

# Estruturas de Seleção

José Ricardo da Silva Junior

Adaptado de slides da prof. Vania Bogorny

# Estruturas de Decisão

- ESTRUTURAS DE DECISÃO
  - Comandos de **decisão** ou **desvio** fazem parte das técnicas de programação, para construir estruturas de algoritmos que *não são* totalmente seqüenciais.
  - Com as instruções de **desvio** pode-se fazer com que o algoritmo proceda de uma ou outra maneira, de acordo com as decisões lógicas tomadas em função dos dados ou resultados anteriores.
  - As principais estruturas de decisão são:
    - SE...ENTÃO
    - SE...ENTÃO...SENÃO
    - CASO FOR...ENTÃO

# SE-ENTÃO

- Formato:

- **Se** <operação lógica> **então** <ações>
- Significado: Se a <operação lógica> resultar em verdadeiro, então executar as <ações>. Senão, simplesmente ignorar as <ações> e seguir para a próxima instrução no algoritmo.
- Usada para decidir se *um conjunto de ações opcionais* deve ser executado ou não, dependendo do valor de algum dado ou de algum resultado que já tenha sido calculado no algoritmo. O valor do dado ou do resultado anterior será testado na operação lógica.

# SE-ENTÃO

- Exemplo da estrutura SE...ENTÃO:  
“Avisar se um número lido for negativo.”

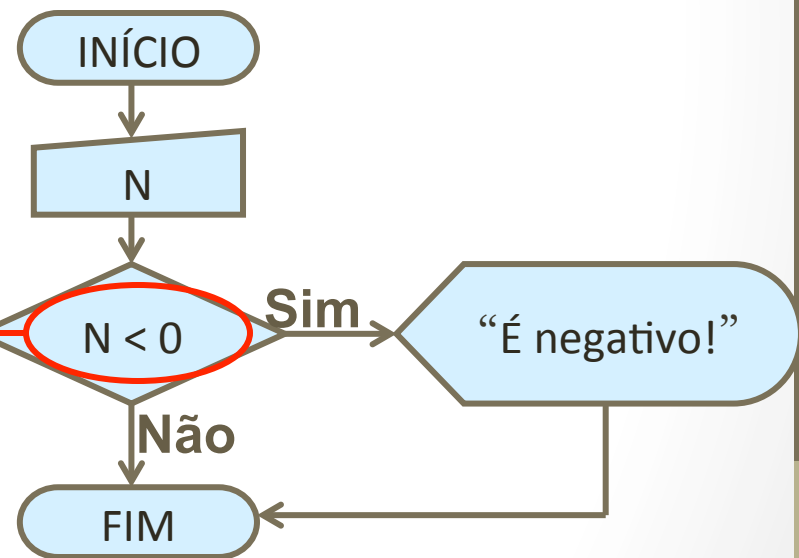
- PSEUDOCÓDIGO:

```
Ler N  
Se (N < 0)  
então exibir “É negativo!”
```

- FLUXOGRAMA:

## OPERAÇÃO LÓGICA:

Poderá ser V ou F, dependendo do valor de N, que foi lido antes.



# SE-ENTÃO

- Comando **if**

```
if (condição)
    comando;
```

```
if (condição) {
    comando1;
    comando2;
    comando3;
}
```

```
if (a<menor)
    menor=a;
```

```
if (a<menor) {
    menor=a;
    printf ( "%d" , menor);
}
```

em pseudo-código:

se (a<menor) então menor=a;

# SE-ENTÃO-SENÃO

- Formato:

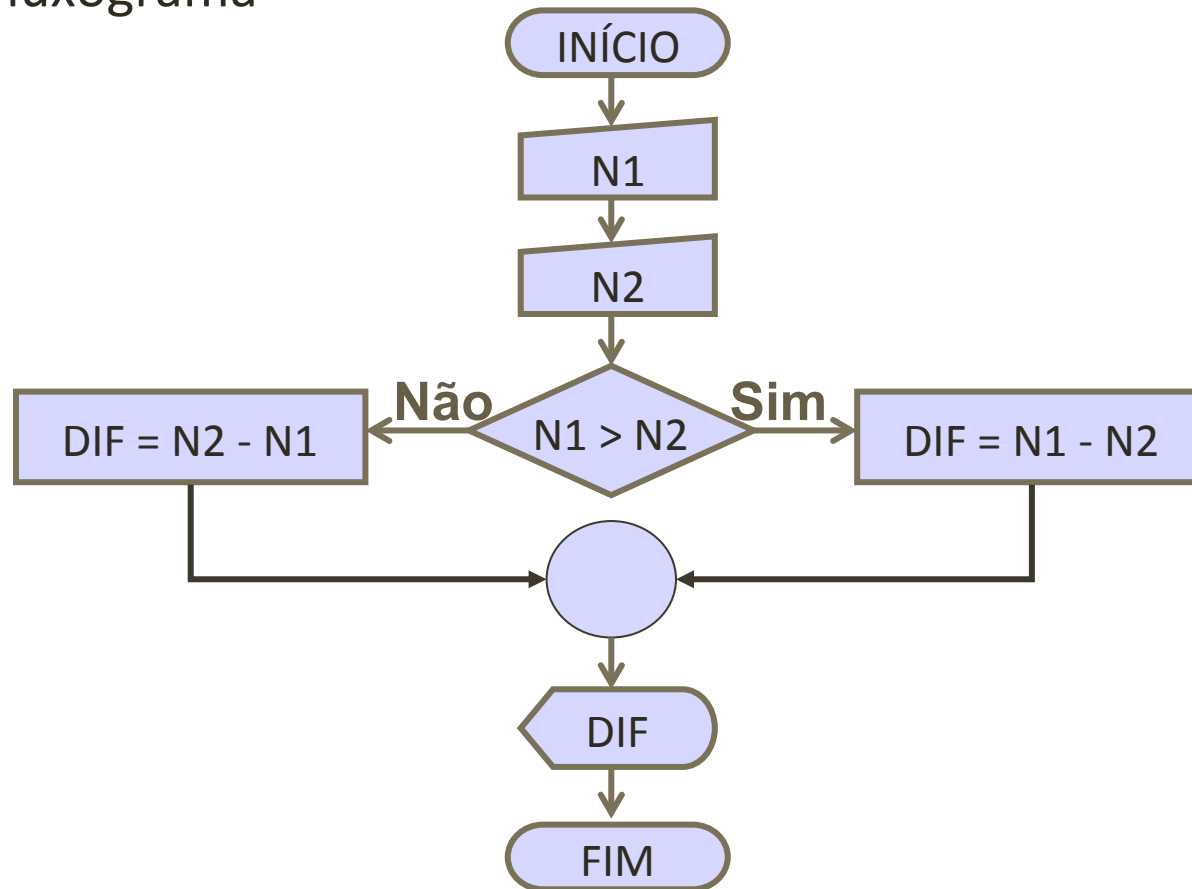
**Se** <operação lógica> **então** <ações 1>  
**senão** <ações 2>

- Significado: Se a <operação lógica> resultar em verdadeiro, então executar <ações 1>. Senão, ignorar <ações 1> e executar <ações 2>.
- Usada para decidir entre *duas alternativas de ações*. *Um* dos dois conjuntos de ações será executado e *o outro não*, dependendo do valor de algum dado ou de algum resultado que já tenha sido calculado no algoritmo. O valor do dado ou do resultado anterior será testado na operação lógica, determinando qual conjunto de ações será executado.



# SE-ENTÃO-SENÃO

- Fluxograma



# SE-ENTÃO-SENÃO

- Comando **if...else**

```
if (condição)
    comando;
else
    comando;

if (condição) {
    comando1;
    comando2;
}
else {
    comando3;
    comando4;
}
```

Executa o comando se a condição for qualquer coisa diferente de zero!

```
if (peso == peso_ideal)
    printf ( "Vc está em forma!" );
else
    printf ( "Necessário fazer dieta!" );
```

em pseudo-código:

```
se (peso == peso_ideal)
    entao exibir "Vc está em forma!"
senao exibir "Necessário dieta!"
```

# SE-ENTÃO-SENÃO

- Encadeamento

**Se** <operação lógica> **então** <ação 1>  
    **senão se** <operação lógica> **então** <ação 2>  
        **senão** <ação 3>

**Exemplo:**

**se** (valor<0) **então** res = 0;  
    **senão se** (valor>10) **então** res = 2;  
        **senão** res = 3;

# CASO-ENTÃO

- Formato:

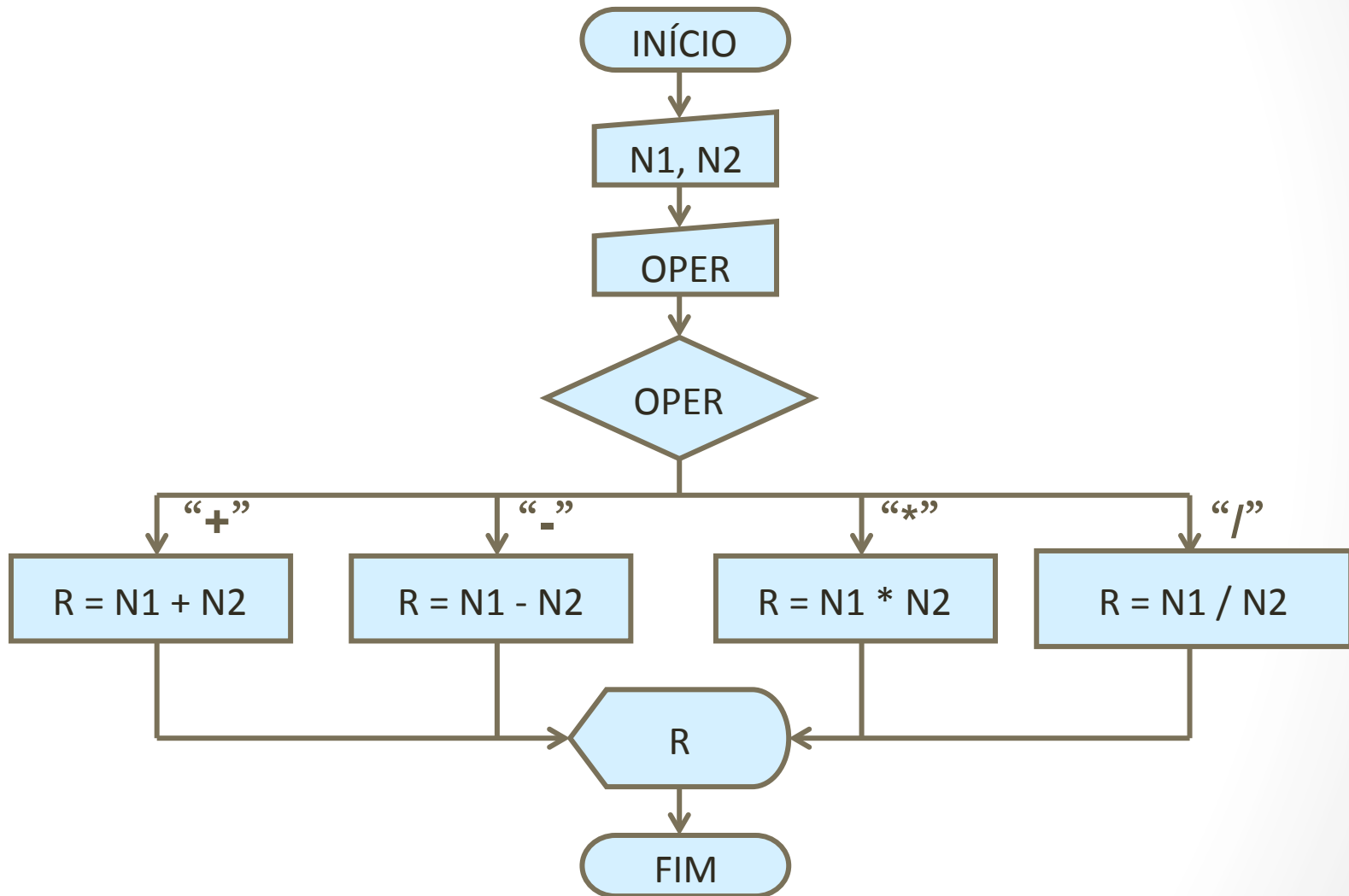
**Caso** <valor 1> **então** <ações 1>  
**Caso** <valor 2> **então** <ações 2>  
**Caso** <valor 3> **então** <ações 3>  
**Caso** <etc., quantos precisar>

- Significado: Se <valor> for igual a <valor 1>, então executar <ações 1>. Se for igual a <valor 2>, ignorar <ações 1> e executar <ações 2>, e assim por diante.
- Usada para escolher *apenas um conjunto de ações* dentre vários alternativos. Aqui o teste não é mais uma operação lógica: *o próprio valor* de algum dado ou resultado anterior (que pode ser de outros tipos além do lógico) é que vai determinar qual desses conjuntos de ações será executado

# CASO-ENTÃO

- Exemplo da estrutura CASO...ENTÃO:  
“Escolher uma das 4 operações básicas para aplicar em dois números.”
  - PSEUDOCÓDIGO:
    - Ler N1
    - Ler N2
    - Ler OPERACAO (que será do tipo caractere)
    - **Caso OPERACAO:**
      - “+” **então** RESULT = N1 + N2
      - “-” **então** RESULT = N1 - N2
      - “\*” **então** RESULT = N1 \* N2
      - “/” **então** RESULT = N1 / N2
    - caso contrario** exibir “Operação não válida!”
  - Exibir RESULT

# CASO-ENTÃO



# CASO-ENTÃO

```
switch (numero){  
    case 1: printf ("Janeiro\n"); break;  
    case 2: printf ("Fevereiro\n"); break;  
    case 3: printf ("Marco\n"); break;  
    case 4: printf ("Abril\n"); break;  
    case 5: printf ("Maio\n"); break;  
    case 6: printf ("Junho\n"); break;  
    case 7: printf ("Julho\n"); break;  
    case 8: printf ("Agosto\n"); break;  
    case 9: printf ("Setembro\n"); break;  
    case 10: printf ("Outubro\n"); break;  
    case 11: printf ("Novembro\n"); break;  
    case 12: printf ("Dezembro\n"); break;  
    default: printf ("Mes invalido\n");  
}
```

# Exercício 1

**Dados dois números A e B, some 100 ao maior número e imprima.**

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    int A, B, soma;
    printf ("Digite o valor de A: ");
    scanf ("%d", &A);
    printf ("Digite o valor de B: ");
    scanf ("%d", &B);
    if (A>B)
        soma=A+100;
    else
        soma=B+100;
    printf ("O maior valor adicionado de 100 resulta em %d\n", soma);
    system("pause");

    return 0;
}
```

# Exercício 2

**Escreva um algoritmo para determinar se uma pessoa é maior ou menor de idade.**

```
#include <stdio.h>
#include <stdlib.h>
main (){
int idade;
printf ("Digite a idade da pessoa: ");
scanf ("%d", &idade);
if (idade>=18)
    printf ("Pessoa eh maior de idade.\n");
else
    printf ("Pessoa eh menor de idade.\n");
system("pause");
}
```

# Exercício 3

Ler um número inteiro, e verificar se o número corresponde a um mês válido no calendário e escrever o nome do mês, senão escrever uma mensagem 'Mês Inválido'.

```
#include <stdlib.h>
#include <stdio.h>

main(){
    int numero;
    printf ("\nDigite o numero do mes: ");
    scanf ("%d", &numero);
    if (numero<=0 || numero>12)
    {
        printf ("\nMes invalido!\n");
        system("pause");
        exit(0);
    }
    switch (numero){
        case 1: printf ("Janeiro\n"); break;
        case 2: printf ("Fevereiro\n"); break;
        case 3: printf ("Marco\n"); break;
        case 4: printf ("Abril\n"); break;
        case 5: printf ("Maio\n"); break;
        case 6: printf ("Junho\n"); break;
        case 7: printf ("Julho\n"); break;
        case 8: printf ("Agosto\n"); break;
        case 9: printf ("Setembro\n"); break;
        case 10: printf ("Outubro\n"); break;
        case 11: printf ("Novembro\n"); break;
        case 12: printf ("Dezembro\n"); break;
    }
    system("pause");
}
```

# Exercícios

- 1. Leia 3 valores inteiros (X, Y, Z), determina e escreve o menor deles.**
- 2. A Secretaria de Meio Ambiente, que controla o índice de poluição, mantém 3 grupos de indústrias que são altamente poluentes do meio ambiente. O índice de poluição aceitável varia de 0,05 até 0,29. Se o índice sobe para 0,3 as indústrias do 1º grupo são intimadas a suspenderem suas atividades, se o índice crescer para 0,4 as indústrias do 1º e 2º grupo são intimadas a suspenderem suas atividades, se o índice atingir 0,5 todos os grupos devem ser notificados a paralisarem suas atividades. Faça um programa que leia o índice de poluição medido e emita a notificação adequada aos diferentes grupos de empresas.**