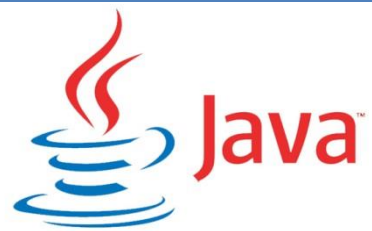


# Lógica de Programação

**Unidade 14** – Laços de Repetição (do while)



**QI ESCOLAS E FACULDADES**  
Curso Técnico em Informática

## SUMÁRIO


<b>INSTRUÇÃO DO WHILE (FAÇA ENQUANTO).....</b>	<b>3</b>
SINTAXE DA ESTRUTURA DO-WHILE .....	3
EXEMPLO DA INSTRUÇÃO.....	3
<b>EXEMPLO COMPLETO .....</b>	<b>5</b>
UML DA CLASSE NUMERO .....	5
CLASSE NUMERO .....	6
CLASSE MAIN.....	7

## INSTRUÇÃO DO WHILE (FAÇA ENQUANTO)

O laço de repetição **do while** caracteriza-se por ter seu teste de execução no final do *loop*. Este tipo de laço sempre irá executar pelo menos uma vez a instrução, e após essa execução a condição é testada para ver se acontecerá um próximo *loop*.

### Sintaxe da estrutura do-while

```
do{
    <instruções>;
}while (condição);
```



**Condição** – O do while executará as instruções enquanto a condição nos parênteses for verdadeira! No momento que a condição for falsa ele para de executar as instruções dentro do bloco *do while*. Observe que na sua primeira execução ele não analisa nenhuma condição: executa primeiro e depois analisa.

**Instruções** – Dentro de um do while podemos utilizar qualquer tipo de instrução, ou seja, podemos: mostrar uma mensagem, declarar variáveis, utilizar *if()*, utilizar um *while()* e assim por diante.

### Exemplo da instrução

Vamos fazer o mesmo exemplo da rotina de sua semana utilizando o *do while*. Imagine sua rotina da semana, durante a semana você acorda, toma café, pega o ônibus e chega ao trabalho. Se fôssemos pensar em um *loop*, como seria?

#### Português:

```
do{
    Acordar
    tomar café
    Pegar o ônibus
    trabalhar
    Voltar para casa
}while (não chegar no fim de semana){
```

#### Sintaxe no programa:

```
Pessoa p1 = new Pessoa();
int dia = 2; //2 significa segunda-feira
do{ //faça os comandos a seguir
    p1.acordar();
    p1.tomarCafe();
    p1.pegarOnibus();
    p1.trabalhar();
```

```
pl.voltarParaCasa();
dia = dia+1; //isso significa que no final de cada loop, irá aumentar um
dia na semana
}while(dia<7){//enquanto o dia for menor que 7
System.out.println("Finalmente fim de semana!");
```

Analisando o exemplo acima, podemos perceber que iniciamos a semana na segunda, onde definimos dia = 2. O *do while* irá executar as instruções sem testar nada, após sua primeira execução ele irá testar se o dia é menor que 7 (sábado). Se for, retorna à linha do “do” e executa novamente.

Este exemplo executará as mesmas ações que o exemplo mostrado na apostila anterior, onde aprendemos o *while*, a diferença é o **while** sempre testará a condição antes de executar, enquanto o **do while** executa uma vez o código e depois testa a condição.

#### **While x do while**

*Ambos são laços de repetição, porém o **while** sempre testará a condição antes de executar, enquanto o **do while** executa uma vez o código e depois testa a condição.*

Vamos considerar o exemplo de uma classe Numero, tendo como atributo valor privado, seus métodos set/get e um método para ver o fatorial do número. Lembre-se que o fatorial é a multiplicação do número com todos os seus antecessores. Exemplo: **Fatorial do número 4 é 24**, pois é  $1*2*3*4 = 24$ . Porém neste exemplo utilizaremos a instrução **do while**.

#### **UML da classe Numero:**

Numero
-valor: int
+getValor():int
+setValor(valor:int):void
+calcularFatorial():int

```

1 public class Numero{
2     private int valor;
3
4     public void setValor(int valor){
5         this.valor = valor;
6     }
7     public int getValor(){
8         return this.valor;
9     }
10    public int calcularFatorial(){
11        int cont = 1;
12        int fatorial = 1;
13        do{
14            fatorial = fatorial * cont;
15            cont = cont + 1;
16        }while(cont<=this.valor);
17        return fatorial;
18    }
19 }

```

## EXEMPLO COMPLETO

Vamos desenvolver um algoritmo que leia um número inteiro e positivo e após a leitura seu programa deverá apresentar um menu ao usuário com as seguintes opções:

- 1 - Visualizar o número digitado
- 2 - Alterar o número
- 3 - Visualizar Fatorial do número
- 4 - Visualizar somatório do número
- 5 - Visualizar quantidade de múltiplos do número
- 6 - Ver se o número é primo
- 0 - Sair

## UML da classe Numero

Numero
-valor:int
+getValor():int +setValor(valor:int):void +calcularFatorial():int +calcularSomatorio():int +contarMultiplos():int +verificarSeEPrimo():String

## Classe Numero

```

1 public class Numero{
2     private int valor;
3
4     public Numero(int valor){
5         this.valor = valor;
6     }
7     public void setValor(int valor){
8         this.valor = valor;
9     }
10    public int getValor(){
11        return this.valor;
12    }
13    public int calcularFatorial(){
14        int cont = 1;
15        int fatorial = 1;
16        do{
17            fatorial = fatorial * cont;
18            cont = cont + 1;
19        }while(cont<=this.valor);
20        return fatorial;
21    }
22    public int calcularSomatorio(){
23        int soma = 0;
24        int cont = 1;
25        do{
26            soma = soma + cont;
27            cont ++;
28        }while(cont<=this.valor);
29        return soma;
30    }
31    public int contarMultiplos(){
32        int contMultiplos = 0;
33        int cont = 1;
34        do{
35            if(this.valor%cont == 0){
36                contMultiplos++;
37            }
38            cont++;
39        }while(cont<=this.valor);
40        return contMultiplos;
41    }
42    public String verificarSeEPrimo(){
43        int contM = contarMultiplos();
44        if(contM == 2){
45            return this.valor+ " é PRIMO.";
46        }
47        else{
48            return this.valor+ " é COMPOSTO.";
49        }
50    }
51 }

```

Perceba que na classe acima utilizamos *do while* nos métodos `calcularFatorial()`, `calcularSomatorio()` e `contarMultiplos()`. Os mesmos métodos poderiam ser feitos com a instrução *while*.

*Para treinar repita a classe utilizando **while** no lugar do **do while**.*

## Classe Main

```

1 import java.util.Scanner;
2 public class Principal{
3     public static void main(String args[]){
4         Scanner ler = new Scanner(System.in);
5
6         int numero;
7         do{
8             System.out.println("Informe um número positivo: ");
9             numero = ler.nextInt();
10            if(numero<0){
11                System.out.println("Amigo, digite apenas números positivos.");
12            }
13        }while(numero<0);
14
15        Numero n1 = new Numero(numero);
16        int escolha;
17        do{
18            System.out.println("1 - Visualizar o número digitado");
19            System.out.println("2 - Alterar o número");
20            System.out.println("3 - Visualizar Fatorial do número");
21            System.out.println("4 - Visualizar somatório do número");
22            System.out.println("5 - Visualizar quantidade de múltiplos do número");
23            System.out.println("6 - Ver se o número é primo");
24            System.out.println("0 - Sair");
25            System.out.print("->"); //seta para o usuário digitar ao lado
26            escolha = ler.nextInt();
27            switch(escolha){
28                case 1:
29                    System.out.println("Número digitado " + n1.getValor());
30                    break;
31                case 2:
32                    int novoNumero;
33                    do{
34                        System.out.println("Informe o novo número: ");
35                        novoNumero = ler.nextInt();
36                        if(novoNumero<0){
37                            System.out.println("Digite apenas números positivos.");
38                        }
39                    }while(novoNumero<0);
40                    n1.setValor(novoNumero);
41                    System.out.println("Número alterado com sucesso.");
42                    break;
43                case 3:
44                    int fatorial = n1.calcularFatorial();
45                    System.out.println("O fatorial de " +n1.getValor()+ " é " +fatorial);
46                    break;
47                case 4:
48                    int somatorio = n1.calcularSomatorio();
49                    System.out.println("O somatorio de " +n1.getValor()+ " é " +somatorio);
50                    break;
51                case 5:
52                    int quantidade = n1.contarMultiplos();
53                    System.out.println("A quantidade de multiplos: " +quantidade);
54                    break;
55                case 6:
56                    System.out.println("O número " +n1.verificarSeEPrimo());
57                    break;
58                case 0:
59                    System.out.println("Sistema encerrado!");
60                    break;
61                default:
62                    System.out.println("Opcao inválida.");
63            }
64        }while(escolha!=0);
65    }
66 }

```