

# Lógica de Programação

**Unidade 5.2** – Inserindo operações nos métodos



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

## SUMÁRIO

<b>INSERINDO OPERAÇÕES NOS MÉTODOS .....</b>	<b>3</b>
COMO FICA O CÓDIGO JAVA BASEADO NO DIAGRAMA: .....	3
USO DO THIS.....	4
ENTENDENDO OS MÉTODOS DA CLASSE CRIADA .....	4

## INSERINDO OPERAÇÕES NOS MÉTODOS

Agora vamos aprender como inserir uma fórmula matemática dentro de um método de nossa classe. Uma operação matemática pode estar presente tanto nos métodos de ação quanto nos métodos de retorno.

*A fórmula pode utilizar apenas os atributos da classe, os argumentos do método e/ou valores fixos para composição do cálculo.*

Para exemplificar, vamos utilizar o diagrama de classe abaixo, que representa um objeto Funcionário, possuindo como atributos o **valorHora** e a **cargaHoraria**, e como métodos **dobrarValorHora**, **calcularSalario**, **aumentarCargaHoraria**:

Funcionario
+valorHora:double +cargaHoraria:double
+dobrarValorHora():void +calcularSalario():double +aumentarCargaHoraria(horas:double):void

- Para dobrar o valor hora do funcionário, basta multiplicar o mesmo por 2.
- Para calcular o salário, baseado nas informações disponíveis, multiplicamos o valor hora do funcionário pela sua carga horária.
- Para aumentar a carga horária, precisamos saber quantas horas serão acrescidas na carga horária atual.

### Como fica o código Java baseado no diagrama:

```
public class Funcionario{
    public double valorHora;
    public double cargaHoraria;

    public void dobrarValorHora(){
        this.valorHora = this.valorHora * 2;
    }
    public double calcularSalario(){
        return this.valorHora*this.cargaHoraria;
    }
    public void aumentarCargaHoraria(double horas){
        this.cargaHoraria = this.cargaHoraria + horas;
    }
}
```

*Observe que ao compor uma fórmula, evitamos usar números, pois não sabemos ainda qual o valor que cada atributo vai armazenar. Assim, a fórmula deve funcionar para qualquer valor que esteja armazenado nos atributos ou argumentos.*

### Uso do this

‘**This**’ significa ‘este’: utilizamos o **this** para representar o uso de um atributo ou método da classe. Em Java isso é uma convenção dentro do código da classe, sempre que utilizar um atributo dentro de um método, ou um método dentro de outro método, usar o **this** na frente do nome.

### Entendendo os métodos da classe criada

```
public void dobrarValorHora() {
    this.valorHora = this.valorHora * 2;
}
```

Atributo que será atualizado

cálculo

O sinal de “=” equivale a “recebe” ou “passa a valer”. É o sinal de **atribuição**. O comando acima é lido da seguinte forma:

*O atributo **valorHora** **recebe** o resultado do cálculo do valor atual do atributo **valorHora** multiplicado por 2.*

```
public double calcularSalario() {
    return this.valorHora * this.cargaHoraria;
}
```

Cálculo que o método deve retornar

O comando **return** indica que o que vem a seguir deve ser a resposta do método quando este for executado. Aqui não se usa o sinal de “=”. O comando acima é lido da seguinte forma:

***Retornar** o resultado do cálculo do atributo **valorHora** multiplicado pelo valor do atributo **cargaHoraria**.*

```
public void aumentarCargaHoraria(double horas) {
    this.cargaHoraria = this.cargaHoraria + horas;
}
```

Este método possui um argumento que será utilizado no cálculo. Repare que o **this** não é utilizado na frente do argumento “**horas**”, somente na frente do atributo “**cargaHoraria**”. O comando acima é lido da seguinte forma:

*O atributo **cargaHoraria** recebe o resultado do cálculo do valor atual do atributo **cargaHoraria** somado ao valor do argumento **horas**.*