



UC Programação de Sistemas Computacionais

Professor Jean Carlo Wagner

03.03.2022

➤ Agenda

- Operadores aritméticos (revisão)
- Operadores relacionais (revisão)
- Operadores lógicos
- Operador ternário
- Estruturas de seleção if
- Estruturas de seleção switch/case

Operadores

The background of the slide is a solid dark purple. Overlaid on this are several thin, wavy lines in a reddish-pink color. These lines flow from the left side towards the right, creating a sense of movement and depth. The lines are closely spaced in some areas and more spread out in others, forming a complex, organic pattern that frames the central text.

➤ Operadores aritméticos

Operação	Operador	Expressão	Resultado
Adição	+	$6 + 4$	10
Subtração	-	$7 - 9$	-2
Multiplicação	*	$12 * 3$	36
Divisão	/	$44 / 2$	22
Módulo (resto da divisão)	%	$10 \% 3$	1

- Precedência de operadores:



ã Precedência de operadores: exemplos

$6 * 4 + 5 - 3$

$6 * 4 + 5 - 3$

$24 + 5 - 3$

$29 - 3$

26

$3 + (2 - 2) * 5$

$3 + (2 - 2) * 5$

$3 + 4 * 5$

$3 + 20$

23

$37 \% 5 * 3$

$37 \% 5 * 3$

$2 * 3$

6

$37 \% (5 * 3)$

$37 \% (5 * 3)$

$37 \% (15)$

7

➤ Operadores relacionais

- Possibilitam comparar valores ou expressões, retornando um resultado lógico verdadeiro ou falso.

Função	Operador	Exemplo
Igual	==	$x == y$
Diferente	!=	$x != y$
Maior que	>	$y > y$
Maior ou igual a	>=	$x >= y$
Menor que	<	$x < y$
Menor ou igual a	<=	$x <= y$

➤ Operadores lógicos

- Permitem avaliar o resultado lógico de diferentes operações aritméticas em uma expressão.

Função	Operador	Exemplo
AND	&&	x && y
OR		x y
NOT	!	!x

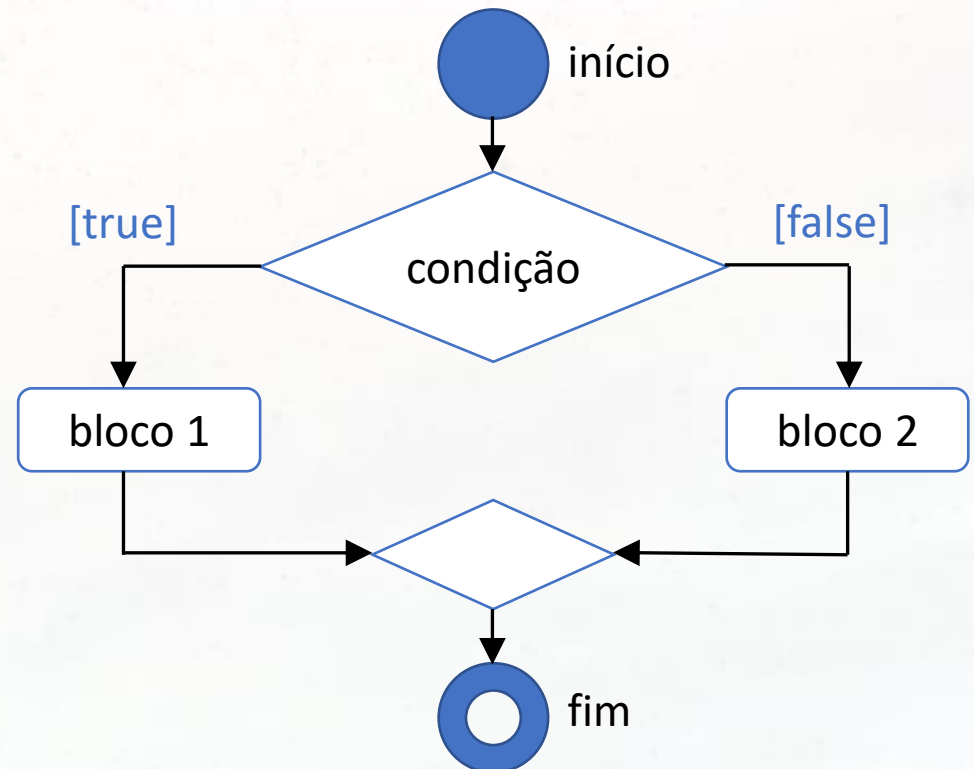
- A linguagem de programação Java também possui algumas sequências de escape, que são atalhos utilizados para representar um caractere especial como, por exemplo, uma quebra de linha (`\n`).

Sequência	Caractere especial	Sequência	Caractere especial
<code>\b</code>	Espaço	<code>\t</code>	Tabulação
<code>\f</code>	Form feed	<code>\"</code>	Aspas duplas
<code>\n</code>	Nova linha	<code>'</code>	Aspas simples
<code>\r</code>	Retorno	<code>\\</code>	Barra invertida

➤ Estruturas condicionais: if-else

- As estruturas condicionais existem em todas as linguagens de programação e possibilitam que a execução de um programa seja desviada de acordo com certas condições. Os comandos condicionais usados em Java são **if-else** e **switch-case**. Essas duas estruturas de desvio possibilitam executar diferentes trechos de um programa com base em certas condições.

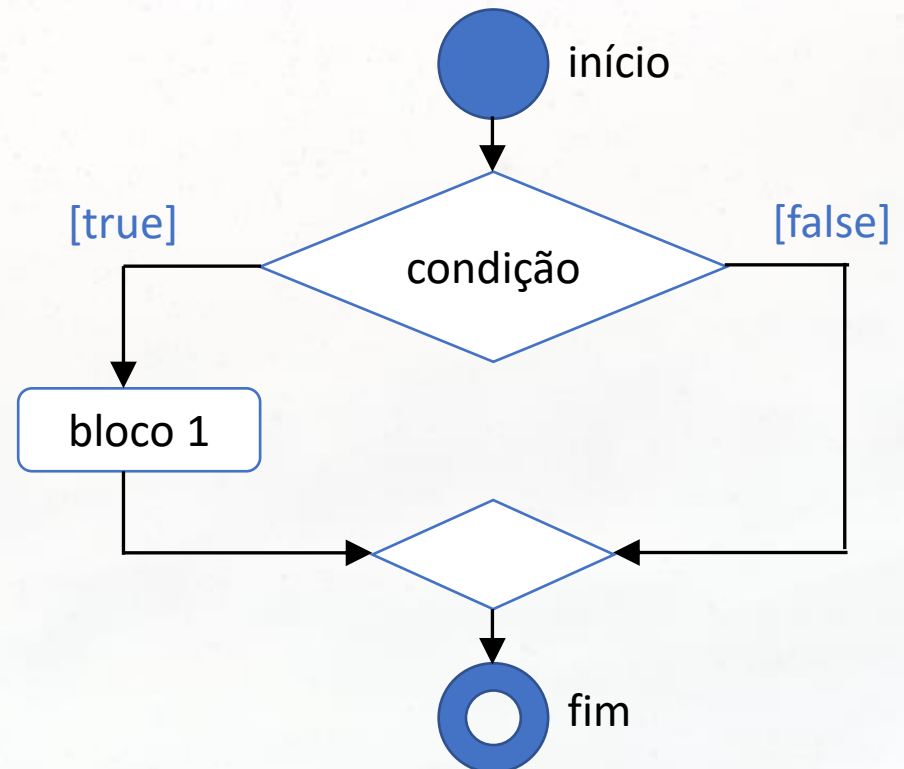
```
if (<condição>) {  
    <instruções para condição verdadeira>  
} else {  
    <instruções para condição falsa>  
}
```



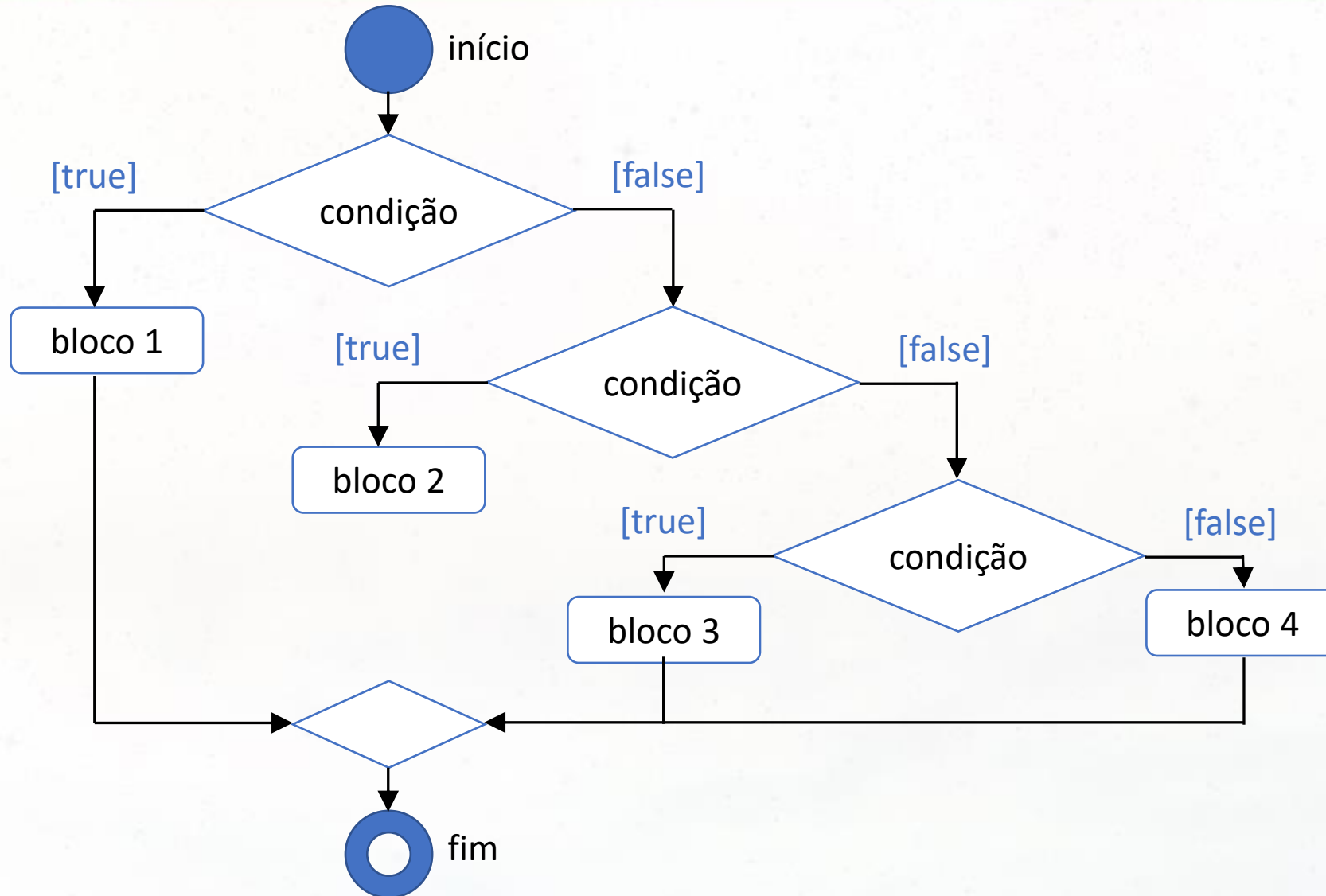
➤ Estruturas condicionais: if-else

- A estrutura **if-else** apresentada não é a única válida, pois existem outras maneiras diferentes de se criar essa estrutura: *if* sem o *else*, *if* com o *else* e *if* com o *else* aninhado.

```
if (<condição>) {  
    <instruções para condição verdadeira>  
}
```



Estruturas condicionais: if-else



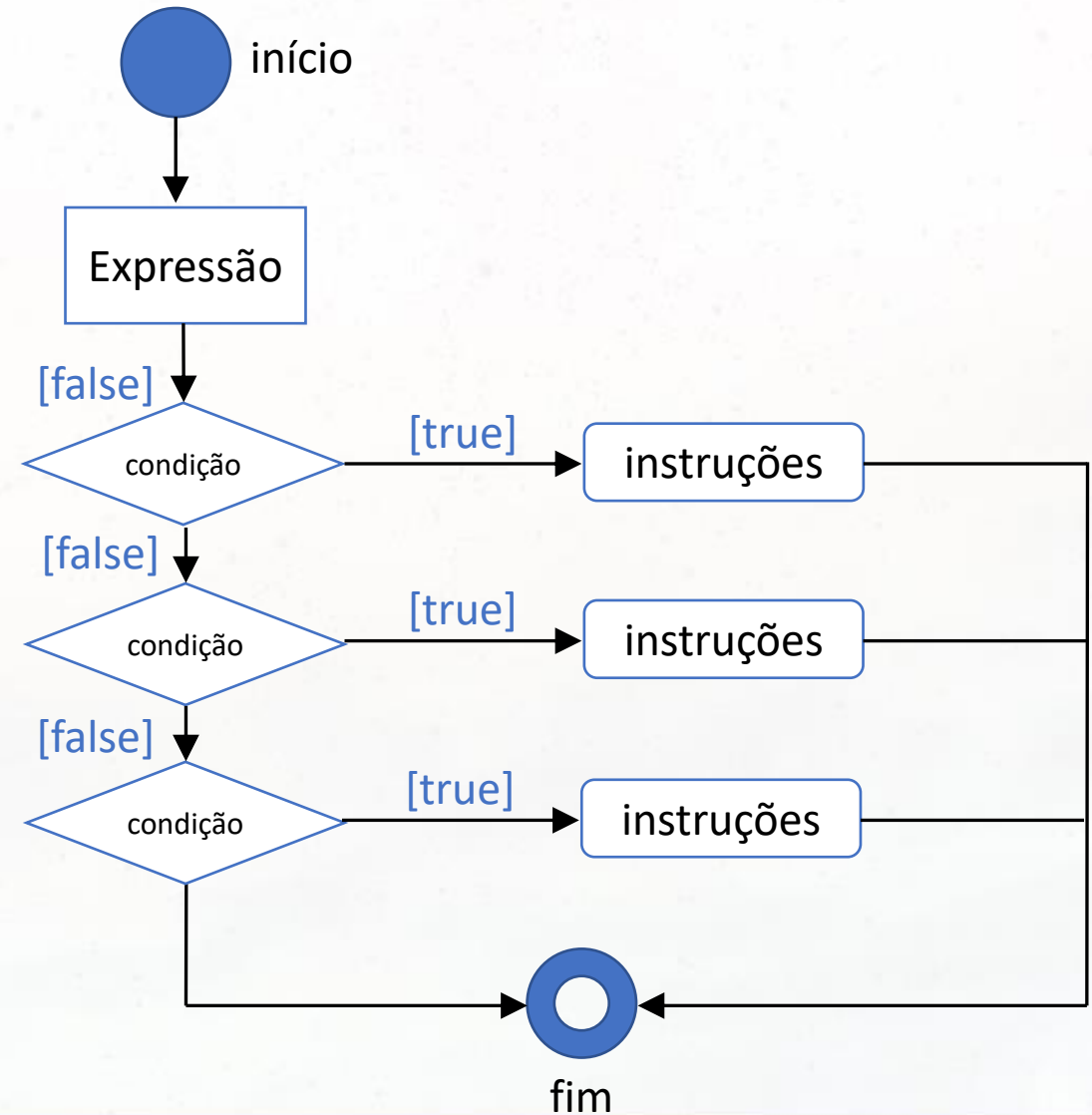
ã Estrutura switch-case

- Ao trabalhar com uma grande quantidade de desvios condicionais contendo instruções *if*, pode-se comprometer a inteligibilidade do programa, dificultando sua interpretação. A estrutura **switch-case** possibilita uma forma mais adequada e eficiente de atender a esse tipo de situação, constituindo-se uma estrutura de controle com múltipla escolha.
- A estrutura **switch-case** equivale a um conjunto de instruções *if* encadeadas, fornecendo maior inteligibilidade.

```
switch (<expressão>) {  
    case 1: instruções; break;  
    case 2: instruções; break;  
    case 3: instruções; break;  
    default: instruções;  
}
```

ǎ Estrutura switch-case

```
switch (<expressão>) {  
  case 1: instruções; break;  
  case 2: instruções; break;  
  case 3: instruções; break;  
  default: instruções;  
}
```



ǎ Dúvidas!?!





ecossistema
ănima