

The background features a collection of colorful 3D blocks in shades of purple, blue, yellow, and green, arranged in a scattered pattern. A faint, light-colored path or line winds through the blocks, starting from the bottom left and extending towards the top right. The overall aesthetic is clean and modern, with a focus on geometric shapes and a soft color palette.

Leitura complementar

# Variáveis, Operadores e Funções

- ✓ Utilização de variáveis;
- ✓ Tipos de variáveis;
- ✓ Nome de variáveis;
- ✓ Declaração de variáveis;
- ✓ Comando de atribuição;
- ✓ Constantes;
- ✓ Operadores aritméticos;
- ✓ Operadores relacionais;
- ✓ Operadores lógicos;
- ✓ Função;
- ✓ Concatenação de alfanuméricos.

# Introdução

**Variáveis** são áreas na memória, utilizadas em programação, que servem para armazenar dados. O conteúdo de uma variável pode ser alterado, mas uma variável só pode conter um dado por vez.

As áreas na memória são divisões na memória. O computador identifica cada divisão por meio de um endereço no formato hexadecimal, ou seja, para facilitar a localização dos dados, as variáveis são encontradas pelos endereços de memória, assim como uma casa é encontrada pelo seu endereço. Utilizamos variáveis frequentemente em programação.

## Utilizando variáveis

A seguir, apresentamos as situações em que utilizamos variáveis.

### Consistência de condições

Com as variáveis podemos verificar a veracidade, ou não, de uma condição para assim obtermos um ou outro resultado. Se, entre várias opções, desejamos escolher uma alternativa, podemos comparar a escolha com uma variável previamente definida e fazer o programa executar uma ou outra determinada sequência de comandos.

### Controle de repetições

As variáveis de memória podem ser usadas para o controle de repetições. Se escrevermos uma sequência de comandos dentro de um programa e desejarmos que esta sequência seja repetida um certo número de vezes, podemos utilizar uma variável numérica e, a cada passagem, aumentar ou diminuir seu valor, de modo que, quando atingido um determinado valor, a sequência de comandos pare de ser executada.

## Comparações de variáveis de memória com campos de registros

Quando trabalhamos com arquivos de banco de dados, podemos fazer comparações, trocas e procuras de registros através das variáveis de memória.

As variáveis ficam armazenadas na memória RAM do computador, enquanto **campos** de um **registro** pertencem ao banco de dados correspondente e, portanto, estão gravados no disco (ou em outro meio de armazenamento) do computador.

## Tipos de variáveis

Os tipos e valores dos conteúdos das variáveis podem variar de linguagem para linguagem. Vejamos:

- **Alfanumérica:** Atribuímos qualquer tipo de caractere a elas, ou seja, letras, números ou sinais;
- **Numérica:** Atribuímos somente números a elas, pois podemos usá-las para efetuar cálculos;
- **Data:** Atribuímos somente datas a elas;
- **Lógica:** Atribuímos somente valores verdadeiros ou falsos (V/F). São utilizadas para testes lógicos;
- **Objeto:** Atribuímos uma referência a um objeto.

# Nomes de variáveis

Para atribuir um nome a uma variável, devemos observar alguns aspectos:

- Não é possível começar um nome de variável com número;

Certo	Errado
X Teste ABC2 J34CD Texto3 VAR1 VNome	1ABC 2Parcela 3Fase

- Só é permitido o uso de underline ( \_ ) no nome. Espaço ou qualquer outro sinal é proibido;

Certo	Errado
Salariofixo Nota_Final Valor_1	Tudo% Qualquer\$valor Valor da Compra

- Para maior facilidade, devemos usar sempre nomes autoexplicativos;
- Devemos, no entanto, cuidar para que as variáveis não tenham nomes de comandos, funções ou campos de um banco de dados, pois isto pode causar problemas durante a execução do programa.

## Declaração de variáveis

As variáveis precisam ser **declaradas**. Para declarar uma variável, devemos informar o nome e o tipo da variável.

Cada um dos tipos de variáveis ocupa espaços diferentes de memória, de acordo com os dados que irão armazenar. A definição do tipo de variável otimiza a utilização do espaço de memória.

Normalmente a variável é declarada no início do programa para que possa ser utilizada no programa inteiro, sendo assim, a declaramos no início do algoritmo.

Veja, a seguir, exemplos de como iremos declarar as variáveis nos algoritmos:

**Declara A, B, C numéricas, D, E alfanuméricas**

**Declara VALORX numérica, TEXTTOY alfanumérica**

**Declara NOTA\_BIMESTRAL1, NOTA\_BIMESTRAL2 numéricas**

**Declara MENSAGEM5, MENSAGEM9 alfanuméricas, HOJE data**

## Comando de atribuição

O **comando de atribuição** serve para armazenar um valor numa variável. As variáveis numéricas que serão utilizadas no programa e não têm um valor inicial predefinido geralmente recebem o valor zero (0).

Utilizaremos o sinal de igual (=) para representar o comando de atribuição.

Onde tiver =, lê-se **recebe**:

**A = 10**

**X = 0**

**Valor = 0**

**Nota = 7,5**

**Nome\_Aluno = "Leandro"**

**MensagemFinal = "Boa noite."**

# Constantes

Ao contrário das variáveis, as **constantes** possuem valor fixo e não sofrem alteração durante o processamento.

# Operadores aritméticos

Em programação, usamos os mesmos **operadores aritméticos** que são usados na matemática, e com as mesmas prioridades.

Em primeiro lugar são executadas as exponenciações e as radiciações, em segundo as multiplicações e as divisões, e depois as adições e as subtrações.

OPERAÇÃO	OPERADOR USADO	PRIORIDADE
Radiciação	//	1º
Exponenciação	^ ou **	1º
Multiplicação	*	2º
Divisão	/	2º
Adição	+	3º
Subtração	-	3º

Usamos também os seguintes operadores para operações matemáticas não-convencionais:

OPERAÇÃO	OPERADOR USADO	PRIORIDADE
Resto da divisão	Mod	2º
Quociente da divisão inteira	Div	2º

Normalmente, as operações são executadas da esquerda para a direita, a não ser que mudemos sua ordem usando parênteses.

Para a construção de algoritmos que contêm fórmulas matemáticas, todas as expressões aritméticas devem ser linearizadas, ou seja, colocadas em linha.

Veja, a seguir, exemplos de como é uma fórmula na matemática tradicional e como ela fica linearizada.

- Exemplo 1

$$2 \times \frac{5 + 3}{2} = 2 * (5 + 3) / 2$$

- Exemplo 2

$$4 + 5^2 - (2 \times 3) = 4 + 5^2 - 2 * 3$$

- Exemplo 3

$$\sqrt{25} - \sqrt[3]{125} + 5 = 25 // 2 - 125 // 3 + 5$$

- Exemplo 4

$$3 \times 4 + 5 - \frac{125}{5^3} = 3 * 4 + 5 - 125 / 5^{**3}$$

- Exemplo 5

$$3 \times \left( (4 + 5) - \left( \frac{25}{5} \right) \right)^2 = 3 * ((4+5)-(25/5)) ** 2$$

Veja, a seguir, o valor de cada variável com a resolução das fórmulas matemáticas respeitando a prioridade de execução dos operadores.

- **Exemplo 1**

$$A = 2 * (5 + 3) / 2$$

$$A = 2 * 8 / 2$$

$$A = 16 / 2$$

$$A = 8$$

- **Exemplo 2**

$$B = 4 + 5 \wedge 2 - 2 * 3$$

$$B = 4 + 25 - 2 * 3$$

$$B = 4 + 25 - 6$$

$$B = 29 - 6$$

$$B = 23$$

- **Exemplo 3**

$$C = 25 // 2 - 125 // 3 + 5$$

$$C = 5 - 125 // 3 + 5$$

$$C = 5 - 5 + 5$$

$$C = 5$$

- **Exemplo 4**

$$D = 3 * 4 + 5 - 125 / 5 ** 3$$

$$D = 3 * 4 + 5 - 125 / 125$$

$$D = 12 + 5 - 125 / 125$$

$$D = 12 + 5 - 1$$

$$D = 17 - 1$$

$$D = 16$$



- **Exemplo 5**

$$E = 3 * ((4 + 5) - (25/5)) ** 2$$

$$E = 3 * (9 - (25/5)) ** 2$$

$$E = 3 * (9 - 5) ** 2$$

$$E = 3 * 4 ** 2$$

$$E = 3 * 16$$

$$E = 48$$

- **Exemplo 6**

$$F = 7 / 3 \quad \rightarrow \quad F = 2,3333...3$$

$$G = 7 \text{ DIV } 3 \quad \rightarrow \quad G = 2$$

$$H = 7 \text{ MOD } 3 \quad \rightarrow \quad H = 1$$

$$7 \text{ MOD } 3 \quad \leftarrow \quad \begin{array}{r} 7 \overline{) 3} \\ 1 \quad 2 \end{array} \quad \rightarrow \quad 7 \text{ DIV } 3$$

Portanto:

- **7 MOD 3** mostra como resultado o número 1;
- **7 DIV 3** mostra como resultado o número 2.

- Exemplo 7

J = 1234 / 10 → J = 123,4  
K = 1234 / 100 → K = 12,34  
L = 1234 / 1000 → L = 1,234

M = 1234 DIV 10 → M = 123  
N = 1234 DIV 100 → N = 12  
P = 1234 DIV 1000 → P = 1

Q = 1234 MOD 10 → Q = 4  
R = 1234 MOD 100 → R = 34  
S = 1234 MOD 1000 → S = 234

T = 1234 DIV 10 DIV 10 → T = 12  
U = 1234 MOD 10 \* 10 → U = 40

V = 1234 MOD 1000 DIV 100 → V = 2  
W = 1234 DIV 100 / 6 DIV 2 → W = 1

- Exemplo 8

NRO = 457  
UNIDADE = NRO MOD 10 → UNIDADE = 7  
DEZENA = NRO DIV 10 MOD 10 → DEZENA = 5  
CENTENA = NRO DIV 100 → CENTENA = 4

## Contadores e acumuladores

- **Contador**

Um **contador** é construído a partir de uma variável que recebe o valor dela mesma mais outro valor.

A linha que descreve um contador **A** é:

$$A = A + 1$$

Nesse exemplo a variável **A** está recebendo o valor dela mesma mais 1, ou seja, está contando de 1 em 1. O valor 1 é uma constante.

- **Acumulador**

Um **acumulador** é uma variável que recebe o valor dela mesma mais o valor de outra variável. A linha que descreve um acumulador **B** de **A** é:

$$B = B + A$$

Nesse exemplo, a variável **B** está recebendo o valor dela mesma mais o valor da variável **A**. A variável **A** representa o valor a ser somado, acumulado na variável **B**.

- **Acumulador de Conta**

A linha que descreve um acumulador **C** do dobro de **B** é:

$$C = C + 2 * B$$

Nesse exemplo, a variável **C** está recebendo o valor dela mesma mais duas vezes o valor da variável **B**.

# Operadores relacionais


Quando encontramos situações nas quais é necessário usar condições, adotamos os seguintes operadores para estabelecer relações:

OPERAÇÃO	OPERADOR USADO
Igual a	=
Maior que	>
Menor que	<
Maior ou igual a	> =
Menor ou igual a	< =
Diferente de	< >

- Exemplos

Num algoritmo:

Se o Cargo for igual a Gerente  
Então conceder um aumento de 10% no salário  
Senão conceder um aumento de 5% no salário  
Fim Se

 **Observação:** Note que o comando FIM SE acima foi colocado pra identificar o fim do comando SE.

Numa linguagem de programação:

IF Cargo = “Gerente”  
Salario = Salario \* 1.10  
ELSE  
Salario = Salario \* 1.05  
ENDIF

# Operadores lógicos

São usados quando existem duas ou mais condições em um teste lógico, e assim como os operadores de comparação, retornam um resultado **VERDADEIRO** ou **FALSO**:

- **NÃO (NOT) – Ordem de Prioridade: 1ª**

Operador lógico que inverte a lógica de uma expressão. Se ela for verdadeira, torna-se falsa, e vice-versa.

- **E (AND) – Ordem de Prioridade: 2ª**

Operador lógico em que a resposta da operação é verdadeira somente se as duas condições forem verdadeiras.

**Se o Cargo for igual a Gerente E a Idade for maior ou igual a 50 anos  
Então conceder um aumento de 10% no salário  
Senão conceder um aumento de 5% no salário  
Fim Se**

- **OU (OR) – Ordem de Prioridade: 3ª**

Operador lógico em que a resposta da operação é verdadeira se pelo menos uma das condições for verdadeira.

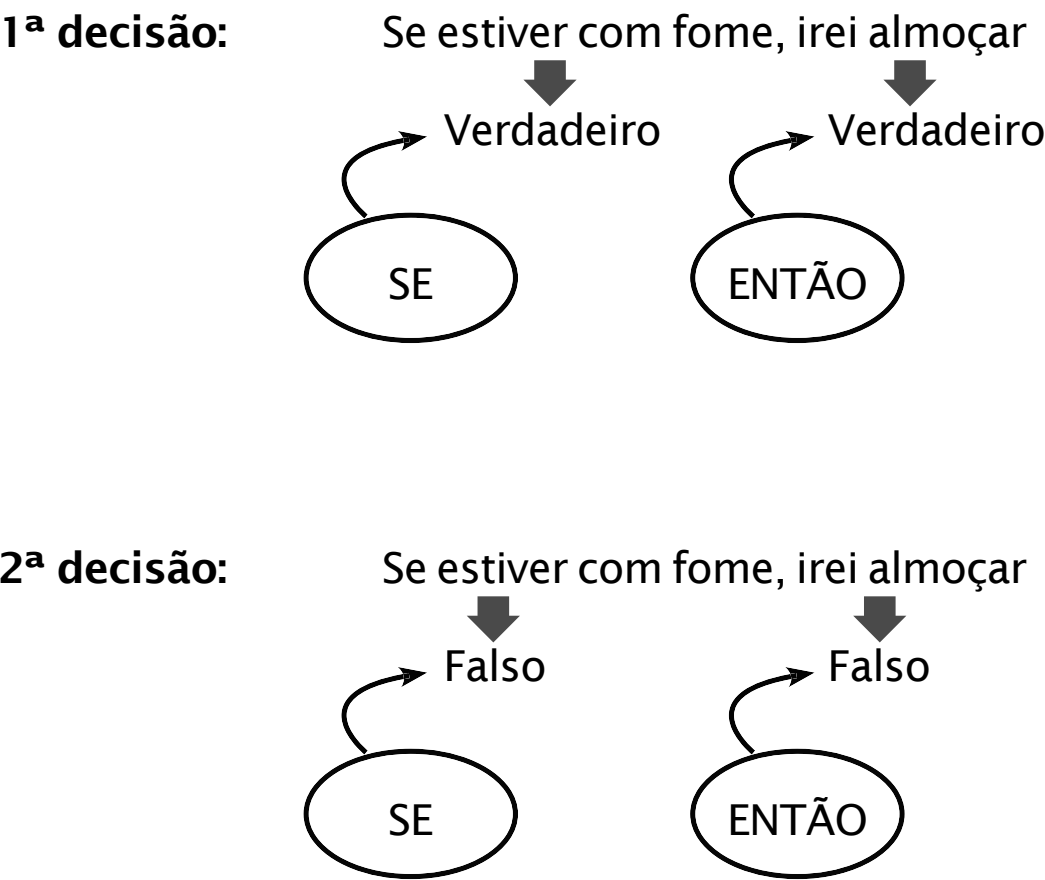
**Se o cargo for igual a Gerente OU a Idade for maior ou igual a 50 anos  
Então conceder um aumento de 10% no salário  
Senão conceder um aumento de 5% no salário  
Fim Se**

## Tabela de decisão

Partindo de uma condição, podemos tomar uma ou outra decisão.

Exemplo:

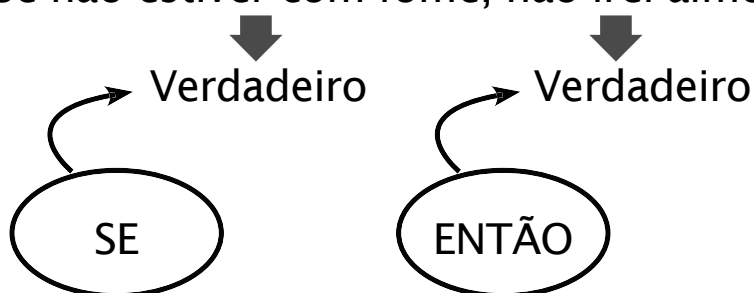
Se estiver com fome irei almoçar.  
Senão não irei almoçar.



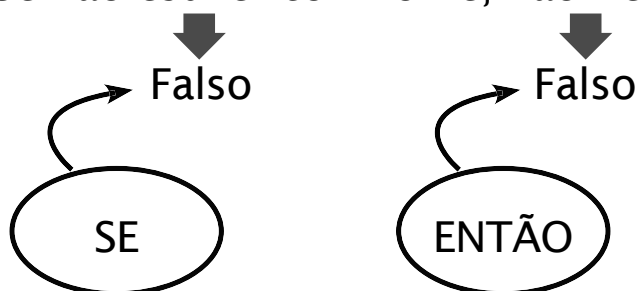
Observe que podemos escrever na negativa e o resultado será o mesmo.

**Se NÃO estiver com fome NÃO irei almoçar.  
Senão irei almoçar.**

**1ª decisão:** Se não estiver com fome, não irei almoçar



**2ª decisão:** Se não estiver com fome, não irei almoçar



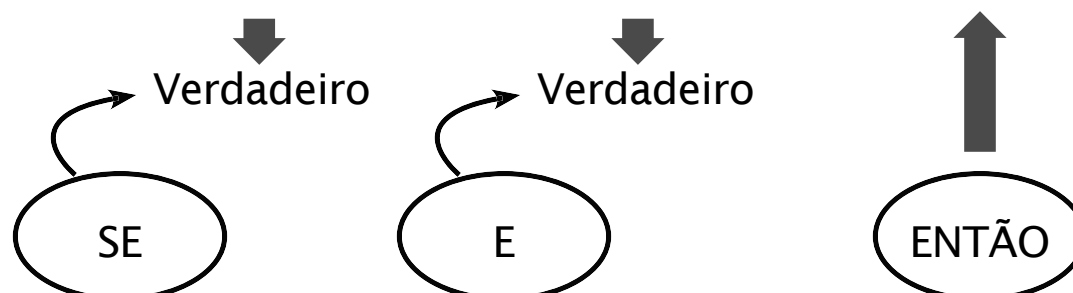
Analisamos, até agora, somente uma condição para que tomássemos uma ou outra decisão.

A **TABELA DE DECISÃO** também é conhecida como **TABELA VERDADE**.

Vamos criar uma tabela unindo duas condições, usando o operador E.

Vamos supor, então, que o almoço só é servido após o meio-dia:

Se estiver com fome e for meio-dia, irei almoçar



Somente quando as duas primeiras proposições forem verdadeiras, a terceira também será.

Se	estiver com fome	e	for meio-dia	então	irei almoçar
	V	+	V	=	V
	V	+	F	=	F
	F	+	V	=	F
	F	+	F	=	F

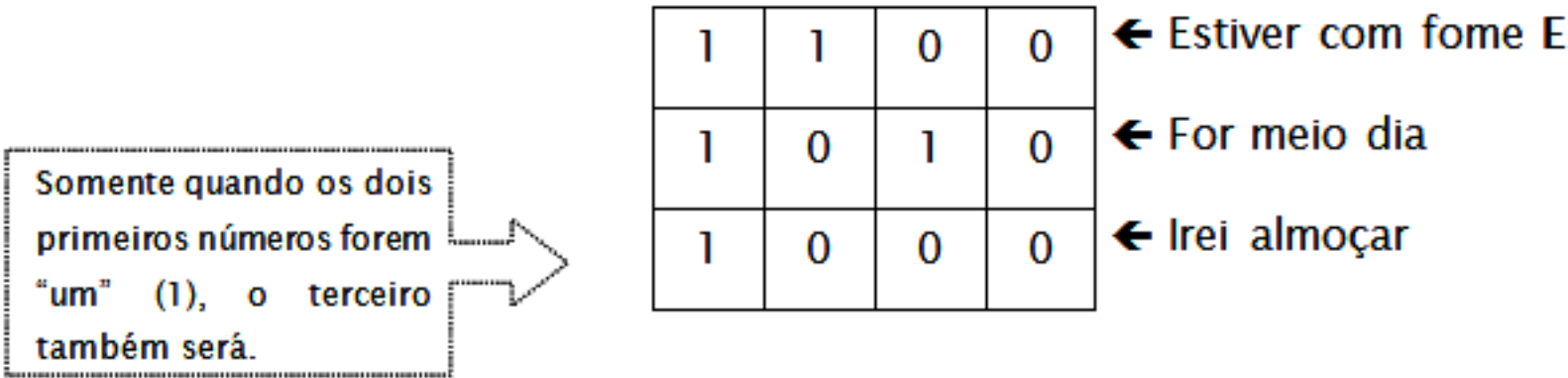
Usando o operador **OU**, basta que apenas uma das duas primeiras proposições seja verdadeira para que a terceira também seja.

Se	estiver com fome	ou	for meio-dia	então	irei almoçar
	V	+	V	=	V
	V	+	F	=	V
	F	+	V	=	V
	F	+	F	=	F

## Tabela de decisão com números binários

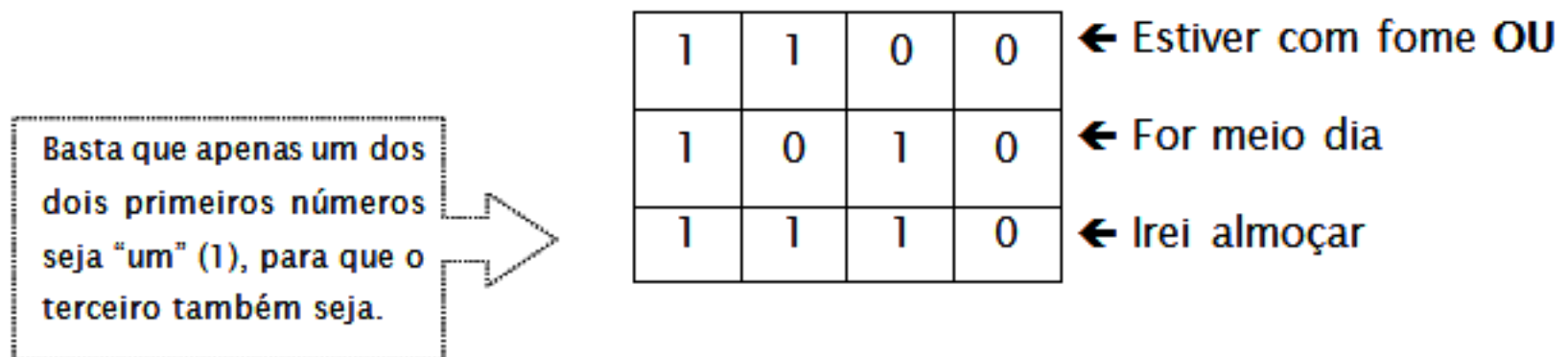
Neste caso o número zero (**0**) tem valor **FALSO** e o número um (**1**) tem valor **VERDADEIRO**.

Usando o operador **E**, somente quando as duas primeiras proposições forem verdadeiras, a terceira também será.

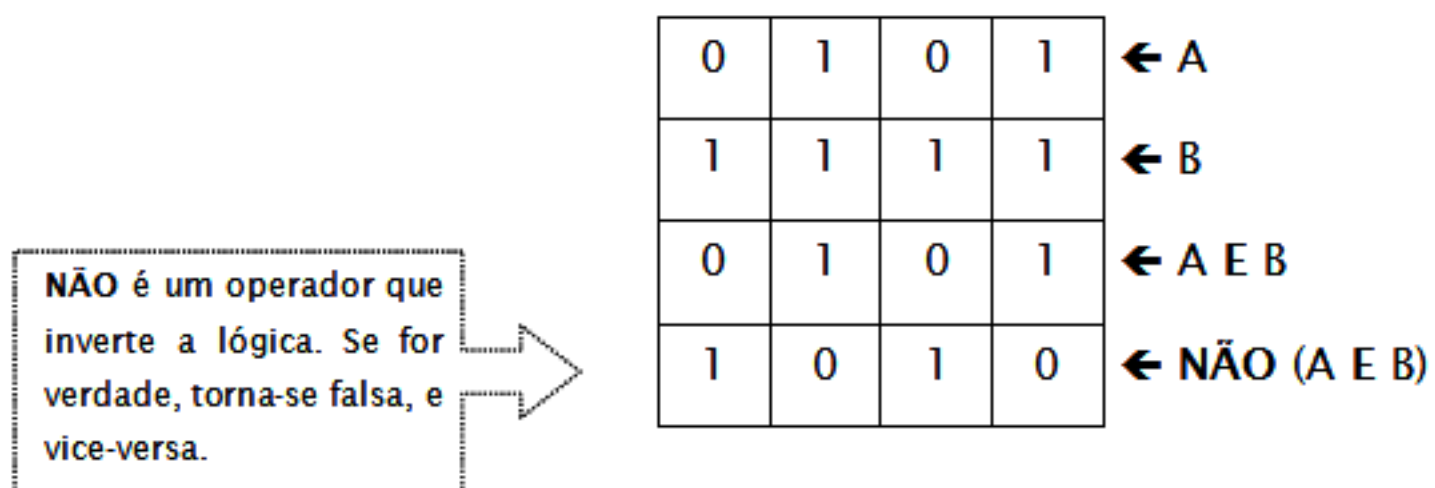




Usando o operador **OU**, basta que apenas uma das duas primeiras proposições seja verdadeira para que a terceira também seja.



Podemos efetuar operações binárias com variáveis. Como exemplo, vamos supor que **A = 0101** e **B = 1111**.



- **Exemplos:**

Dadas as variáveis:

- **Numéricas:** A = 10, B = 5, C = 2
- **Lógicas:** X = FALSO, Y = VERDADEIRO

Vejam os resultados das expressões:

- a)  $A < B \text{ E } C \geq 2$   
FALSO E VERDADEIRO  
Resposta: FALSO
- b)  $A - 5 < B \text{ E } C \leq A / 2$   
FALSO E VERDADEIRO  
Resposta: FALSO
- c)  $A < B \text{ OU } C < B$   
FALSO OU VERDADEIRO  
Resposta: VERDADEIRO
- d)  $B < A \text{ OU } C > B \text{ E } 2 * C < B$   
VERDADEIRO OU FALSO E VERDADEIRO  
VERDADEIRO OU FALSO  
Resposta: VERDADEIRO
- e)  $B < A \text{ E } C > B \text{ E } 2 * C < B$   
VERDADEIRO E FALSO E VERDADEIRO  
FALSO E VERDADEIRO  
Resposta: FALSO
- f)  $C < A \text{ E } A < C * 6$   
VERDADEIRO E VERDADEIRO  
Resposta: VERDADEIRO
- g) **NÃO**  $X \text{ E } Y$   
**NÃO** FALSO E VERDADEIRO  
VERDADEIRO E VERDADEIRO  
Resposta: VERDADEIRO
- h) **NÃO**  $Y \text{ OU } X \text{ E } C < A / B$   
**NÃO** VERDADEIRO OU FALSO E FALSO  
FALSO OU FALSO E FALSO  
FALSO OU FALSO  
Resposta: FALSO

i)  $Y \text{ OU } X \text{ E } C < A / B$

VERDADEIRO OU FALSO E FALSO

VERDADEIRO OU FALSO

Resposta: VERDADEIRO

j)  $(X \text{ OU } Y) \text{ E } C < A / B$

(FALSO OU VERDADEIRO) E FALSO

VERDADEIRO E FALSO

Resposta: FALSO

## Função

**Função** é uma rotina que retorna um valor específico.

- **STR( )**: Transforma número em caracteres numéricos;
- **VAL( )**: Transforma caracteres numéricos em número;
- **LEN( )**: Retorna um inteiro contendo o número de caracteres de uma sequência de caracteres, ou seja, conta os caracteres.

Exemplos:

**A = 10**

**B = STR(A) → B = "10"**

**C = STR(A \* 5) → C = "50"**

**D = STR(A - 6) → D = "4"**

**E = STR(2000) → E = "2000"**

**TOT = "123"**

**N1 = VAL ( TOT ) → N1 = 123**

**NOME = "JOAQUIM JOSE DA SILVA XAVIER"**

**N2 = LEN ( NOME ) → N2 = 28**

## Concatenação de alfanuméricos

Os símbolos para concatenação de alfanuméricos são: **&** ou **+**.

Exemplo:

```
DIA = 19  
MES = "abril"  
ANO = 2013  
CIDADE = "São Paulo"
```

A seguir, temos um exemplo de como fica a criação da variável **DATA** utilizando as variáveis anteriores para resultar no texto **São Paulo, 19 de abril de 2013**:

```
DATA = CIDADE & ", " & STR(DIA) & " de " & MES & " de " & STR(ANO)
```