



# O REI DOS PROGRAMADORES

*Grand Line do Código Java*



Nadia Castro

## **SOBRE A AUTORA**



Desenvolvedora Full Stack de Automação na Gráfica Marx, onde crio soluções que otimizam e transformam processos no setor gráfico por meio da tecnologia.

Graduada em Análise e Desenvolvimento de Sistemas pelo Instituto Federal de Santa Catarina, também acumulo formações complementares na área de programação.

Tenho verdadeira paixão por tecnologia, inovação e aprendizado contínuo — e é essa busca constante por conhecimento que impulsiona meu trabalho no desenvolvimento de soluções cada vez mais eficientes e inteligentes. Fora do trabalho, sou girls gamer, otome e não posso ver um gato na rua que quero fazer carinho ✨.

# INTRODUÇÃO

Descubra a Linguagem que move o mundo digital.

Java é uma das linguagens de programação mais populares do mundo, usada em aplicativos móveis, sistemas bancários, e-commerce e muito mais.

Neste ebook abordaremos os conceitos fundamentais de forma simples e prática.

O que você vai aprender:

- Como funciona a programação em Java
- Os tipos de dados essenciais
- Como controlar o fluxo do seu programa
- Os fundamentos da Programação Orientada a Objetos

```
java

public class MeuPrimeiroPrograma {
    public static void main(String[] args) {
        System.out.println("Olá, mundo Java!");
    }
}
```

# CAPÍTULO 1 – OS TIPOS

## Organizando as informações no Java

Em Java, toda informação precisa ter um tipo definido. É como organizar gavetas - cada uma serve para um tipo específico de objeto.

### Tipos Primitivos Essenciais:

Números Inteiros (int):

```
java
int idade = 25;
int pontuacao = 1500;
System.out.println("Idade: " + idade + " anos");
```

Números Decimais (double):

```
java
double preco = 29.99;
double temperatura = 36.5;
System.out.println("Preço: R$ " + preco);
```

Texto (String):

```
java
String nome = "Maria Silva";
String cidade = "São Paulo";
System.out.println("Nome: " + nome + ", Cidade: " + cidade);
```

Verdadeiro ou Falso (boolean):

```
java
boolean estaChovendo = true;
boolean temDesconto = false;

if (estaChovendo) {
    System.out.println("Leve um guarda-chuva!");
}
```

Exemplo Prático - Sistema de Cadastro:

```
java
public class Pessoa {
    public static void main(String[] args) {
        String nome = "João";
        int idade = 30;
        double salario = 3500.50;
        boolean temFilhos = true;

        System.out.println("=== CADASTRO ===");
        System.out.println("Nome: " + nome);
        System.out.println("Idade: " + idade + " anos");
        System.out.println("Salário: R$ " + salario);
        System.out.println("Tem filhos: " + temFilhos);
    }
}
```

# CAPÍTULO 2 – AS ESTRUTURAS

## Ensinando seu programa a tomar decisões

As estruturas de controle são como semáforos no trânsito - elas decidem qual caminho seu programa deve seguir.

Tomando Decisões com IF/ELSE:

```
java

int nota = 85;

if (nota >= 70) {
    System.out.println("Aprovado! Parabéns!");
} else if (nota >= 50) {
    System.out.println("Recuperação. Estude mais!");
} else {
    System.out.println("Reprovado. Não desista!");
}
```

Repetindo Ações com FOR:

```
java

// Contando de 1 a 5
for (int i = 1; i <= 5; i++) {
    System.out.println("Número: " + i);
}

// Calculando tabuada
int numero = 7;
for (int i = 1; i <= 10; i++) {
    System.out.println(numero + " x " + i + " = " + (numero * i));
}
```

## Repetindo com WHILE:

```
java

int contador = 0;
while (contador < 3) {
    System.out.println("Tentativa: " + (contador + 1));
    contador++;
}
```

## Exemplo Prático - Sistema de Login:

```
import java.util.Scanner;

public class SistemaLogin {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String senhaCorreta = "java123";
        int tentativas = 0;
        boolean loginSucesso = false;

        while (tentativas < 3 && !loginSucesso) {
            System.out.print("Digite a senha: ");
            String senha = scanner.nextLine();

            if (senha.equals(senhaCorreta)) {
                System.out.println("Login realizado com sucesso!");
                loginSucesso = true;
            } else {
                tentativas++;
                System.out.println("Senha incorreta. Tentativas restantes: " +
                    (3 - tentativas));
            }
        }

        if (!loginSucesso) {
            System.out.println("Acesso bloqueado!");
        }

        scanner.close();
    }
}
```

## CAPÍTULO 3 – POO

### Construindo seus próprios tipos de dados

A Programação Orientada a Objetos (POO) é como criar moldes para fabricar objetos no mundo real. Você define as características (atributos) e ações (métodos) que seus objetos podem ter.

Criando uma Classe:

```
public class Carro {  
    // Atributos (características)  
    private String marca;  
    private String modelo;  
    private int ano;  
    private boolean ligado;  
  
    // Construtor (como criar o objeto)  
    public Carro(String marca, String modelo, int ano) {  
        this.marca = marca;  
        this.modelo = modelo;  
        this.ano = ano;  
        this.ligado = false;  
    }  
}
```



```
// Métodos (ações que o objeto pode fazer)
public void ligar() {
    if (ligado) {
        ligado = true;
        System.out.println("Carro ligado! Vrum vrum!");
    } else {
        System.out.println("O carro já está ligado!");
    }
}

public void desligar() {
    if (ligado) {
        ligado = false;
        System.out.println("Carro desligado.");
    } else {
        System.out.println("O carro já está desligado!");
    }
}

public void acelerar() {
    if (ligado) {
        System.out.println("Acelerando o " + marca + " " + modelo + "!");
    } else {
        System.out.println("Primeiro ligue o carro!");
    }
}

public String getInfo() {
    return marca + " " + modelo + " (" + ano + ")";
}
}
```

## Usando a Classe:

```
public class TesteCarro {
    public static void main(String[] args) {
        // Criando objetos
        Carro meuCarro = new Carro("Toyota", "Corolla", 2020);
        Carro carroAmigo = new Carro("Honda", "Civic", 2019);

        // Usando os métodos
        System.out.println("Meu carro: " + meuCarro.getInfo());
        meuCarro.ligar();
        meuCarro.acelerar();

        System.out.println("\nCarro do amigo: " + carroAmigo.getInfo());
        carroAmigo.acelerar(); // Tentativa sem ligar
        carroAmigo.ligar();
        carroAmigo.acelerar();
    }
}
```

## Exemplo Prático - Sistema Bancário Simples:

```
public class ContaBancaria {
    private String titular;
    private double saldo;

    public ContaBancaria(String titular, double saldoInicial) {
        this.titular = titular;
        this.saldo = saldoInicial;
    }

    public void depositar(double valor) {
        if (valor > 0) {
            saldo += valor;
            System.out.println("Depósito de R$ " + valor + " realizado!");
        }
    }

    public void sacar(double valor) {
        if (valor > 0 && valor <= saldo) {
            saldo -= valor;
            System.out.println("Saque de R$ " + valor + " realizado!");
        } else {
            System.out.println("Saldo insuficiente!");
        }
    }

    public void consultarSaldo() {
        System.out.println("Saldo atual: R$ " + saldo);
    }
}

// Usando a conta
public class TesteBanco {
    public static void main(String[] args) {
        ContaBancaria conta = new ContaBancaria("Maria", 1000.0);

        conta.consultarSaldo();
        conta.depositar(500.0);
        conta.sacar(200.0);
        conta.consultarSaldo();
    }
}
```

# PARABÉNS SUA JORNADA JAVA COMEÇOU!

Este Ebook foi gerado por IA, e diagramado por humano.  
O passo a passo se encontra no meu Github .

<https://github.com/nadiacastr0/creating-ebooks-with-ai>

O conteúdo foi gerado com fins didáticos de construção, não foi realizado uma validação cuidadosa humana no conteúdo e pode conter erros gerados por uma IA.

*A programação é como aprender um idioma - quanto mais você pratica, mais fluente fica!*

**Nadia Castro**  
<https://portfolio.app>