

# LÓGICA DE PROGRAMAÇÃO

# Lógica

- Lógica é a técnica de encadear pensamentos para atingir determinado objetivo.

# Instruções

- Instruções são um conjunto de regras ou normas definidas para a realização ou emprego de algo. Em informática, é o que indica a um computador uma ação elementar a executar.

# Seqüência Lógica e Algoritmo

- Seqüência Lógica são passos executados até atingir um objetivo ou solução de um problema.
- Um algoritmo é formalmente uma seqüência finita de passos que levam a execução de uma tarefa. Podemos pensar em algoritmo como uma receita, uma seqüência de instruções que dão cabo de uma meta específica.

# Exemplo de algoritmo

- “Chupar uma bala”
  - Pegar a bala
  - Retirar o papel
  - Jogar o papel no lixo
  - Chupar a bala
- Pode-se escrever este algoritmo de outra forma?
- Como ficaria se o pote de balas tivesse balas de diversos sabores?

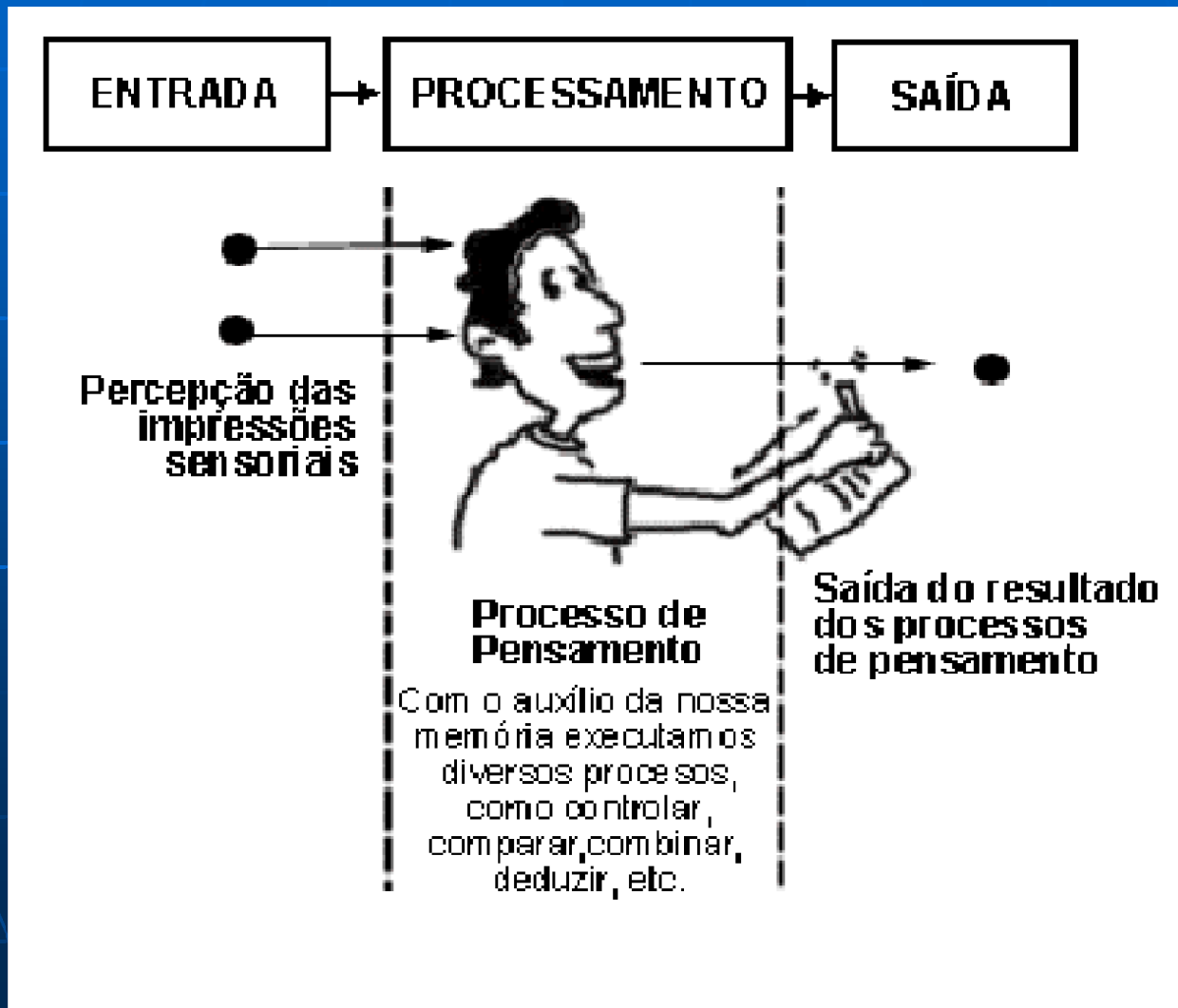
# Técnicas para algoritmos

- Para escrever um algoritmo precisa-se descrever a seqüência de instruções de maneira simples e objetiva. Para isso procura-se utilizar algumas técnicas:
  - Usar somente um verbo por frase
  - Imaginar que você está desenvolvendo um algoritmo para pessoas que não trabalham com informática
  - Usar frases curtas e simples
  - Ser objetivo
  - Procurar usar palavras que não tenham sentido dúbio

# Etapas de um algoritmo

- Normalmente um algoritmo possui três etapas distintas:
  - ENTRADA: São os dados de entrada do algoritmo
  - PROCESSAMENTO: São os procedimentos utilizados para chegar ao resultado final
  - SAÍDA: São os dados já processados

# Etapas de um algoritmo



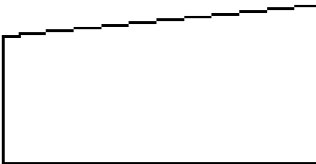
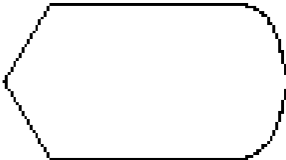




# Diagrama de blocos

- O diagrama de blocos é uma forma padronizada e eficaz para representar os passos lógicos de um determinado processamento.
- Com o diagrama podemos definir uma seqüência de símbolos, com significado bem definido, portanto, sua principal função é a de facilitar a visualização dos passos de um processamento.

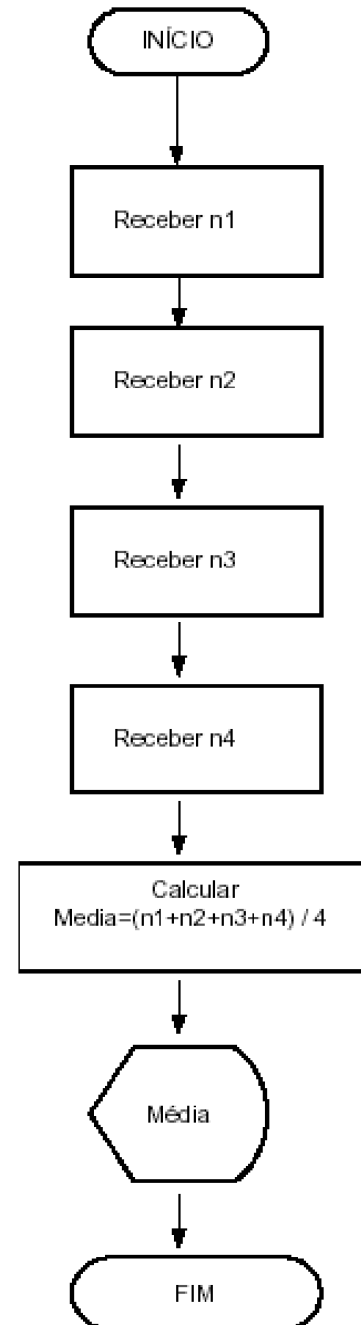
# Diagrama de blocos

Símbolo	Função
 TERMINAL	Indica o INÍCIO ou FIM de um processamento  Exemplo: Início do algoritmo
 PROCESSAMENTO	Processamento em geral  Exemplo: Calculo de dois números
 ENTRADA DE DADO MANUAL	Indica entrada de dados através do Teclado  Exemplo: Digite a nota da prova 1
 EXIBIR	Mostra informações ou resultados  Exemplo: Mostre o resultado do calculo

### “CHUPAR UMA BALA”



### “CALCULAR A MÉDIA DE 4 NOTAS”

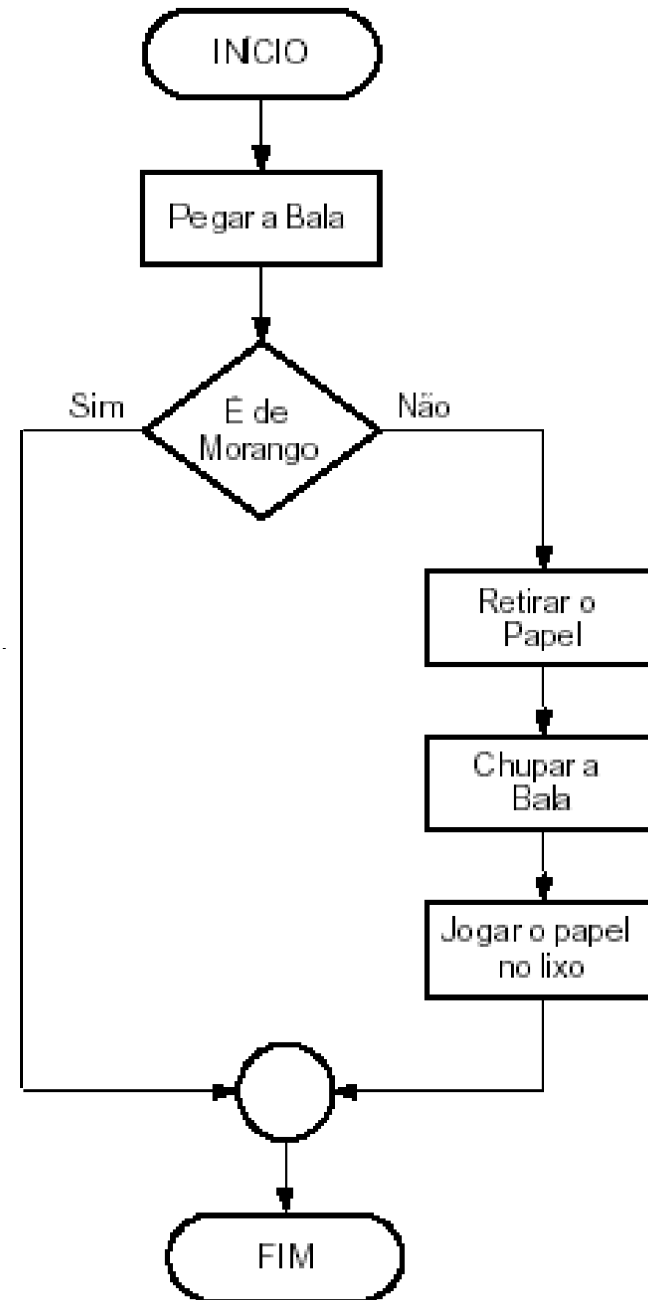


# Operação Lógica

- Operações Lógicas são utilizadas quando se torna necessário tomar decisões em um algoritmo.
- Num algoritmo, toda decisão terá sempre como resposta o resultado VERDADEIRO ou FALSO.

# Operações Lógicas

- “Chupar uma bala” (não gosto de morango)
- Pegar a bala
- A bala é de morango?
- Se **sim**, não chupe a bala
- Se **não**, continue com o algoritmo
- Retirar o papel
- Chupar a bala
- Jogar o papel no lixo



# Estruturas de Decisão

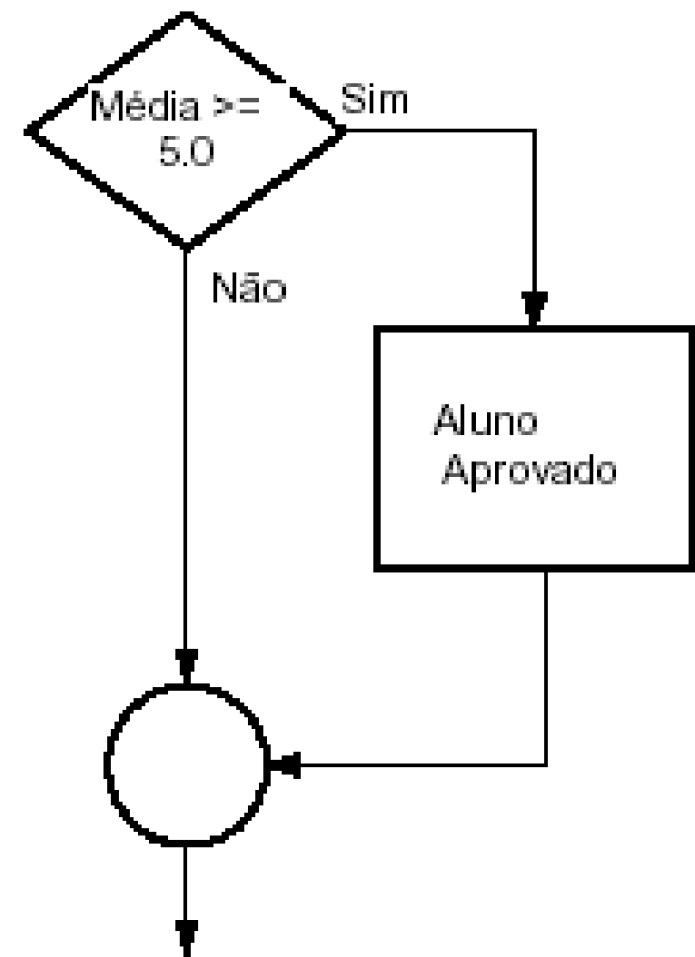
- SE ... ENTÃO / IF ... THEN

- Exemplo:

- Se Media Maior ou Igual a 5.0  
Então Aluno Aprovado

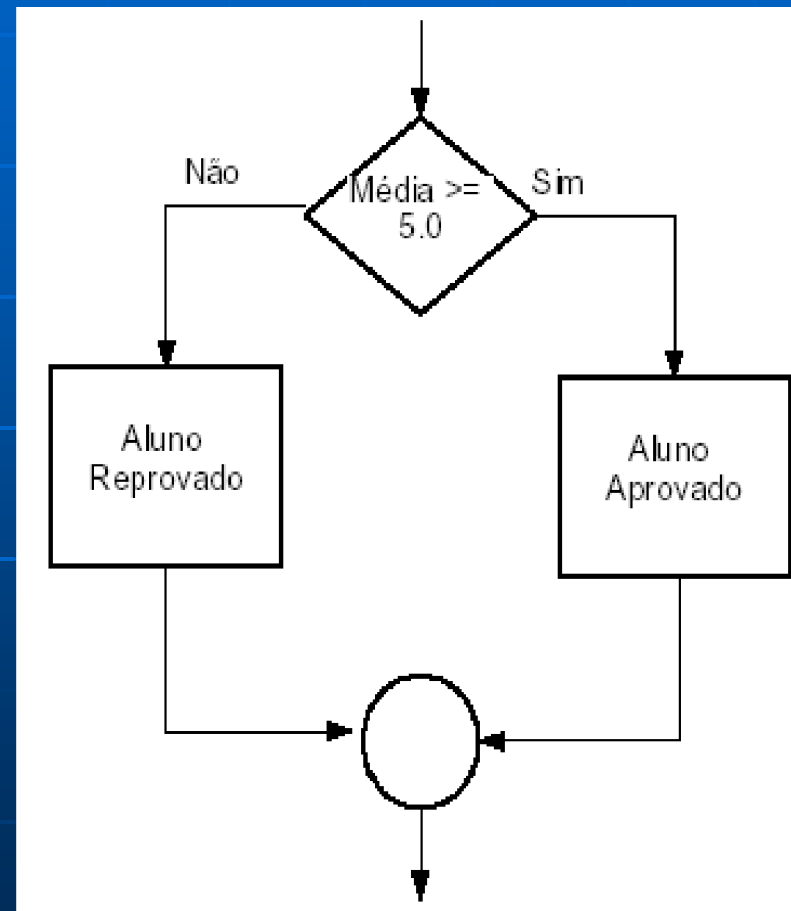
- Visual Basic

```
IF MEDIA >= 5 Then  
    Text1 = "APROVADO"  
ENDIF
```



# Estruturas de Decisão

- SE ... ENTÃO ... SENÃO /  
IF ... THEN ... ELSE
- Exemplo:
  - Se Media Maior ou Igual a 5.0  
Então Aluno Aprovado  
Senão Aluno Reprovado
  - Visual Basic  
IF MEDIA >= 5 Then  
Text1 = "APROVADO"  
ELSE  
Text1 = "REPROVADO"  
ENDIF



# Estruturas de Decisão

- Exercício: escreva o diagrama de blocos e o código Visual Basic do algoritmo abaixo:
  - Se a média for menor que 5.0, então, o aluno estará reprovado.
  - Senão, se a média for menor que 7.0, então, o aluno deve realizar nova prova.
  - Senão, o aluno estará aprovado.



# Estruturas de Decisão

## ■ SELECIONE ... CASO / SELECT ... CASE

### ■ Exemplo:

- Entre com número da opção (OP).
- Selecione: caso OP igual a 1, título "opção 1"; caso OP igual a 2, título "opção 2"; caso OP igual a 3, título "opção 3"; caso OP igual a 4, título "opção 4"; caso OP igual a 5, título "opção 5"; senão título igual a "opção errada".

- Visual Basic

TITULO = ""

OP = INPUTBOX("DIGITE A OPÇÃO")

SELECT CASE OP

CASE 1

TITULO = "OPÇÃO 1"

CASE 2

TITULO = "OPÇÃO 2"

CASE 3

TITULO = "OPÇÃO 3"

CASE 4

TITULO = "OPÇÃO 4"

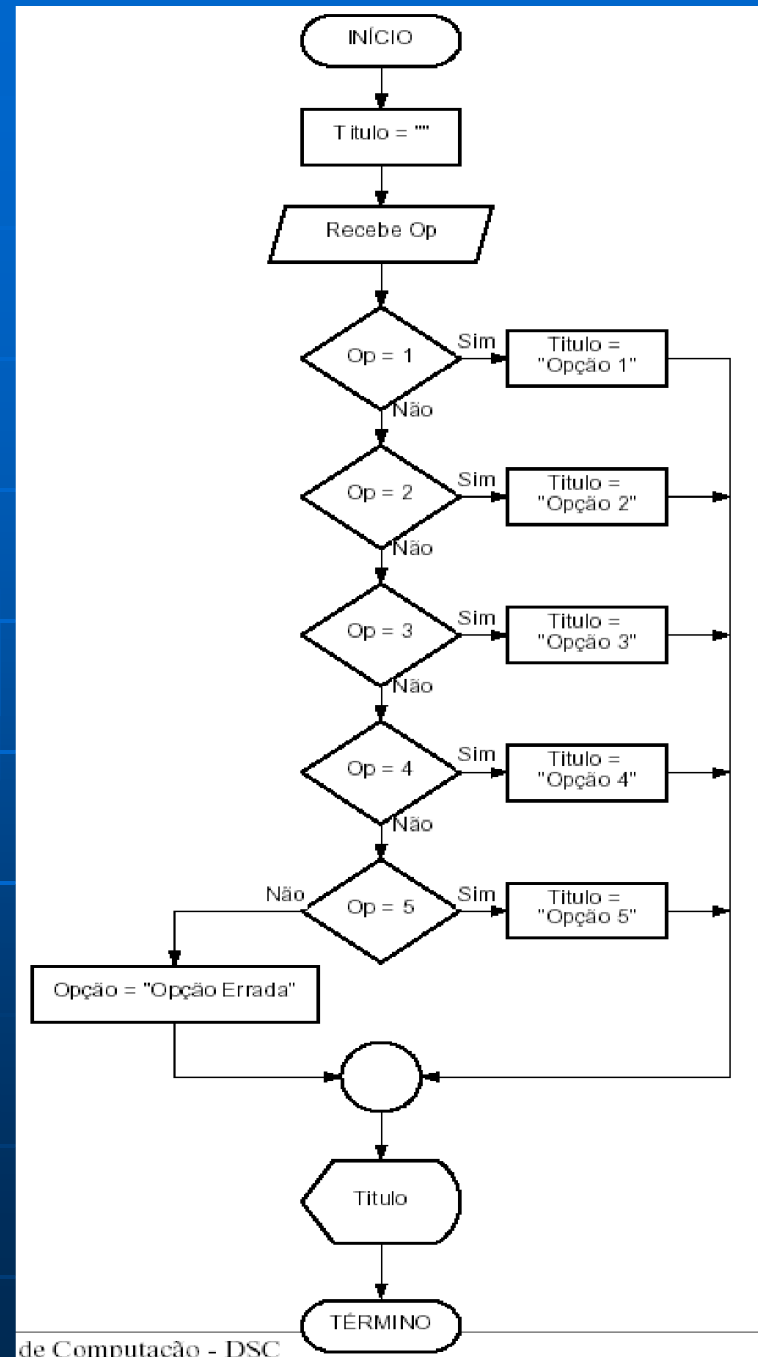
CASE 5

TITULO = "OPÇÃO 5"

CASE ELSE

TITULO = "OPÇÃO ERRADA"

END SELECT



# Estruturas de Repetição

- Enquanto x, processar ... (Do While ... Loop);
- Até que x, processar ... (Do Until ... Loop);
- Processar ..., Enquanto x (Do ... Loop While);
- Processar..., Até que x (Do ... Loop Until);
- Para ... Até ... Seguinte (For ... To ... Next)

# Estruturas de Repetição

Enquanto x, processar ...  
(Do While ... Loop)

## ■ Exemplo:

- Número igual a 0
- Enquanto número menor ou igual a 100, some mais um ao número

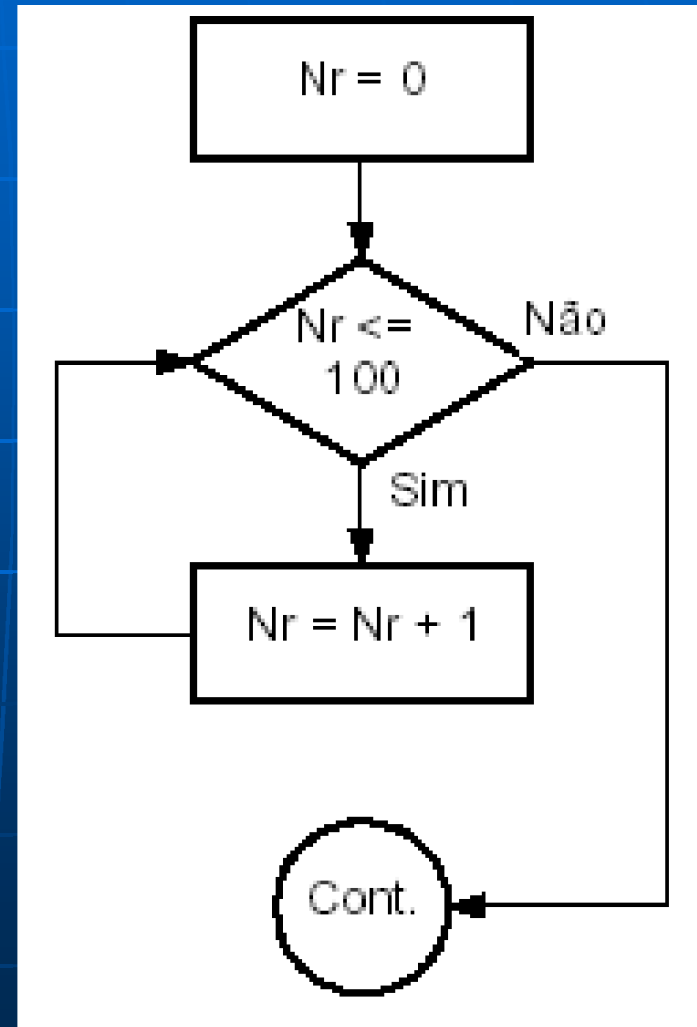
- Visual Basic

Nr = 0

Do While Nr <= 100

    Nr = Nr + 1

Loop



# Estruturas de Repetição

Até que x, processar ...  
(Do Until ... Loop)

## ■ Exemplo:

- Número igual a 0
- Até que número seja igual a 100, some mais um ao número

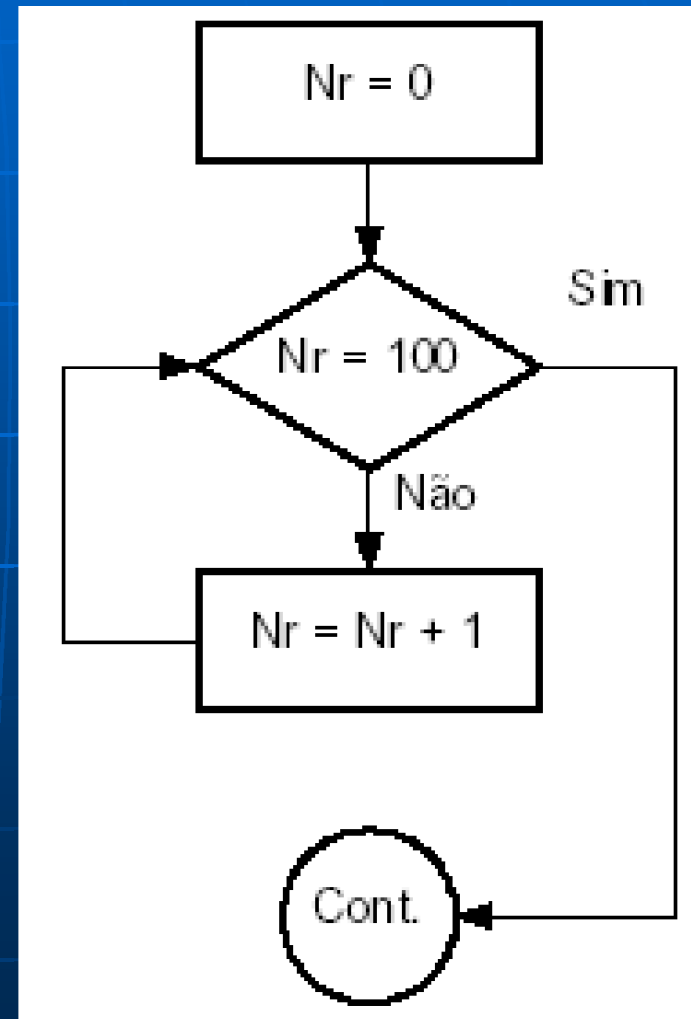
- Visual Basic

Nr = 0

Do Until Nr = 100

    Nr = Nr + 1

Loop



# Estruturas de Repetição

Processar ..., Até que x  
(Do ... Loop While)

- Exemplo:

- Número igual a 0
- Some mais um ao número; repita enquanto número menor ou igual a 100

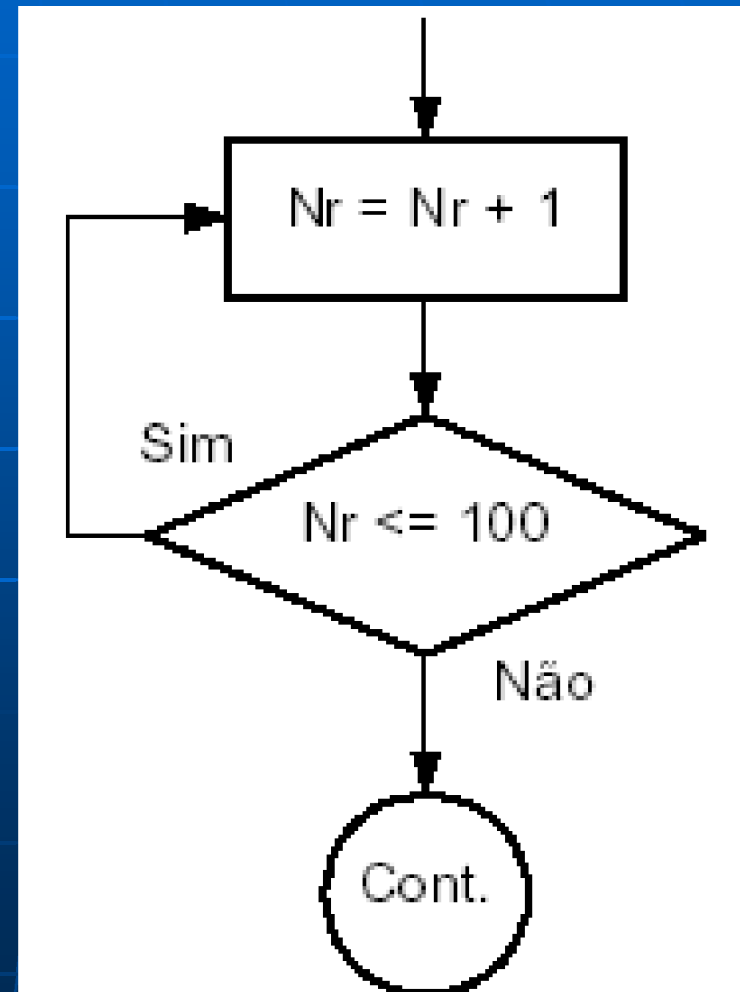
- Visual Basic

Nr = 0

Do

    Nr = Nr + 1

Loop While Nr <= 100



# Estruturas de Repetição

Processar ..., Enquanto x  
(Do ... Loop Until)

## ■ Exemplo:

- Número igual a 0
- Some mais um ao número;  
repita até que o número seja  
igual a 100

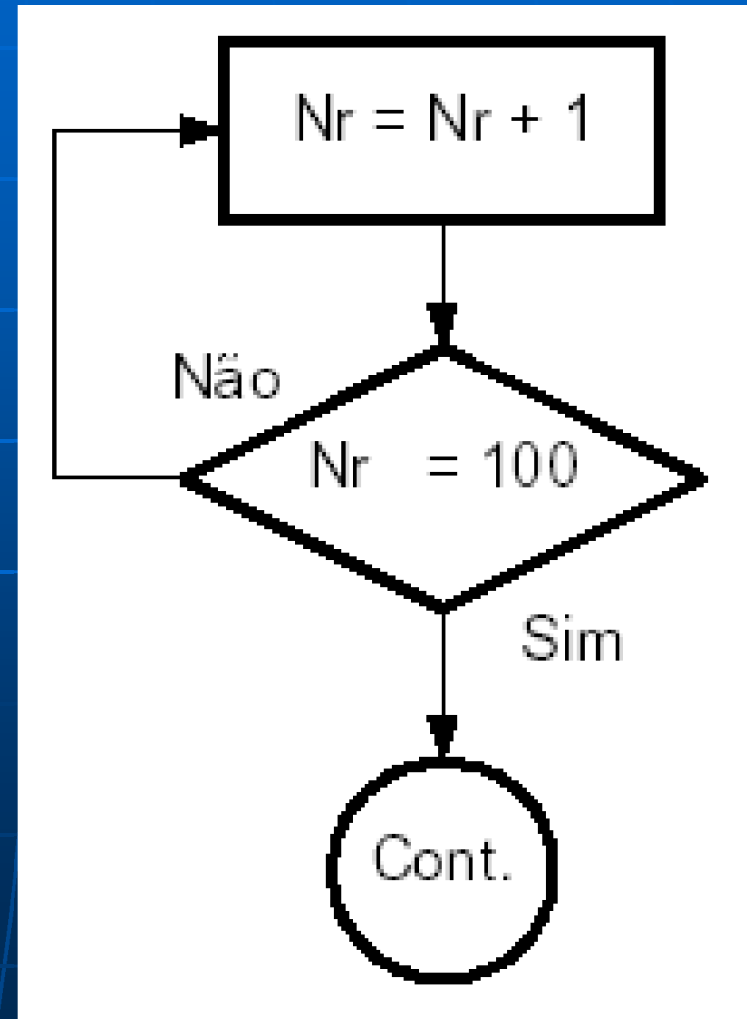
- Visual Basic

Nr = 0

Do

Nr = Nr + 1

Loop Until Nr = 100



# Estruturas de Repetição

Para ... Até ... Seguinte  
(For ... To ... Next)

- Exemplo:

- Número igual a 0
- Para número de 0 até 100, some um ao número

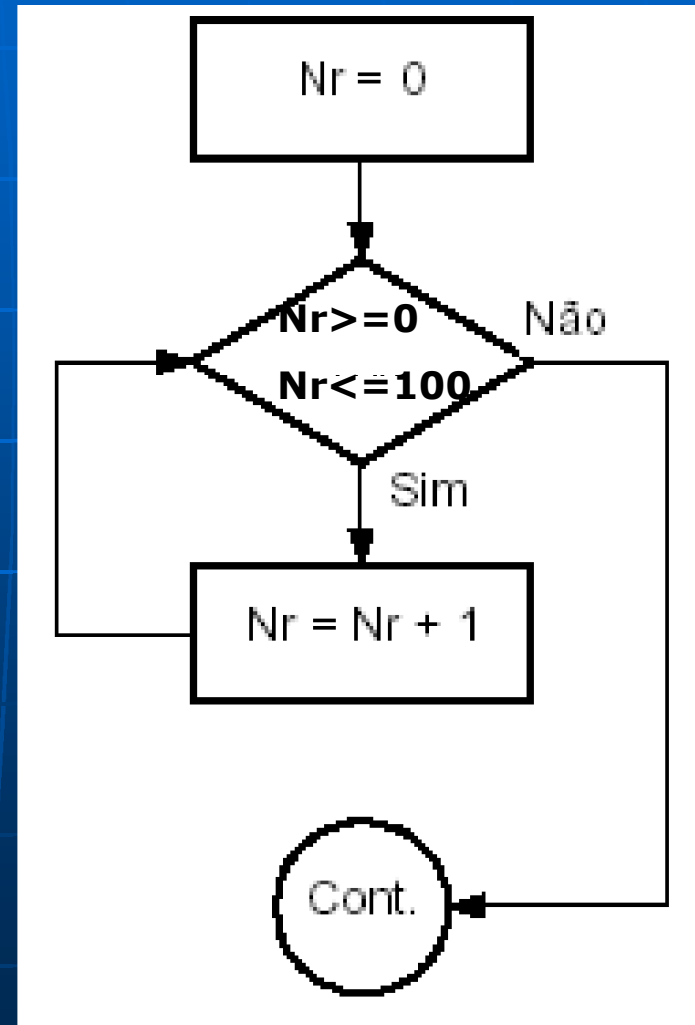
- Visual Basic

Nr = 0

For Nr = 0 to 100

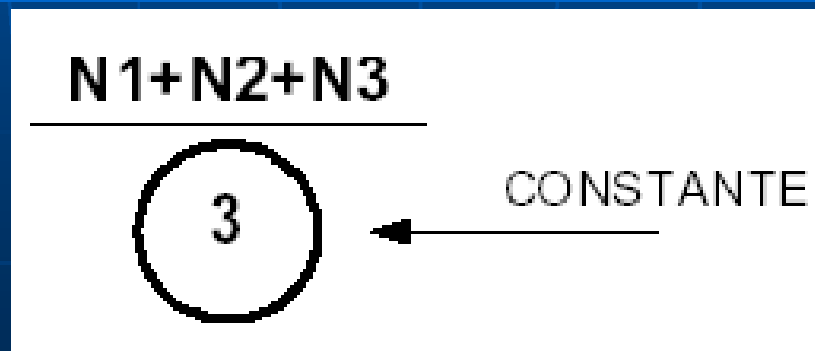
    Nr = Nr + 1

Next



# Constante

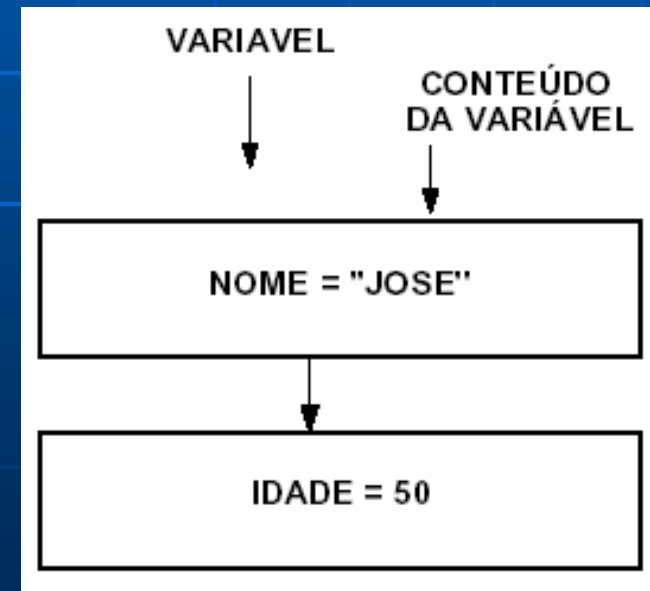
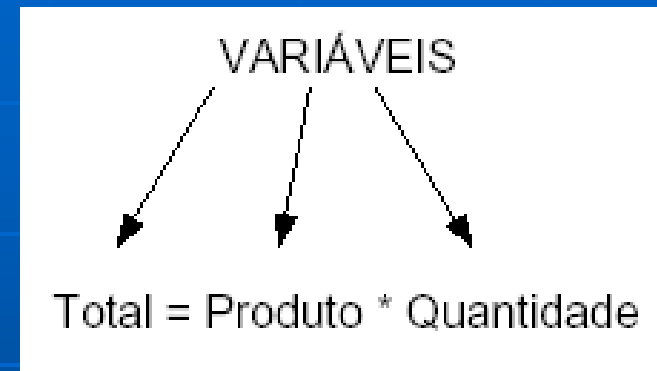
- Constante é um determinado valor fixo que não se modifica ao longo do tempo, durante a execução de um programa. Conforme o seu tipo, a constante é classificada como sendo numérica, lógica e literal.





# Variável

- Variável é a representação simbólica dos elementos de um certo conjunto. Cada variável corresponde a uma posição de memória, cujo conteúdo pode se alterar ao longo do tempo durante a execução de um programa. Embora uma variável possa assumir diferentes valores, ela só pode armazenar um valor a cada instante



# Tipos de variáveis

- **Numéricas:** Específicas para armazenamento de números, que posteriormente poderão ser utilizados para cálculos. Podem ser ainda classificadas como Inteiras ou Reais.
- **Alfanuméricas:** Para dados que contenham letras e/ou números. Pode em determinados momentos conter somente dados numéricos ou somente literais. Se usado somente para armazenamento de números, não poderá ser utilizada para operações matemáticas.
- **Lógicas:** Armazenam somente dados lógicos que podem ser Verdadeiro ou Falso.

# Operadores

- Os operadores são meios pelo qual são incrementados, decrementados, comparados e avaliados dados dentro do computador. Temos três tipos de operadores:
  - Operadores Aritméticos
  - Operadores Relacionais
  - Operadores Lógicos

# Operadores Aritméticos

- Os operadores aritméticos são os utilizados para obter resultados numéricos.

OPERAÇÃO	SÍMBOLO
Adição	+
Subtração	-
Multiplicação	*
Divisão	/
Exponenciação	**

# Operadores Aritméticos

- Hierarquia das operações aritméticas:

- 1<sup>o</sup> ( ) Parênteses
- 2<sup>o</sup> Exponenciação
- 3<sup>o</sup> Multiplicação, divisão (o que aparecer primeiro)
- 4<sup>o</sup> + ou - (o que aparecer primeiro)
  
- $1 + 7 * 2 ** 2 - 1 = ?$
- $3 * (1 - 2) + 4 * 2 = ?$

# Operadores Relacionais

- Os operadores relacionais são utilizados para comparar ***String*** de caracteres e números. Os valores a serem comparados podem ser caracteres ou variáveis.
- Estes operadores sempre retornam valores lógicos (verdadeiro ou falso / True ou False).

Descrição	Símbolo
Igual a	=
Diferente de	<> ou #
Maior que	>
Menor que	<
Maior ou igual a	>=
Menor ou igual a	<=

# Operadores Relacionais

- Se  $A = 5$  e  $B = 3$  então:
- $A=B$  (falso ou verdadeiro?)
- $A<>B$  (falso ou verdadeiro?)
- $A>B$  (falso ou verdadeiro?)
- $A<B$  (falso ou verdadeiro?)
- $A>=B$  (falso ou verdadeiro?)
- $A<=B$  (falso ou verdadeiro?)

# Operadores Lógicos

- Os operadores lógicos servem para combinar resultados de expressões, retornando se o resultado final é verdadeiro ou falso.
  - E / AND: Uma expressão AND (E) é verdadeira se todas as condições forem verdadeiras
  - OU / OR: Uma expressão OR (OU) é verdadeira se pelo menos uma condição for verdadeira
  - NÃO / NOT: Um expressão NOT (NÃO) inverte o valor da expressão ou condição, se verdadeira inverte para falsa e vice-versa.



# Operadores Lógicos

1º Valor	Operador	2º Valor	Resultado
T	AND	T	
T	AND	F	
F	AND	T	
F	AND	F	
T	OR	T	
T	OR	F	
F	OR	T	
F	OR	F	
T	NOT		
F	NOT		

# Operadores Lógicos

- Considerando os valores das três variáveis, indique se é Falsa ou Verdadeira a expressão.  $A = 5$ ;  $B = 8$ ;  $C = 1$

- $A = B \text{ AND } B > C$
- $A < > B \text{ OR } B < C$
- $A > B \text{ NOT}$
- $A < B \text{ AND } B > C$
- $A > = B \text{ OR } B = C$
- $A < = B \text{ NOT}$