

# Operadores lógicos e relacionais

---

Prof.: Maiquel de Brito

6 de agosto de 2018

BLU3101 - Introdução à Informática para Automação

Departamento de Engenharias

UFSC Blumenau

# Table of contents

1. Operadores relacionais
2. Execução condicional
3. Expressões lógicas

# Operadores relacionais

---

Escreva um programa que leia dois números e, em seguida, imprima

- “*Números iguais*” se os números lidos forem iguais
- “*Primeiro maior*” se o primeiro número for maior que o segundo
- “*Segundo maior*” se o segundo número for maior que o primeiro

## Exercício

Elaborar um algoritmo para, entre duas pessoas, descobrir quem é o mais velho.

**Operadores matemáticos:** retornam números

Ex.:  $2 + 3 \rightsquigarrow$  retorna um 5

# Operadores relacionais

**Operadores matemáticos:** retornam números

Ex.:  $2 + 3 \rightsquigarrow$  retorna um 5

**Operadores relacionais:** comparam dois elementos e retornam *verdadeiro* (ou *True*) ou *falso* (ou *False*)

Ex.:  $2 < 3 \rightsquigarrow$  retorna um *verdadeiro*

## Operador `>` (maior):

`x > y`: retorna *True* se `x` for maior que `y`

### Exemplos

`5 > 2`

`2 > 5`

`5 > 5`



## Operador `>` (maior):

`x > y`: retorna *True* se `x` for maior que `y`

### Exemplos

`5 > 2`  $\rightsquigarrow$  *true*

`2 > 5`

`5 > 5`

## Operador `>` (maior):

`x > y`: retorna *True* se `x` for maior que `y`

### Exemplos

`5 > 2`  $\rightsquigarrow$  *true*

`2 > 5`  $\rightsquigarrow$  *false*

`5 > 5`

## Operador `>` (maior):

`x > y`: retorna *True* se `x` for maior que `y`

### Exemplos

`5 > 2`  $\rightsquigarrow$  *true*

`2 > 5`  $\rightsquigarrow$  *false*

`5 > 5`  $\rightsquigarrow$  *false*

## Operador $<$ (menor):

$x < y$ : retorna *True* se  $x$  for menor que  $y$

Exemplos

$5 < 2$

$2 < 5$

$5 < 5$

## Operador $<$ (menor):

$x < y$ : retorna *True* se  $x$  for menor que  $y$

### Exemplos

$5 < 2 \rightsquigarrow \text{false}$

$2 < 5$

$5 < 5$

## Operador $<$ (menor):

$x < y$ : retorna *True* se  $x$  for menor que  $y$

### Exemplos

$5 < 2 \rightsquigarrow false$

$2 < 5 \rightsquigarrow true$

$5 < 5$

## Operador $<$ (menor):

$x < y$ : retorna *True* se  $x$  for menor que  $y$

### Exemplos

$5 < 2 \rightsquigarrow \text{false}$

$2 < 5 \rightsquigarrow \text{true}$

$5 < 5 \rightsquigarrow \text{false}$

## Operador $\geq$ (maior ou igual):

$x \geq y$ : retorna *True* se  $x$  for maior ou igual a  $y$

### Exemplos

$5 \geq 2$

$2 \geq 5$

$5 \geq 5$



## Operador $\geq$ (maior ou igual):

$x \geq y$ : retorna *True* se  $x$  for maior ou igual a  $y$

### Exemplos

$5 \geq 2 \rightsquigarrow \text{true}$

$2 \geq 5$

$5 \geq 5$

## Operador $\geq$ (maior ou igual):

$x \geq y$ : retorna *True* se  $x$  for maior ou igual a  $y$

### Exemplos

$5 \geq 2 \rightsquigarrow \text{true}$

$2 \geq 5 \rightsquigarrow \text{false}$

$5 \geq 5$

# Operadores relacionais

## Operador $\geq$ (maior ou igual):

$x \geq y$ : retorna *True* se  $x$  for maior ou igual a  $y$

### Exemplos

$5 \geq 2 \rightsquigarrow \text{true}$

$2 \geq 5 \rightsquigarrow \text{false}$

$5 \geq 5 \rightsquigarrow \text{true}$

## Operador `<=` (menor ou igual):

`x <= y`: retorna *True* se `x` for menor ou igual a `y`

### Exemplos

`5 <= 2`

`2 <= 5`

`5 <= 5`

## Operador $\leq$ (menor ou igual):

$x \leq y$ : retorna *True* se  $x$  for menor ou igual a  $y$

### Exemplos

$5 \leq 2 \rightsquigarrow \text{false}$

$2 \leq 5$

$5 \leq 5$

## Operador $\leq$ (menor ou igual):

$x \leq y$ : retorna *True* se  $x$  for menor ou igual a  $y$

### Exemplos

$5 \leq 2 \rightsquigarrow \text{false}$

$2 \leq 5 \rightsquigarrow \text{true}$

$5 \leq 5$

# Operadores relacionais

## Operador $\leq$ (menor ou igual):

$x \leq y$ : retorna *True* se  $x$  for menor ou igual a  $y$

### Exemplos

$5 \leq 2 \rightsquigarrow \text{false}$

$2 \leq 5 \rightsquigarrow \text{true}$

$5 \leq 5 \rightsquigarrow \text{true}$

# Operadores relacionais

**Operador == (igual):** `x == y`: retorna *True* se `x` for igual a `y`

Exemplos

`5 == 2`

`2 == 5`

`5 == 5`



# Operadores relacionais

**Operador == (igual):**  $x == y$ : retorna *True* se  $x$  for igual a  $y$

Exemplos

$5 == 2 \rightsquigarrow \text{false}$

$2 == 5$

$5 == 5$

# Operadores relacionais

**Operador == (igual):**  $x == y$ : retorna *True* se  $x$  for igual a  $y$

Exemplos

$5 == 2 \rightsquigarrow \text{false}$

$2 == 5 \rightsquigarrow \text{false}$

$5 == 5$

# Operadores relacionais

**Operador == (igual):**  $x == y$ : retorna *True* se  $x$  for igual a  $y$

Exemplos

$5 == 2 \rightsquigarrow \text{false}$

$2 == 5 \rightsquigarrow \text{false}$

$5 == 5 \rightsquigarrow \text{true}$

## Operador `!=` (diferente):

`x != y`: retorna *True* se `x` for diferente de `y`

### Exemplos

`5 != 2`

`2 != 5`

`5 != 5`

## Operador `!=` (diferente):

`x != y`: retorna *True* se `x` for diferente de `y`

### Exemplos

`5 != 2`  $\rightsquigarrow$  *true*

`2 != 5`

`5 != 5`

## Operador $\neq$ (diferente):

$x \neq y$ : retorna *True* se  $x$  for diferente de  $y$

### Exemplos

$5 \neq 2 \rightsquigarrow \text{true}$

$2 \neq 5 \rightsquigarrow \text{true}$

$5 \neq 5$

## Operador `!=` (diferente):

`x != y`: retorna *True* se `x` for diferente de `y`

### Exemplos

`5 != 2`  $\rightsquigarrow$  *true*

`2 != 5`  $\rightsquigarrow$  *true*

`5 != 5`  $\rightsquigarrow$  *false*

# Execução condicional

---



Em algumas situações, é necessário que um programa execute alguns comandos somente se uma condição for verdadeira

Exemplos:

- se a média for maior ou igual a 7.0, então imprima  
*“Você foi aprovado! Parabéns!”*
- se a idade for igual ou maior que 16 anos, então imprima  
*“Você pode votar!”*

# Execução condicional em C

```
if <expressão relacional>{  
    comando 1  
    comando 2  
    ⋮  
    comando n  
}
```

## Exemplo

```
1  int i = 0;  
2  printf("Informe sua idade: ");  
3  scanf("%d",&i);  
4  if(idade < 16){  
5      printf("Você não pode votar,")  
6  }
```

Escrever um programa em que o usuário informe sua nota. Se a nota for maior ou igual a 7.0, então imprima  
*“Você foi aprovado! Parabéns!”*

1. Escrever um programa em que o usuário informe nota. Se a nota for maior ou igual a 7.0, então imprima "*Você foi aprovado! Parabéns!*". Se a nota for menor que 7.0, imprima "*Você foi reprovado :(*"
2. Escrever um programa em que o usuário informe sua idade. Se a idade for igual ou maior que 16 anos, então imprima "*Você pode votar!*". Se a idade for menor que 16 anos, imprima "*Você não pode votar!*"

# Execução condicional em C

**else – “senão”:** determina comandos a serem executados caso a condição do “if” não seja verdadeira

## Exemplo

```
1  int i = 0;
2  printf("Informe sua idade: ");
3  scanf("%d",&i);
4  if(i<16)
5      printf("Você não pode votar\n");
6  else
7      printf("Você já pode votar\n");
8
```

# Execução condicional em C

**else – “senão”**: determina comandos a serem executados caso a condição do “if” não seja verdadeira

## Exemplo

```
1  int i = 0;
2  printf("Informe sua idade: ");
3  scanf("%d",&i);
4  if(i<16)
5      printf("Você não pode votar\n");
6  else if (i<18)
7      printf("Você já pode votar\n");
8  else
9      printf("Você é obrigado a votar\n");
10
```

# Execução condicional com switch

**switch:** testa uma expressão lógica contra diversos valores `int` ou `char`.  
executa comandos associados aos valores que coincidem  
com a expressão lógica

## Sintaxe:

```
switch(expressão)  
case valor 1  
    sequência de comandos  
    break;  
case valor 2  
    sequência de comandos  
    break;  
case valor 2  
default  
    sequência de comandos  
    break;
```

**break:** interrompe a execução do  
laço

**default:** acionado caso nenhum  
valor corresponda à expressão  
lógica. É opcional.

# Expressões lógicas

---



# Expressões lógicas

**Expressões relacionais** retornam *valores lógicos*

(i.e verdadeiro ou falso)

## Exemplos

$2 \geq 1 \rightsquigarrow \text{verdadeiro}$

$2 \geq 2 \rightsquigarrow \text{verdadeiro}$

$2 \geq 5 \rightsquigarrow \text{falso}$

**Expressões lógicas** comparam o resultado de expressões relacionais

e também retornam (*verdadeiro ou falso*)

**Expressões lógicas** comparam o resultado de expressões relacionais e também retornam (*verdadeiro ou falso*)

## Exemplos

$2 \geq 1$  e  $2 \geq 2 \rightsquigarrow ???$

$2 \geq 2$  ou  $2 \geq 5 \rightsquigarrow ???$

**Operador and (“e”):** *expressão\_1* && *expressão\_2*

retorna *True* se a *expressão\_1* e *expressão\_2* forem verdadeiras  
as duas expressões devem ser verdadeiras ao mesmo tempo

Exemplos

2 >= 1    &&    2 >= 2

2 >= 2    &&    2 >= 5

2 >= 1    &&    2 >= 2    &&    2 >= 5

**Operador and (“e”):** *expressão\_1* && *expressão\_2*

retorna *True* se a *expressão\_1* e *expressão\_2* forem verdadeiras  
as duas expressões devem ser verdadeiras ao mesmo tempo

Exemplos

2 >= 1    &&    2 >= 2     $\rightsquigarrow$  verdadeiro

2 >= 2    &&    2 >= 5

2 >= 1    &&    2 >= 2    &&    2 >= 5

**Operador and (“e”):** *expressão\_1* && *expressão\_2*

retorna *True* se a *expressão\_1* e *expressão\_2* forem verdadeiras  
as duas expressões devem ser verdadeiras ao mesmo tempo

Exemplos

2 >= 1    &&    2 >= 2     $\rightsquigarrow$  verdadeiro

2 >= 2    &&    2 >= 5     $\rightsquigarrow$  falso

2 >= 1    &&    2 >= 2    &&    2 >= 5

**Operador and (“e”):** *expressão\_1* && *expressão\_2*

retorna *True* se a *expressão\_1* e *expressão\_2* forem verdadeiras  
as duas expressões devem ser verdadeiras ao mesmo tempo

Exemplos

2 >= 1    &&    2 >= 2     $\rightsquigarrow$  verdadeiro

2 >= 2    &&    2 >= 5     $\rightsquigarrow$  falso

2 >= 1    &&    2 >= 2    &&    2 >= 5     $\rightsquigarrow$  falso

Escrever um programa que solicite a idade de três pessoas.

Ao final, o programa deve informar se a primeira pessoa é a mais velha, ou se a segunda pessoa é a mais velha, ou se a terceira pessoa é a mais velha ou se as três pessoas têm a mesma idade.

**Operador or (“ou”):** *expressão\_1* || *expressão\_2*

retorna *True* se a *expressão\_1* **ou** *expressão\_2* forem verdadeiras  
pelo menos uma das duas expressões deve ser verdadeira

Exemplos

2 >= 1 || 2 >= 2

2 >= 2 || 2 >= 5

2 >= 3 || 2 >= 5



**Operador or (“ou”):** *expressão\_1* || *expressão\_2*

retorna *True* se a *expressão\_1* **ou** *expressão\_2* forem verdadeiras  
pelo menos uma das duas expressões deve ser verdadeira

Exemplos

2 >= 1 || 2 >= 2  $\rightsquigarrow$  verdadeiro

2 >= 2 || 2 >= 5

2 >= 3 || 2 >= 5

**Operador or (“ou”):** *expressão\_1* || *expressão\_2*

retorna *True* se a *expressão\_1* **ou** *expressão\_2* forem verdadeiras  
pelo menos uma das duas expressões deve ser verdadeira

Exemplos

2 >= 1 || 2 >= 2  $\rightsquigarrow$  verdadeiro

2 >= 2 || 2 >= 5  $\rightsquigarrow$  verdadeiro

2 >= 3 || 2 >= 5

**Operador or (“ou”):** *expressão\_1* || *expressão\_2*

retorna *True* se a *expressão\_1* **ou** *expressão\_2* forem verdadeiras  
pelo menos uma das duas expressões deve ser verdadeira

Exemplos

2 >= 1		2 >= 2	↗	verdadeiro
2 >= 2		2 >= 5	↗	verdadeiro
2 >= 3		2 >= 5	↗	falso

Em um parque de diversões, um determinado brinquedo só permite a entrada de pessoas com mais de 7 anos ou com altura igual ou superior a 1,60m.

Escreva um programa que solicite a idade e a altura de uma pessoa e, ao final, informe se essa pessoa pode utilizar o brinquedo.

## Tabela Verdade

Sejam *expr 1* e *expr 2* duas expressões relacionais quaisquer.

<i>expr 1</i>	<i>expr 2</i>	<i>expr 1</i> && <i>expr 2</i>	<i>expr 1</i>    <i>expr 2</i>
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	F

## Precedência de operadores

$x=8$

$y=6$

$z=1$

## Precedência de operadores

x=8

y=6

z=1

x<5 && y<5 || z<5

## Precedência de operadores

x=8

y=6

z=1

x<5 && y<5 || z<5     $\rightsquigarrow$  verdadeiro ou falso?



## Precedência de operadores

x=8

y=6

z=1

x<5 && y<5 || z<5     $\rightsquigarrow$  verdadeiro

## Precedência de operadores

x=8

y=6

z=1

x<5 && y<5 || z<5      $\rightsquigarrow$  verdadeiro

x<5 and (y<5 or z<5)      $\rightsquigarrow$  verdadeiro ou falso?

## Precedência de operadores

x=8

y=6

z=1

x<5 && y<5 || z<5      $\rightsquigarrow$  verdadeiro

x<5 and (y<5 or z<5)      $\rightsquigarrow$  falso

## Precedência de operadores

1. Parênteses
2. and
3. or

**Operador not (“não”):**  $! \text{ expressão\_1}$

retorna a *negação* de uma expressão relacional

retorna *verdadeiro* se uma expressão for *falsa*

retorna *falso* se uma expressão for *verdadeira*

Exemplos

$!(2 == 5) \rightsquigarrow \text{verdadeiro}$

$!(2 == 2) \rightsquigarrow \text{falso}$

$!(2 != 5) \rightsquigarrow \text{falso}$

$!(2 != 2) \rightsquigarrow \text{verdadeiro}$

**Operador not (“não”):** *continuação*

`2>1 && 2>5`  $\rightsquigarrow$  falso

`!(2>1 && 2>5)`  $\rightsquigarrow$  verdadeiro

`!(2>1 || 2>5)`  $\rightsquigarrow$  falso

## Exercício

Crie um programa que leia três números e depois informe o maior número, o número do meio e o menor número lidos.

Assuma que os números digitados serão todos diferentes entre si.

Em um teatro, a primeira fileira é reservada para

- menores de 12 anos
- mulheres com mais de 55 anos
- homens com mais de 60 anos

Escreva um programa que solicite a idade e o sexo do usuário (“M” para masculino e “F” para feminino) e que, a partir disso, informe se o usuário pode sentar na primeira fila.



# Exercício

Uma lanchonete tem uma promoção em que dá desconto para os seguintes *combos*:

- hamburger + fritas
- hamburger + refrigerante
- fritas + sorvete

– Não é concedido desconto se o combo incluir qualquer item a mais (ex.: hamburger + fritas + sorvete)

– Escreva um programa que solicite se o usuário deseja hamburger, fritas, refrigerante e sorvete e, a partir disso, informe se usuário tem direito a desconto.

Para perguntar se o usuário deseja algum dos itens, utilizar `input("Hamburger <S>? ")`. O usuário deve informar "S" se quiser um item, ou deixar em branco caso não queira.

# Exercício

Em uma promoção, são distribuídos prêmios conforme o sorteio de números entre zero e 99. Os prêmios são dados da seguinte forma:

- R\$ 100,00 para números pares não múltiplos de 5
- R\$ 200,00 para números pares múltiplos de 7
- R\$ 100,00 para números ímpares
- R\$ 200,00 para números ímpares múltiplos de 3 e não múltiplos de 7

Crie um programa de computador em que o usuário informe um número e que, a partir disso, informe qual será o valor do prêmio