

# Algoritmos e Estruturas de dados 1

## Estrutura Condicional

Prof. Ivre Marjorie  
[ivre.marjorie@gmail.com](mailto:ivre.marjorie@gmail.com)

---

# Estrutura Condicional em Algoritmos

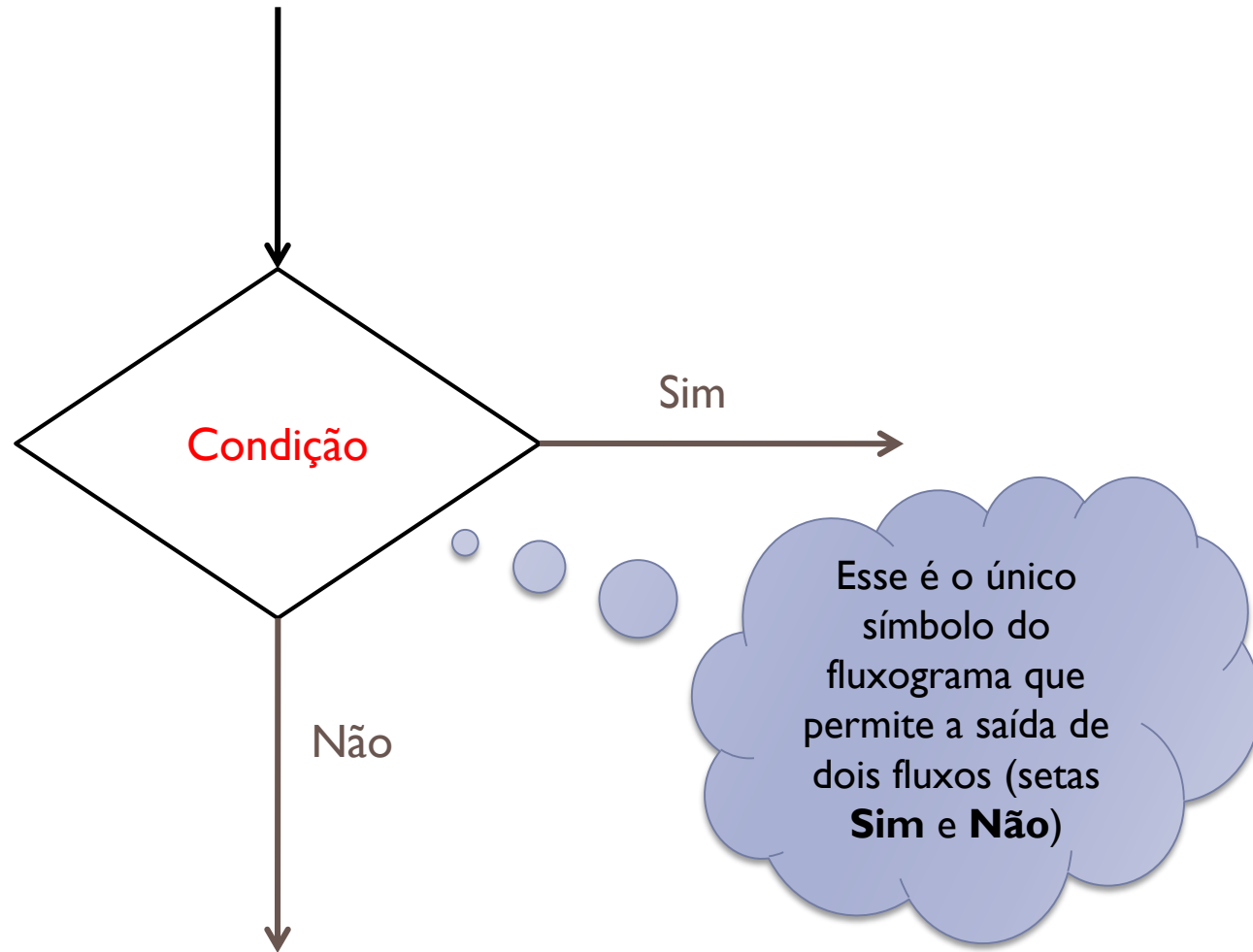


# Estrutura Condicional - Fluxograma

---

Símbolo usado para representar a **tomada de decisão**.

Se a condição for verdadeira o fluxo continua para o lado do **Sim**, se a condição for falsa, o fluxo continua para o lado do **Não**



# Estrutura Condicional - Pseudocódigo

---

- ▶ A estrutura condicional em pseudocódigo pode ser feita da forma simples:

SE (**condição**)  
    **Comando I**  
FIM\_SE

SE ( $x > 2$ )  
    Escreva “o valor de x é maior que dois”  
FIM\_SE  
SE ( $x \leq 2$ )  
    Escreva “o valor de x é menor ou igual a dois”  
FIM\_SE

- O **comando I** só será executado se a **condição** for verdadeira
  - Uma condição é uma comparação que possui dois valores possíveis, verdadeiro ou falso.
-

# Estrutura Condicional - Pseudocódigo

---

## ► Ou composta:

```
SE (condição)  
  Comando1  
SENÃO  
  Comando2  
FIM_SE_SENÃO
```

```
SE ( $x > 2$ )  
  Escreva “o valor de x é maior que dois”  
  
SENÃO  
  Escreva “o valor de x é menor ou igual a dois”  
FIM_SE_SENÃO
```

➤ Se a **condição** for *verdadeira*, será executado o **comando1**, caso contrário, se a **condição** for *falsa*, será executado o **comando2**.

---



# Estrutura Condicional - Pseudocódigo

---

## *Comparando...*

### Simples:

**SE (x > 2)**

Escreva “o valor de x é maior que dois”

**FIM\_SE**

**SE (x <= 2)**

Escreva “o valor de x é menor ou igual a dois”

**FIM\_SE**

### Composta:

**SE (x > 2)**

Escreva “o valor de x é maior que dois”

**SENÃO**

Escreva “o valor de x é menor ou igual a dois”

**FIM\_SE\_SENÃO**



# Estrutura Condicional - Pseudocódigo

---

## Comparando...

### Simple:

**SE (x > 2)**

Escreva “o valor de x é maior que dois”

**FIM\_SE**

**SE (x <= 2)**

Escreva “o valor de x é menor ou igual a dois”

**FIM\_SE**

### Composta:


**SE (x > 2)**

Escreva “o valor de x é maior que dois”

**SENÃO**

Escreva “o valor de x é menor ou igual a dois”

**FIM\_SE\_SENÃO**



Na estrutura simples o teste da condição é feito em cada um deles, ou seja, verifica se é **verdadeiro** ou **falso**. Caso nenhum deles seja verdadeiro, nenhum será executado. E se mais de um for Verdadeiro, mais de um será executado.

# Estrutura Condicional - Pseudocódigo

---

## *Comparando...*

### Simples:

**SE (x > 2)**

Escreva “o valor de x é maior que dois”

**FIM\_SE**

**SE (x <= 2)**

Escreva “o valor de x é menor ou igual a dois”

**FIM\_SE**

### Composta:

**SE (x > 2)**


Escreva “o valor de x é maior que dois”

**SENÃO**

Escreva “o valor de x é menor ou igual a dois”

**FIM\_SE\_SENÃO**

Na estrutura composta o teste da condição é feito no primeiro, e caso seja verdadeiro somente executar os comando relacionados a ele e finalizará. Caso seja falso, irá verificar o próximo teste de condição e assim como o primeiro se for verdadeiro executará apenas ele. E somente no caso de todos os teste forem falsos, que irá executar os comandos relacionados ao Senão.





# Estrutura Condicional

---

## Estrutura Condicional Simple em Pseudocódigo

```
SE (condição)  
    comando1  
    comando2  
    comando3  
FIM_SE
```



# Estrutura Condicional

---

## Estrutura Condicional Composta em Pseudocódigo

```
SE condição
    comando1
    comando2
SENÃO
    comando3
    comando4
FIM_SE_SENÃO
```

Se a **condição** for verdadeira, o **comando1** e o **comando2** serão executados; caso contrário, se a **condição** for falsa, o **comando3** e o **comando4** serão executados.

---



# Exemplos

---

- 1- Faça um algoritmo (**pseudocódigo** e **fluxograma**) que receba um número como entrada. Verifique se o número é maior ou igual a 2 e mostre na tela uma mensagem.
  
- 2- Faça um algoritmo (**pseudocódigo** e **fluxograma**) que receba dois números como entrada e mostre na tela o maior.



# Exemplo I

---

ALGORITMO

DECLARE num NUMERICO

ESCREVA "Digite um numero: "

LEIA num

**SE (num  $\geq$  2)**

ESCREVA "O numero digitado é maior ou igual a 2"

**SENÃO**

ESCREVA "O numero digitado é menor que 2"

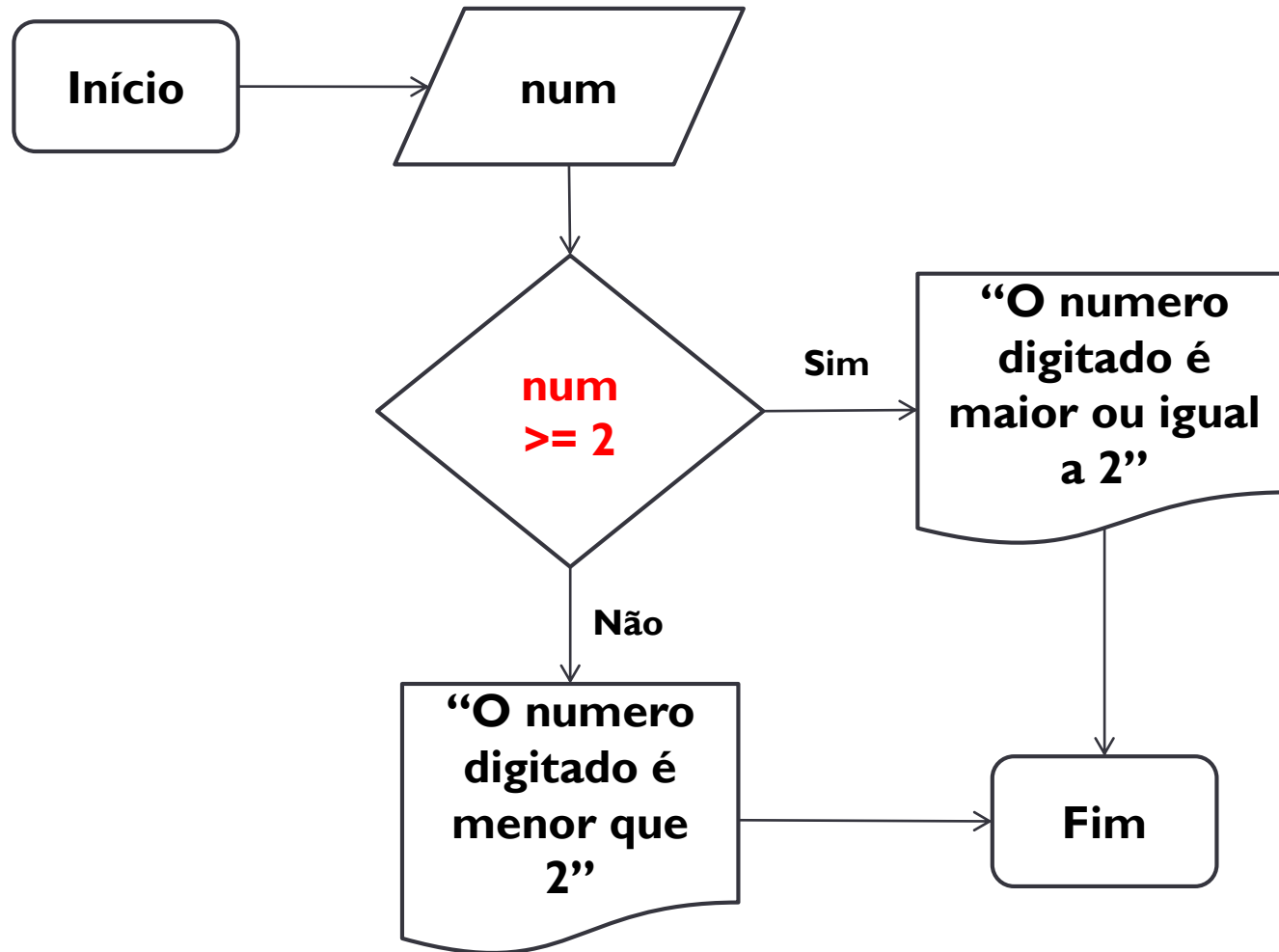
**FIM\_SE\_SENÃO**

FIM\_ALGORITMO

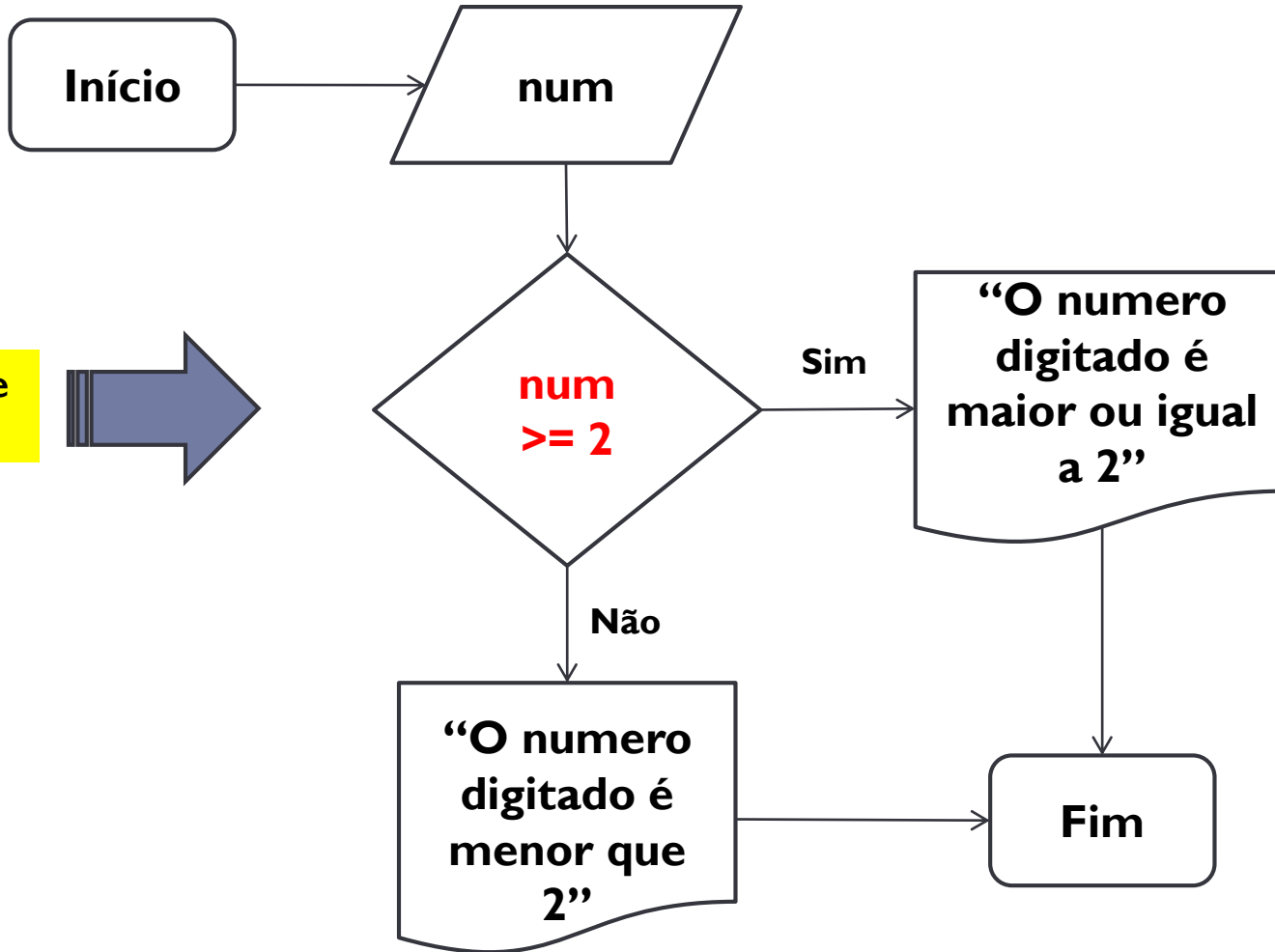
---



# Exemplo I



# Exemplo I



# Exemplo 2

## ALGORITMO

DECLARE num1, num2 NUMERICO

ESCREVA "Digite o primeiro numero"

LEIA num1

ESCREVA "Digite o segundo numero"

LEIA num2

**SE (num1 > num2)**

ESCREVA "O maior numero é ", num1

**SENÃO SE (num2 > num1)**

ESCREVA "O maior numero é ", num2

**SENÃO**

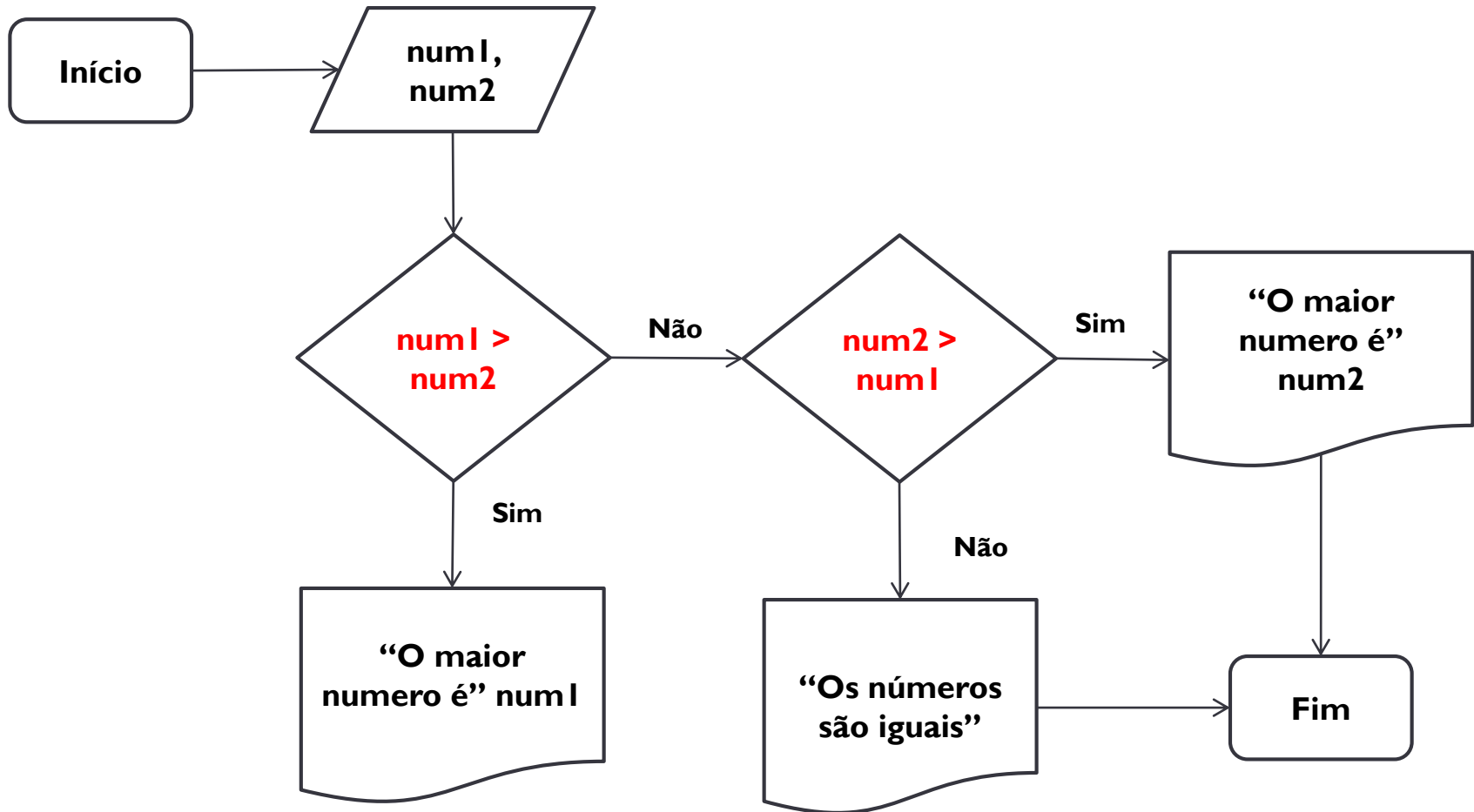
ESCREVA "Os números são iguais !"

**FIM\_SE\_SENÃO**

---

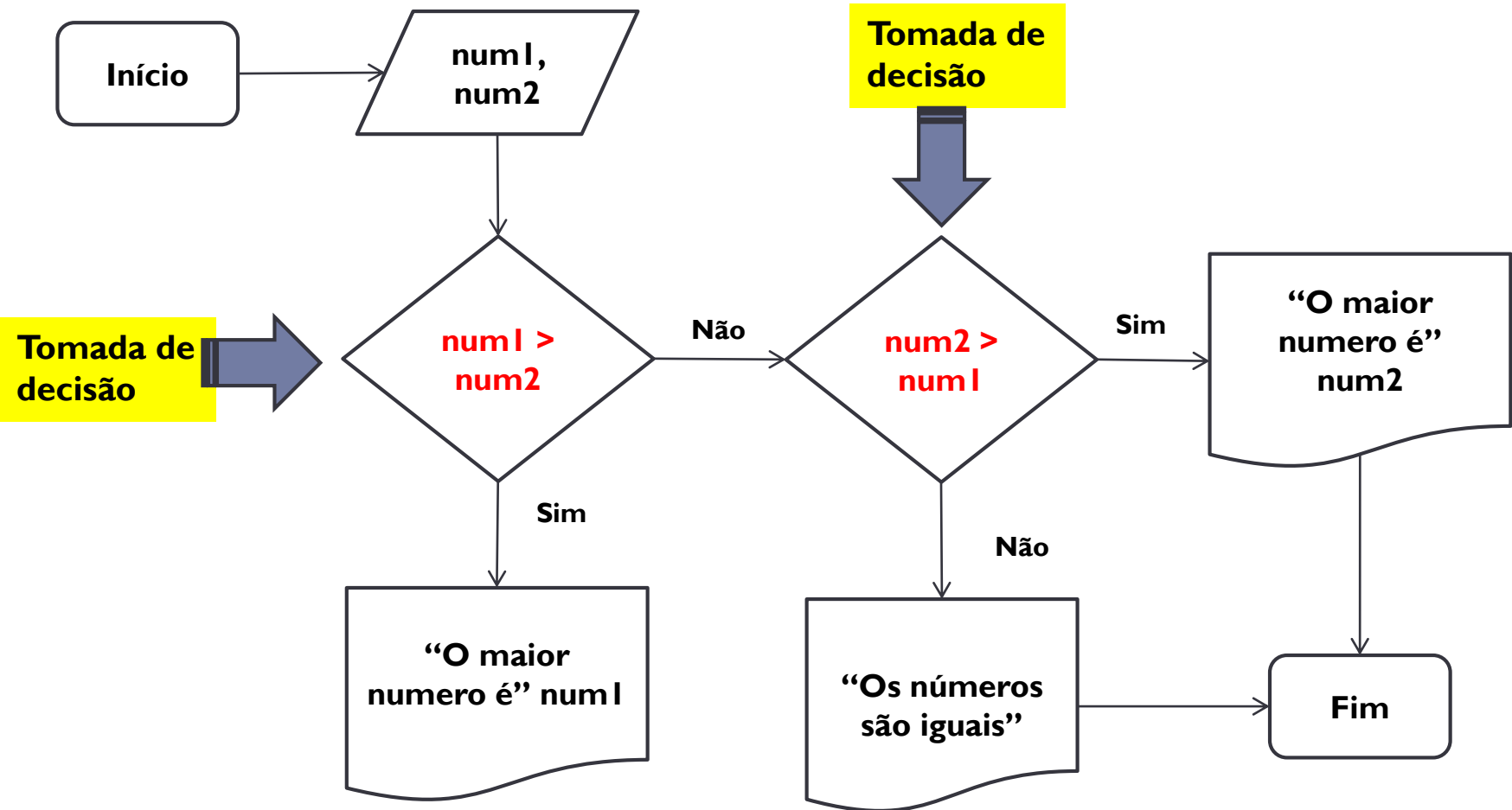
FIM\_ALGORITMO

# Exemplo 2





# Exemplo 2



---

# Estrutura Condicional em C



# Estrutura Condicional

---

## Estrutura Condicional Simples em C

```
if (condição)  
{  
    comando;  
}
```

➤ O **comando** só será executado se a **condição** for verdadeira, Uma condição é uma comparação que possui dois valores possíveis: verdadeiro ou falso.

▶ **Obs.:** Todas as condições devem estar entre **parênteses ( )**.

# Estrutura Condicional

---

## Estrutura Condicional Simples em C

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



# Estrutura Condicional

---

## Estrutura Condicional Simples em C

- ▶ Em c, torna-se obrigatória a utilização de chaves quando houver mais de um comando a ser executados.
- ▶ Os **comandos** entre as **chaves { }** só serão executados se a condição for *verdadeira*.



# Estrutura Condicional

---

## Estrutura Condicional Composta em C

```
if (condição)  
{  
    comando I;  
}  
else  
{  
    comando2 ;  
}
```

- Se a **condição** for *verdadeira*, será executado o **comando I**, caso contrário, se a **condição** for *falsa*, será executado o **comando2**.

# Estrutura Condicional

---

## Estrutura Condicional Composta em C

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



# Estrutura Condicional

---

## Estrutura Condicional Composta em C

- ▶ Se a **condição** for *verdadeira*, o **comando1** e o **comando2** serão executados; caso contrário, se a **condição** for *falsa*, o **comando3** e o **comando4** serão executados.

**Obs.:** Todas as condições devem estar entre **parênteses ( )**.

---





# Estrutura Condicional

## Estrutura Condicional Composta em C

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

É possível na estrutura condicional composta, ter mais de uma condição, para isso, usamos o **else + if(condição)**

# Estrutura Condicional

---

## **Estrutura Condicional Composta em C**

- ▶ É possível usar quantas condições forem necessárias.
- ▶ A principal diferença entre a estrutura condicional composta e a simples, é que na estrutura composta, apenas um bloco de comandos será executado
  - ▶ Aquele cuja condição for verdadeira, ou então o que estiver no else
- ▶ Já na estrutura simples é possível que exista mais de uma condição verdadeira e os blocos de comandos serão executados
  - ▶ Todas as condições são testadas



# Operadores Lógicos em C

- ▶ Os principais operadores lógicos são:

## Operador Lógico

Operador	Exemplo	Comentário
<b>&amp;&amp;</b>	if (x >= 3 <b>&amp;&amp;</b> X <= 7 )	Significa que a condição verificada será X >= 3 <b>E</b> X <= 7 , ou seja, a condição será verdadeira se X estiver entre 3 e 7
<b>  </b>	if (x >= 3 <b>  </b> X >= 7 )	Significa que a condição verificada será X >= 3 <b>OU</b> X >= 7 , ou seja, tanto se X for maior que 3 como se X for maior que 7 a condição será verdadeira
<b>!</b>	if (x!=0)	A ! significa <i>negação</i> , ou seja, nesse exemplo verifica se X é diferente de 0



# Tabela verdade

---

Tabela <b>E</b> <b>&amp;&amp;</b>	Tabela <b>OU</b> <b>  </b>	Tabela <b>NÃO</b> <b>!</b>
<b>V e V = V</b>	<b>V ou V = V</b>	<b>Não V = F</b>
<b>V e F = F</b>	<b>V ou F = V</b>	<b>Não F = V</b>
<b>F e V = F</b>	<b>F ou V = V</b>	
<b>F e F = F</b>	<b>F ou F = F</b>	



# Operadores Lógicos em C

---

## Exemplos

```
if (num == 2)
{
    Console.WriteLine("numero igual a 2 ");
}
```

```
if (num > 5 && num < 10)
{
    Console.WriteLine("numero entre 5 e 10");
}
```



# Operadores Lógicos em C

---

## Exemplos

```
if (num == 5 || num == 10)
{
    Console.WriteLine("numero igual 5 ou igual a 10");
}
```

```
if (num != 0)
{
    Console.WriteLine("numero diferente de 0");
}
```



# Exemplos

---

- 1- Faça um programa em C que receba um número como entrada. Verifique se o número é maior ou igual a 2 e mostre na tela uma mensagem.
  
- 2- Faça um programa em C que receba dois números como entrada e mostre na tela o maior.



# Exemplo I

---

```
int main()
{
    int num;
    printf("Digite um numero: ");
    scanf("%d", &num);
    if(num >=2)
    {
        printf("O numero %d e maior ou igual a dois", num);
    }
    else
    {
        printf("O numero %d e menor que dois", num);
    }
    return 0;
}
```





# Exemplo 2

```
int main()
{
    float num1, num2;
    printf("Digite o primeiro numero: ");
    scanf("%f", &num1);
    printf("Digite o segundo numero: ");
    scanf("%f", &num2);
    if(num1 > num2)
    {
        printf("O numero %f e maior do que o numero %f", num1, num2);
    }
    else if(num2 > num1)
    {
        printf("O numero %f e maior do que o numero %f", num2, num1);
    }
    else
    {
        printf("Os numeros %f e %f sao iguais", num1, num2);
    }
    return 0;
}
```

# Estrutura Condicional

---

## Estrutura **CASE**

```
switch(variável)  
{  
    case valor1:  
        lista de comandos;  
    break;  
    case valor2:  
        lista de comandos;  
    break;  
    ...  
    default: lista de comandos;  
}
```

# Estrutura Condicional

---

## Estrutura **CASE**

- Em situações mutuamente exclusivas, isto é, se uma situação for executada, as demais não serão, um **comando seletivo** é o mais indicado (estrutura CASE)
- O comando **switch**(**variável**) avalia o valor da **variável** para decidir qual **case** será executado
- Cada **case** está associado a UM possível valor da variável, que deve ser obrigatoriamente, do tipo char ou int



# Estrutura Condicional

---

## Estrutura **CASE**

```
switch(variável)  
{  
    ...  
    break;  
    default: lista de comandos;  
}
```

- O comando **break**; deve ser utilizado para impedir a execução dos comando definidos nos **cases** subsequentes



# Estrutura Condicional

---

## Estrutura **CASE**

```
switch(variável)  
{  
    ...  
    break;  
    default: lista de comandos;  
}
```

- O comando **default:** é executado quando o valor da variável não coincidir com nenhum dos especificados nos *cases*

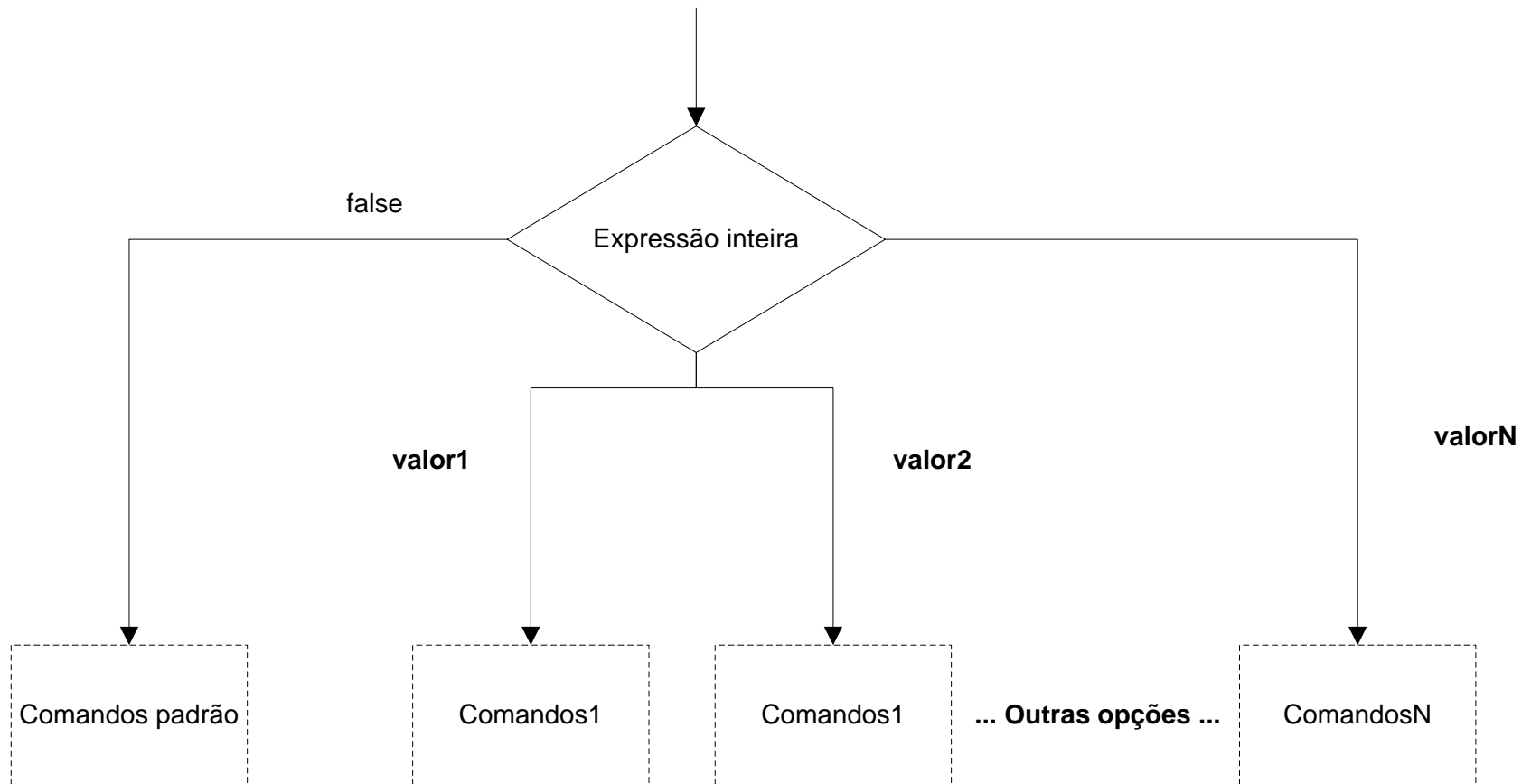


# Exemplo

```
int main()
{
    int valor;
    printf("Digite um valor: ");
    scanf("%d", &valor);
    switch(valor)
    {
        case 1:
            printf("Valor digitado: UM");
            break;
        case 2:
            printf("Valor digitado: DOIS");
            break;
        case 3:
            printf("Valor digitado: TRES");
            break;
        default:
            printf("Valor digitado diferente de UM, DOIS e TRES");
    }
    printf("\nFim do programa!");
    return 0;
}
```

# Estrutura Condicional

## 3.2- Estrutura CASE (Fluxograma)



# Estrutura Condicional

---

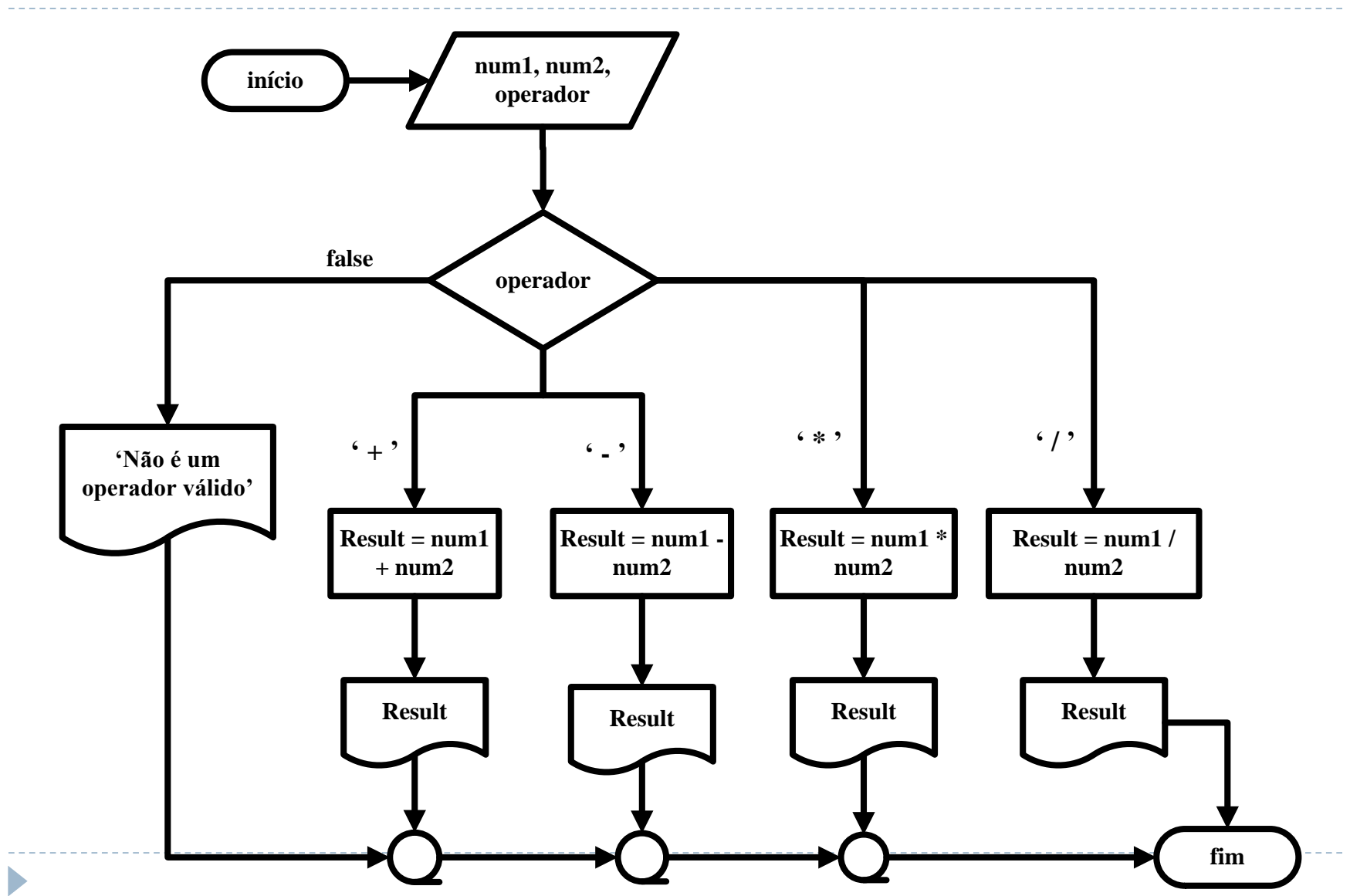
## Exemplo - Estrutura **CASE** (Fluxograma)

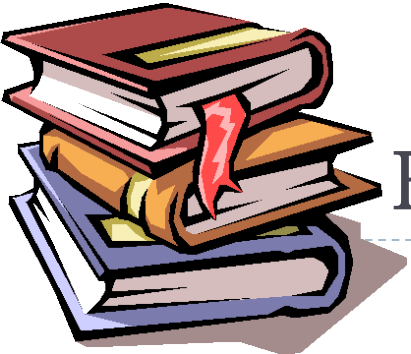
- Como um exemplo para a Estrutura CASE, considere o seguinte problema:
  - Elaborar um fluxograma que simule uma calculadora simples que **some, subtraia, multiplique e divida** dois números





# Exemplo - Estrutura CASE (Fluxograma)





## Referência Bibliográfica

---

- ▶ ASCENCIO, Ana Fernanda Gomes e CAMPOS, Edilene A. Veneruchi. **Fundamentos da Programação de Computadores – Algoritmos, Pascal e C/C++**. 2ª Edição - São Paulo: Pearson Prentice Hall, 2007. Capítulo 4.
- ▶ SOUZA, A. Furlan; GOMES, Marcelo Marques; SOARES, Marcio Vieira e CONCILIO, Ricardo. **Algoritmos e Lógica de Programação**. 2ª ed. Ver. e ampl. São Paulo: Cengage Learning 2011. Capítulo 4.
- ▶ MIZRAHI, Victorine Viviane. **Treinamento em linguagem C**. São Paulo: Pearson Prentice Hall, 2008. 2ª edição. Curso Completo.

