ))
    Calc2 --> Join
    Join --> Output[/C/]
    Output --> Fim([Fim])
```

```
algoritmo "Operador NAO"
// Função : Exemplo de Operador NAO
// Autor : Fabrício Curvello Gomes
// Data : 8/9/2010
// Seção de Declarações
var
    A, B, C, X: inteiro
inicio
// Seção de Comandos
Escreval ("Informe 3 números:")
Leia (A, B, X)
Se não (X > 5) entao
    C <- (A + B) * X
senao
    C <- (A - B) * X
fimse
Escreval ("O valor de C é:", C)
finalgoritmo
```

Na verdade, esta sentença equivale a: Se (X < 5) entao


16




Dúvidas?




17




Bibliografia



Estudo Dirigido de Algoritmos
José Augusto N. G. Manzano e Jayr Figueiredo de Oliveira
Ed. Érica



Introdução aos Algoritmos
Bruno Tonet e Cristian Koliver
(Acompanha o software VisuAlg)



A Linguagem de Programação do VisuAlg
(Acompanha o software VisuAlg)

18