



**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
BAHIA

JAVA - ESTRUTURAS DE CONTROLE

IF-ELSE



- A decisão if-else é útil para executar instruções que dependem de uma certa condição;

- Sintaxe do if-else:

```
if (condicaoBooleana) {
```

```
    // código Java da condição obtida
```

```
} else {
```

```
    // código Java da condição não obtida
```

```
}
```

- O termo condicaoBooleana é uma expressão que só poderá retornar true ou false;

- Para concatenar expressões booleanas, basta usar os operadores lógicos a seguir:

- “E” lógico equivale a &&;

- “OU” lógico equivale a ||

- Para negar uma expressão booleana, basta adicionar o sinal de exclamação (!) à frente da expressão.

- NÃO lógico equivale a !

IF-ELSE

```
public static void main(String[] args) {  
    int idade = 20;  
    if (idade < 21) {  
        System.out.println("Não pode entrar");  
    } else {  
        System.out.println("Pode entrar");  
    }  
  
    boolean amigoDoDono = false;  
    if (idade < 21 && amigoDoDono == false) {  
        System.out.println("Não pode entrar");  
    } else {  
        System.out.println("Pode entrar");  
    }  
  
    if (idade < 21 && !amigoDoDono) {  
        System.out.println("Não pode entrar");  
    } else {  
        System.out.println("Pode entrar");  
    }  
}
```

ATIVIDADE



- Na aplicação “Balança Trimestral”, identifique o menor e o maior valor e imprima a mensagem conforme abaixo:
- O menor valor foi R\$ _____,____ referente ao mês de _____
- O maior valor foi R\$ _____,____ referente ao mês de _____

- Dica: para realizar as comparações, crie uma variável auxiliar.

```
float gastosJaneiro = 15000;
float gastosFevereiro = 23000;
float gastosMarco = 17000;
//variável auxiliar para comparações
float maiorValor = gastosJaneiro;
//identifica o maior valor
if (maiorValor < gastosFevereiro) {
    maiorValor = gastosFevereiro;
}
if (maiorValor < gastosMarco) {
    maiorValor = gastosMarco;
}
//equipara o valor com meses e imprime a mensagem
if (maiorValor == gastosJaneiro) {
    System.out.println("O maior valor foi R$ " + maiorValor + " referente ao mês de Janeiro");
} else {
    if (maiorValor == gastosFevereiro) {
        System.out.println("O maior valor foi R$ " + maiorValor + " referente ao mês de Fevereiro");
    } else {
        System.out.println("O maior valor foi R$ " + maiorValor + " referente ao mês de Março");
    }
}
```

DECISÃO SWITCH-CASE



- A estrutura switch-case identifica o valor de uma variável (switch), e a testa em uma lista de valores possíveis (case).
- Sintaxe do switch-case:
switch (variável) {
 case valor1 :// comandos Java para valor1
 case valor2 :// comandos Java para valor2
 case valor3 :// comandos Java para valor3
 default :// se nenhum dos valores anteriores ocorreu
}

- Os tipos da variável do switch só podem ser:
 - Byte;
 - Short;
 - Char;
 - Int;
 - Strings – cadeia de caracteres;
 - Enumeradores.
- Ao final de bloco de comandos do case, deverá ser colocado o break: para evitar que o case seguinte seja executado.

DECISÃO SWITCH-CASE

```
public static void main(String[] args) {  
    int nota = 8;  
    switch (nota) {  
        case 10:  
            System.out.println("Excelente");  
            break;  
        case 9:  
            System.out.println("Muito bom");  
            break;  
        case 8:  
            System.out.println("Bom");  
            break;  
        case 7:  
            System.out.println("Ná média");  
            break;  
        default:  
            System.out.println("Nota baixa");  
    }  
}
```

REPETIÇÃO COM WHILE



- O while é utilizado para repetir um trecho de código até que uma condição seja atingida:
 - Laço (loop).
 - Sintaxe do while:
- O termo condicaoBooleana é uma expressão que só poderá retornar true ou false:
 - Podem ser usados o && e || para concatenação de expressões.

```
while (condicaoBooleana) {  
    // comandos a serem executados  
}
```

```
int idade = 15;  
while (idade < 18) {  
    System.out.println(idade);  
    idade = idade + 1;  
}
```

```
int i = 0;  
while (i != 10) {  
    System.out.println(i);  
    i = i + 1;  
}
```


REPETIÇÃO COM FOR



- Outro comando de loop é o for;
- Repetir determinado trecho de código até uma condição ser atingida.
- Sintaxe do for.

```
for(inicialização; condição; incremento) {  
    // comandos a serem executados  
}
```

```
for(int i = 0; i < 10; i++) {  
    // comandos a serem executados  
}
```

ATIVIDADE



- Escreva um programa em java que:
 - Imprima todos os números pares existentes entre 0 e 100 (0, 2, 4, 6, 8, 10...)
 - Imprima todos os múltiplos de 3 existentes entre 0 e 100 (0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, ..)
 - Números primos entre 0 e 100

ATIVIDADE



- Imprima todos os números pares existentes entre 0 e 100 (0, 2, 4, 6, 8, 10...)

```
System.out.print("Pares entre 0 e 100: ");  
for (int i = 0; i <= 100; i++) {  
    if (i % 2 == 0) {  
        System.out.print(i + ", ");  
    }  
}
```

ATIVIDADE



- Imprima todos os múltiplos de 3 existentes entre 0 e 100 (0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, ..)

```
System.out.println("Múltiplos de 3 entre 0 e 100: ");  
for (int i = 0; i <= 100; i++) {  
    if (i % 3 == 0) {  
        System.out.print(i + ", ");  
    }  
}
```

ATIVIDADE

- Números primos entre 0 e 100

```
System.out.println("Números primos: ");
//números a serem verificados
for (int i = 0; i <= 100; i++) {
    int divisores = 1;
    //dividindo por todos os demais
    for (int j = 2; j <= i; j++) {
        if (i % j == 0) {
            divisores++;
        }
    }
    //1 sempre é primo
    if (i == 1) {
        System.out.print(i + ", ");
    }
    //dividido por 1 e por ele meso
    if (divisores == 2) {
        System.out.print(i + ", ");
    }
}
```