

Operadores Relacionais

São símbolos especiais capazes de realizar comparações entre determinados operandos e retornar um resultado

Tipos:

- Similaridade: igual (==), diferente (!=)
- Tamanho: maior(>), maior ou igual (>=), menor(<), menor ou igual (<=)

Obs: Não conseguimos comparar todas as variáveis e classes, por exemplo, não tem como comparar uma classe string (texto) com uma variável tipo int (número inteiro). Assim como não conseguimos dizer se uma variável booleana é maior ou menor que uma variável int.

Operadores Lógicos

São símbolos especiais capazes de realizar comparações lógicas entre operandos lógicos ou expressões e, em seguida, retornar um resultado

Tipos:

- Conjunção: ambos os operandos ou expressões são verdadeiras (&& - and)

Simbologia:

- &&

Terminologia:

- and(e)

O-E	O-E	R
V	V	V
V	F	F
F	V	F
F	F	F

- Disjunção: ambos os operandos ou expressões são falsos (|| - or)

Simbologia:

- ||

Terminologia:

- or(ou)

O-E	O-E	R
V	V	V
V	F	V
F	V	V
F	F	F

- Disjunção exclusiva: operações que só são verdade quando ambos os operandos são opostos(^)

Simbologia:

- \wedge

Terminologia:

- xor

O-E	O-E	R
V	V	F
V	F	V
F	V	V
F	F	F

- Negação: operação que inverte o valor lógico de um operando ou expressão(!)

Simbologia:

- !

Terminologia:

- inversão

O-E	R
V	F
F	V



LÓGICA PROPOSICIONAL



Anhanguera

Tabelas Verdades

Conjunção (E – AND):

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

Negação (NÃO – NOT):

p	$\neg p$
V	F
F	V

Disjunção (OU – OR):

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F

Condicional:

p	q	$p \rightarrow q$
V	V	V
V	F	F
F	V	V
F	F	V

Bi condicional:

p	q	$p \leftrightarrow q$
V	V	V
V	F	F
F	V	F
F	F	V

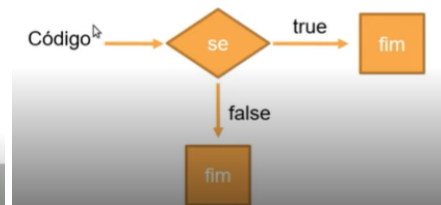
Controle de Fluxo

São estruturas que tem a capacidade de direcionar o fluxo de execução do código.

Tipos:

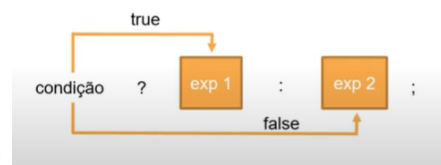
- Decisão: if, if-else, if-else-if, switch, e operador ternário
- Repetição: for, while, do while
- Interrupção: break, continue, return
- Decisão: if, if-else, if-else-if

```
if (condição) {  
}  
if (condição) {  
} else {  
}  
if (condição) {  
} else if (condição) {  
} else {  
}
```



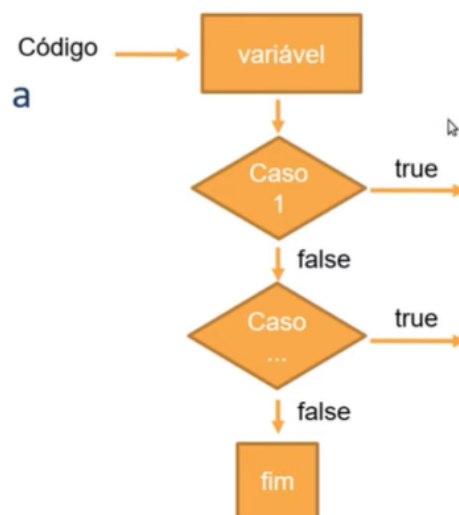
- Decisão: operador ternário

```
condição ? true : false;    condição ? true : null;  
ligado ? desligar : ligar;  emMovimento ? freia : null;
```



- Decisão: switch

```
Variável:      switch (variável) {  
byte           case 1 :  
short          break;  
char           case ... :  
int            break;  
Enum           default:  
String         break;  
                }
```



switch - usar para valores exatos

if - usar para expressões booleanas

evitar o uso de muitos if's aninhados e em flecha

usar o default do switch para apontar um valor errado, não usar para casos genéricos

Blocos

São grupos de 0 ou mais códigos que trabalham em conjunto para executar uma operação representado com chaves {}

Tipos:

- Locais: dentro de métodos
- Estáticos: dentro de classes
- Instância: dentro de classes