

POR CRISTIANE PAGINE

# JAVA

## CONDICIONAIS E ESTRUTURAS DE REPETIÇÃO JAVA

2025

# 1. Condicionais

As estruturas condicionais em Java permitem a execução de diferentes blocos de código com base em condições lógicas. Elas são fundamentais para a tomada de decisões nos programas.

## if – Condição Simples

Executa um bloco de código se a condição for verdadeira.

```
int numero = 10;
if (numero > 0) {
    System.out.println("O número é positivo.");
}
```

## if-else – Condição com Alternativa

Executa um bloco se a condição for verdadeira e outro se for falsa.

```
int numero = -5;
if (numero > 0) {
    System.out.println("O número é positivo.");
} else {
    System.out.println("O número é negativo ou zero.");
}
```

## **else if – Múltiplas Condições**

Usado para verificar várias condições sequenciais.

```
int nota = 75;
if (nota >= 90) {
    System.out.println("Aprovado com excelência.");
} else if (nota >= 60) {
    System.out.println("Aprovado.");
} else {
    System.out.println("Reprovado.");
}
```

## **switch – Alternativa para Múltiplas Opções**

Mais eficiente quando há muitas comparações de um mesmo valor.

```
int dia = 3;
switch (dia) {
    case 1:
        System.out.println("Segunda-feira");
        break;
    case 2:
        System.out.println("Terça-feira");
        break;
    case 3:
        System.out.println("Quarta-feira");
        break;
    default:
        System.out.println("Dia inválido");
}
```

## 2.Laços de Repetição

Os laços de repetição em Java são estruturas que permitem a execução repetitiva de um bloco de código enquanto uma condição for verdadeira. Eles são essenciais para a automação de tarefas e processamento de dados.

### **for – Laço com Contador**

Utilizado quando o número de repetições é conhecido.

```
for (int i = 0; i < 5; i++) {  
    System.out.println(i);  
    // Exibe os números de 0 a 4  
}
```

### **while – Laço com Condição**

Executa o bloco de código enquanto a condição for verdadeira.

```
int x = 0;  
while (x < 5) {  
    System.out.println(x);  
    x++; // Incrementa x até 5  
}
```

## **do-while – Execução Garantida pelo Menos uma Vez**

O bloco é executado pelo menos uma vez antes de verificar a condição.

```
int y = 0;
do {
    System.out.println(y);
    y++;
} while (y < 5);
```

Essas estruturas são muito usadas para percorrer listas, processar entradas do usuário e criar loops eficientes em aplicações Java.

## **while – Laço com Condição**

Executa o bloco de código enquanto a condição for verdadeira.

```
int x = 0;
while (x < 5) {
    System.out.println(x);
    x++; // Incrementa x até 5
}
```

## 3. Operador Ternário

O operador ternário (`? :`) é uma forma simplificada de escrever expressões condicionais em Java. Ele substitui um `if-else` simples, tornando o código mais conciso e legível.

Sintaxe:

`condição ? expressão_se_verdadeiro : expressão_se_falso;`

Exemplo:

```
int numero = 10;
String resultado = (numero % 2 == 0) ? "Par" :
"Ímpar";
System.out.println("O número é " + resultado);
//Saída: O número é Par
```

Exemplo comparando ternário com `if-else`:

```
//usando if-else
int numero = 10;
String resultado;
if (numero % 2 == 0) {
    resultado = "Par";
} else {
    resultado = "Ímpar";
}
System.out.println("O número é " + resultado);

//usando ternário
String resultado = (numero % 2 == 0) ? "Par" :
"Ímpar";
```