

P. G. Gonçalves



Código Ninja

O Caminho para a
Dominação do Java



Java para Iniciantes

Comandos Essenciais

Se você está começando sua jornada com Java, entender os principais comandos é essencial para evoluir como programador. Vamos aprender os comandos básicos com exemplos práticos para facilitar o seu aprendizado.



01 IMPRIMINDO NA TELA

O comando `System.out.println()` é um dos primeiros que todo programador aprende.

System.out.println()



É usado para imprimir texto na tela. Ideal para testar o que você está fazendo no código.

Exemplo:

```
public class Exemplo {  
    public static void main(String[] args) {  
        System.out.println("Olá, Mundo!"); // Imprime na tela  
    }  
}
```

Explicação: Esse código imprime a mensagem "Olá, Mundo!" na tela, o que é comum em programas iniciais.



02

DECLARAÇÃO DE VARIÁVEIS

int, String, double

Variáveis



Em Java, você precisa declarar o tipo da variável antes de usá-la. Isso ajuda o computador a entender o que você quer fazer com ela.

Exemplo:

```
public class Exemplo {  
    public static void main(String[] args) {  
        int idade = 25; // Variável inteira  
        String nome = "Carlos"; // Variável de texto  
        double salario = 3000.50; // Variável de ponto flutuante  
  
        System.out.println(nome + " tem " + idade + " anos e ganha R$ " + salario);  
    }  
}
```

Explicação: Aqui, declaramos três variáveis e imprimimos seus valores na tela. O tipo de cada variável é importante para garantir que ela armazene o tipo correto de dados.



03.

ESTRUTURA CONDICIONAL:

if e else

Condicionais



Com a estrutura condicional, podemos tomar decisões dentro do código. O if verifica uma condição, e o else executa um código alternativo quando a condição não é verdadeira.

Exemplo:

```
public class Exemplo {  
    public static void main(String[] args) {  
        int idade = 18;  
  
        if (idade ≥ 18) {  
            System.out.println("Você é maior de idade.");  
        } else {  
            System.out.println("Você é menor de idade.");  
        }  
    }  
}
```

Explicação: Neste exemplo, verificamos se a idade é maior ou igual a 18. Se for, o programa imprime "Você é maior de idade". Caso contrário, imprime "Você é menor de idade"



04

LAÇO DE REPETIÇÃO

for

Loop



O laço for é usado quando sabemos o número de repetições de antemão. Ele é ótimo para percorrer listas ou realizar ações repetitivas.

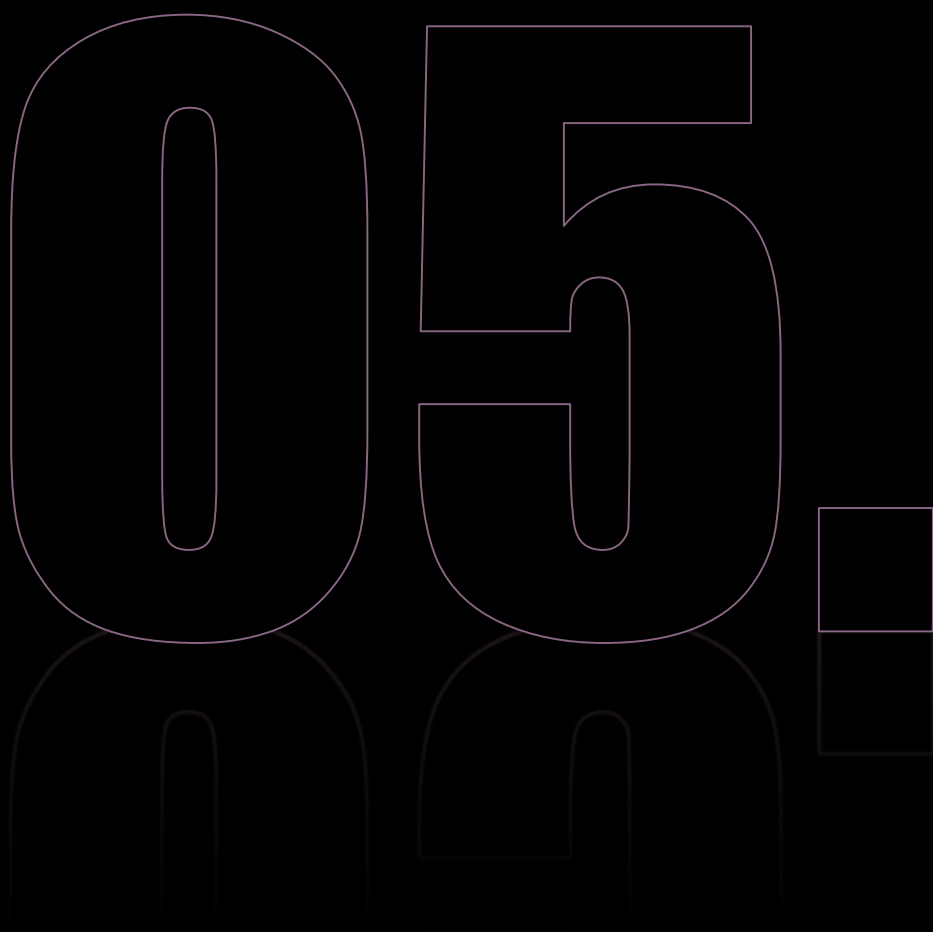
Exemplo:

```
public class Exemplo {  
    public static void main(String[] args) {  
        for (int i = 1; i ≤ 5; i++) {  
            System.out.println("Contagem: " + i);  
        }  
    }  
}
```

Explicação: Esse código vai imprimir os números de 1 a 5.

A estrutura do for consiste em inicializar a variável *i*, definir a condição para continuar ($i \leq 5$) e incrementar *i* após cada repetição.





MÉTODOS

Criando e usando

Variáveis



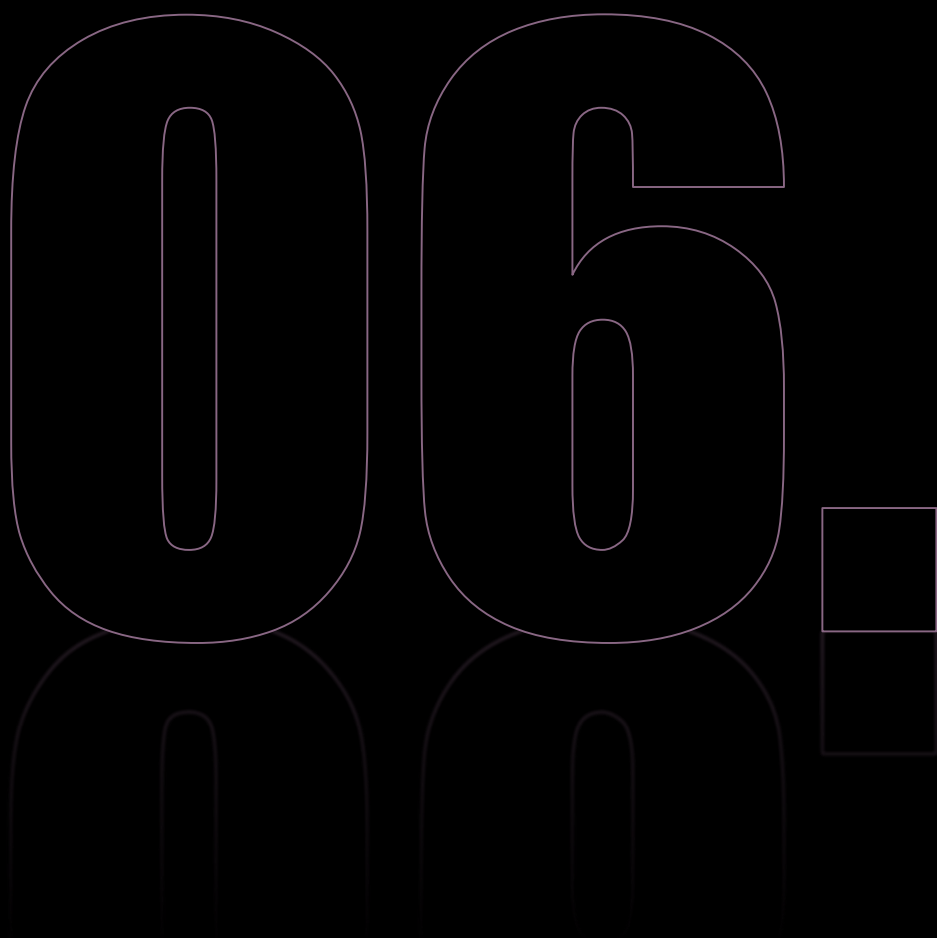
Em Java, podemos organizar nosso código criando métodos. Eles ajudam a dividir tarefas em blocos menores e reutilizáveis.

Exemplo:

```
public class Exemplo {  
    public static void main(String[] args) {  
        saudacao(); // Chamando o método  
    }  
  
    public static void saudacao() {  
        System.out.println("Olá, bem-vindo ao Java!");  
    }  
}
```

Explicação: Criamos um método chamado `saudacao()` que, quando chamado, imprime uma mensagem na tela. Isso ajuda a evitar repetir o mesmo código em várias partes do programa.





ARRAYS

Usando Arrays para Armazenar Dados

Arrays



Arrays são usados para armazenar múltiplos valores de um mesmo tipo em uma única variável. Isso é útil quando você tem uma lista de itens e precisa manipulá-los.

Exemplo:

```
public class Exemplo {  
    public static void main(String[] args) {  
        String[] frutas = {"Maçã", "Banana", "Laranja"};  
  
        for (int i = 0; i < frutas.length; i++) {  
            System.out.println("Fruta: " + frutas[i]);  
        }  
    }  
}
```

Explicação: Nesse exemplo, criamos um array de frutas e usamos um laço for para imprimir cada uma delas.





SCANNER

Lendo Dados do Usuário

Scanner



Com o comando Scanner, podemos ler dados inseridos pelo usuário no console. Isso é importante para tornar seu programa interativo. **Exemplo:**

```
import java.util.Scanner;

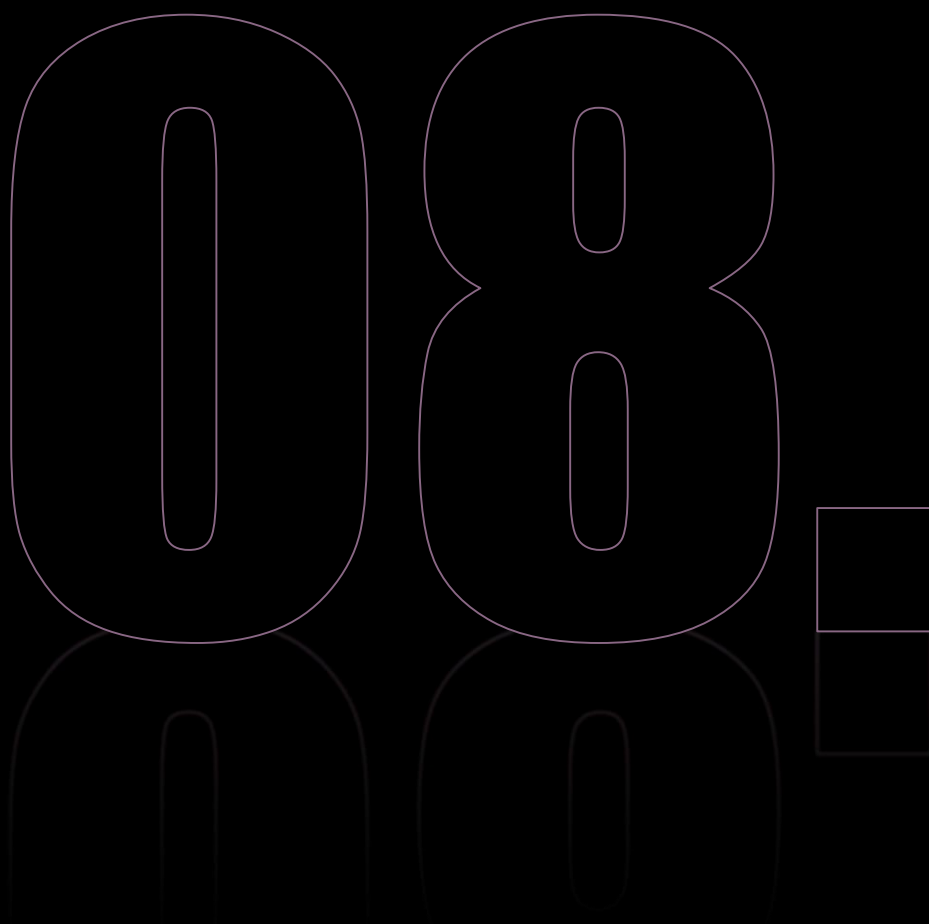
public class Exemplo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Digite seu nome: ");
        String nome = scanner.nextLine(); // Lê o nome do usuário

        System.out.println("Olá, " + nome + "!");
    }
}
```

Explicação: Aqui, pedimos para o usuário digitar seu nome, que é lido com o `scanner.nextLine()`, e depois o programa responde com uma saudação personalizada.





SWITCH

Alternativa para Múltiplas Condições

Switch



O switch é útil quando temos várias opções para comparar uma variável, sendo mais legível do que usar múltiplos if e else.

Exemplo:

```
public class Exemplo {  
    public static void main(String[] args) {  
        int dia = 3;  
  
        switch (dia) {  
            case 1:  
                System.out.println("Domingo");  
                break;  
            case 2:  
                System.out.println("Segunda-feira");  
                break;  
            case 3:  
                System.out.println("Terça-feira");  
                break;  
            default:  
                System.out.println("Dia inválido");  
        }  
    }  
}
```

Explicação: Dependendo do valor de dia, o programa vai imprimir o nome do dia correspondente. O break é necessário para encerrar a execução do switch.



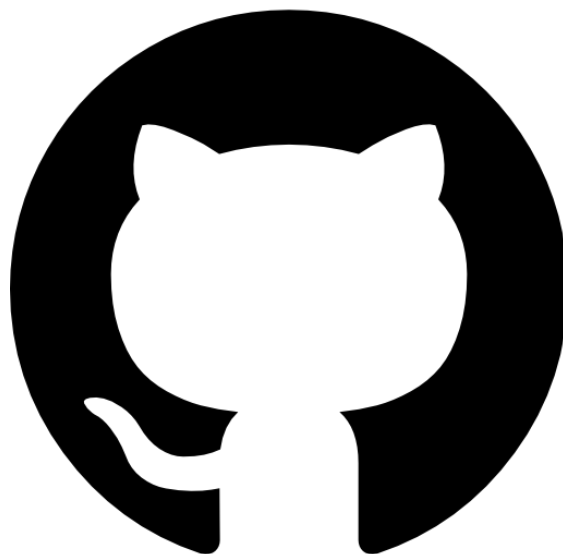
AGRADECIMENTOS

Obrigado por ler até aqui



Esse ebook foi gerado por IA, e diagramado por humano.
O passo a passo se encontra no meu GitHub.

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



<https://github.com/PedroGGoncalves/>

