

Marcos Antônio Pereira Simões

Specialist in Teaching for Professional Education – SENAC - 2012

Specialist in Information Technology Governance – SENAC – 2010

Technologist in Data Processing – Estácio de Sá - 2007

E-mail:

marcos.matriz@notacontrol.com.br



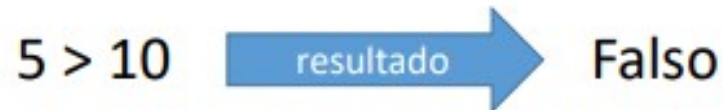
Analista e Desenvolvedor de Sistemas

Lógica de Programação

Dinâmica Gratidão

Lógica de Programação

Expressões comparativas



Lógica de Programação

Operadores comparativos

C, C++,
Java, C# →

Operador	Significado
>	maior
<	menor
>=	maior ou igual
<=	menor ou igual
==	igual
!=	diferente

Lógica de Programação

Exemplos de expressões comparativas

(suponha x igual a 5)

$X > 0$

Resultado: V

$X == 3$

Resultado: F

$10 \leq 30$

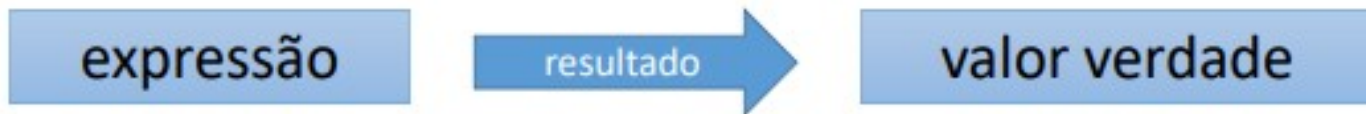
Resultado: V

$X \neq 2$

Resultado: V

Lógica de Programação

Expressões lógicas



Lógica de Programação

Operadores lógicos

C, C++,
Java, C# →

Operador	Significado
&&	E
	OU
!	NÃO

Lógica de Programação

Ideia por trás do operador "E"

Você pode obter uma habilitação de motorista se:

- For aprovado no exame psicotécnico,
E
- For aprovado no exame de legislação,
E
- For aprovado no exame de direção

**Todas condições
devem ser
verdadeiras!**

Lógica de Programação

Exemplos de expressões lógicas

(suponha x igual a 5)

$X \leq 20 \ \&\& \ X == 10$ Resultado: F

$\underbrace{}_{\text{V}} \quad \underbrace{}_{\text{F}}$

$X > 0 \ \&\& \ X != 3$ Resultado: V

$\underbrace{}_{\text{V}} \quad \underbrace{}_{\text{V}}$

$X \leq 20 \ \&\& \ X == 10 \ \&\& \ X != 3$ Resultado: F

$\underbrace{}_{\text{V}} \quad \underbrace{}_{\text{F}} \quad \underbrace{}_{\text{V}}$

Lógica de Programação

Tabela verdade do operador "E"

A	B	A && B
F	F	F
F	V	F
V	F	F
V	V	V

Lógica de Programação

Ideia por trás do operador "OU"

Você pode obter estacionar na vaga especial se:

- For idoso(a),
OU
- For uma pessoa com deficiência,
OU
- For uma gestante

**Pelo menos uma
condição deve
ser verdadeira!**

Lógica de Programação

Exemplos de expressões lógicas

(suponha x igual a 5)

$X == 10$ || $X \leq 20$ Resultado: V

F **V**

$X > 0$ || $X \neq 3$ Resultado: V

V **V**

$X \leq 0$ || $X \neq 3$ || $X \neq 5$ Resultado: V

F **V** **F**

Lógica de Programação

Tabela verdade do operador "OU"

A	B	A B
F	F	F
F	V	V
V	F	V
V	V	V

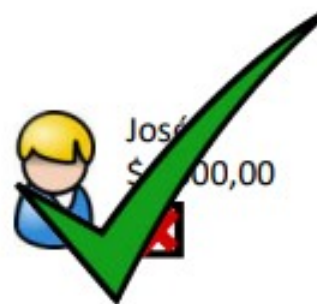
Lógica de Programação

Ideia por trás do operador "NÃO"

Você tem direito a receber uma bolsa de estudos se você:

NÃO

- Possuir renda maior que \$ 3000,00




O operador
"NÃO" inverte a
condição


Lógica de Programação

Exemplos de expressões lógicas

(suponha x igual a 5)

$!(X == 10)$

F

Resultado: V

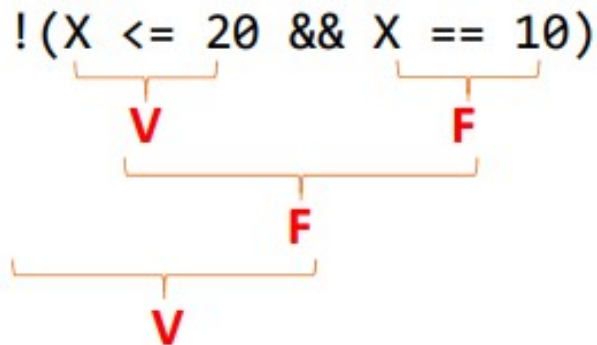
$!(X >= 2)$

V

Resultado: F

Lógica de Programação

Exemplos de expressões lógicas

(suponha x igual a 5)



Resultado: V

Lógica de Programação

Tabela verdade do operador "NÃO"

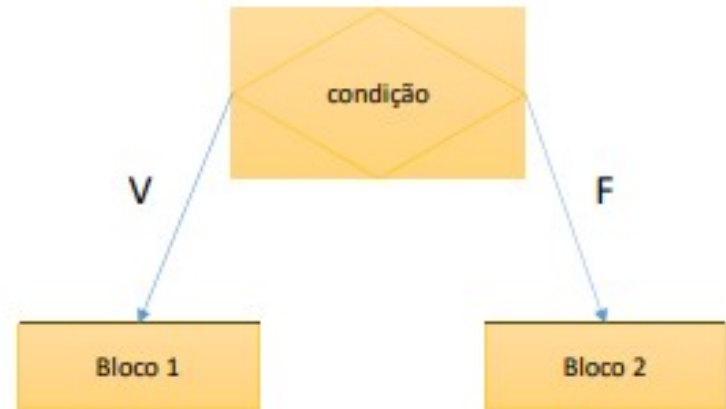
A	! A
F	V
V	F

Lógica de Programação

Conceito

Estrutura condicional:

É uma **estrutura de controle** que permite definir que um certo **bloco de comandos** somente será executado dependendo de uma **condição**



Lógica de Programação

Sintaxe da estrutura condicional

Simples:

```
if ( <condição> ) {  
    <comando 1>  
    <comando 2>  
}
```

REGRA:

V: executa o bloco de comandos
F: pula o bloco de comandos

*Importante:
Repare na indentação!*

Lógica de Programação

Sintaxe da estrutura condicional

Composta:

```
if ( <condição> ) {  
    <comando 1>  
    <comando 2>  
}  
else {  
    <comando 3>  
    <comando 4>  
}
```

*Importante:
Repare na endentação!*

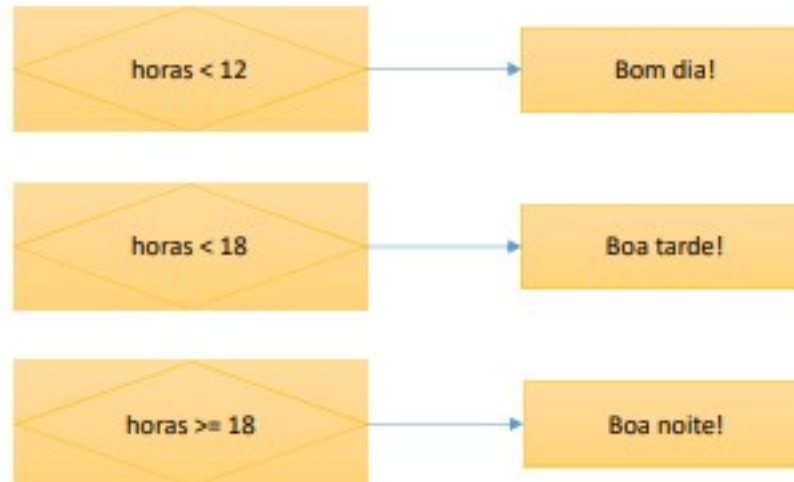
REGRA:

V: executa somente o bloco do **if**

F: executa somente o bloco do **else**

Lógica de Programação

E se eu tiver mais de duas possibilidades?



Lógica de Programação

Encadeamento de estruturas condicionais

```
if ( condição 1 ) {  
    comando 1  
    comando 2  
}  
else {  
    if ( condição 2 ) {  
        comando 3  
        comando 4  
    }  
    else {  
        comando 5  
        comando 6  
    }  
}
```

*Importante:
Repare na endentação!*

Lógica de Programação

Encadeamento de estruturas condicionais

```
if ( condição 1 ) {  
    comando 1  
    comando 2  
}  
else if ( condição 2 ) {  
    comando 3  
    comando 4  
}  
else if ( condição 3 ) {  
    comando 5  
    comando 6  
}  
else {  
    comando 7  
    comando 8  
}
```

*Importante:
Repare na endentação!*

Lógica de Programação

Operadores de Atribuição Cumulativa:

Operador	Descrição	Exemplo
=	Atribuição simples	$C = A + B$ atribui o valor de $A + B$ em C
+=	Atribuição aditiva	$C += A$ equivale a $C = C + A$
-=	Atribuição subtrativa	$C -= A$ equivale a $C = C - A$
*=	Atribuição multiplicativa	$C *= A$ equivale a $C = C * A$
/=	Atribuição de divisão	$C /= A$ equivale a $C = C / A$
%=	Atribuição de módulo	$C \% = A$ equivale a $C = C \% A$

Lógica de Programação

Lista de Exercícios 1 ao 18

Lógica de Programação

Switch-case:

Switch/case é uma estrutura de condição que define o código a ser executado com base em uma comparação de valores.

Quando se tem três ou mais opções de fluxo a serem tratadas com base no valor de uma variável, ao invés de várias estruturas if-else encadeadas, alguns preferem utilizar a estrutura switch-case.

```
1  switch (variável ou valor)
2  {
3  case valor1:
4      // código 1
5  break;
6  case valor2:
7      // código 2
8  break;
9  }
```

Lógica de Programação

Estruturas Repetitivas:

- Enquanto (while);
- Para (for);
- Faça-enquanto (do while)

Lógica de Programação

Estrutura repetitiva "enquanto"

É uma **estrutura de controle** que **repete** um bloco de comandos **enquanto** uma **condição** for verdadeira.

Quando usar: quando **não** se sabe previamente a quantidade de repetições que será realizada.

Por exemplo:

Fazer um programa que lê números inteiros até que um zero seja lido. Ao final mostra a soma dos números lidos.

Entrada	Saída
5 2 4 0	11

Lógica de Programação

Sintaxe / regra

```
while ( condição ) {  
    comando 1  
    comando 2  
}
```

Regra:

V: executa e volta

F: pula fora

Lógica de Programação

Resolução:

```
int x = int.Parse(Console.ReadLine());  
int soma = 0;
```

```
while (x != 0) {  
    soma += x;  
    x = int.Parse(Console.ReadLine());  
}
```

```
Console.WriteLine("A soma dos valores digitados é: " + soma);  
Console.ReadLine();
```