



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E  
TECNOLOGIA DO RIO GRANDE DO NORTE  
CAMPUS JOÃO CÂMARA

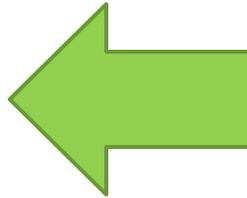
# PSEUDO-CÓDIGO

Nickerson Fonseca Ferreira  
nickerson.ferreira@ifrn.edu.br

# Introdução

2

- Um algoritmo pode ser representado em diferentes formas:
  - Linguagem natural;
  - Fluxograma;
  - Pseudo-código;



# Pseudo-Código

3

- Vamos nos aproximar mais da linguagem de programação.
- Pseudo-código é a forma intermediária entre a linguagem natural e a linguagem de programação.
- Utilizada para representar algoritmos de uma forma mais semelhante às linguagens de programação.
- O pseudo-código que iremos utilizar é o Portugol.

# Estrutura de um algoritmo

4

```
algoritmo "nome_do_algoritmo"
```

```
// Comentário são precedidos de duas barras.
```

```
// Autor: Nickerson.
```

```
// Função: O que ele faz?
```

```
var
```

```
// Declarações de variáveis...
```

```
inicio
```

```
// Comandos ...
```

```
fim
```

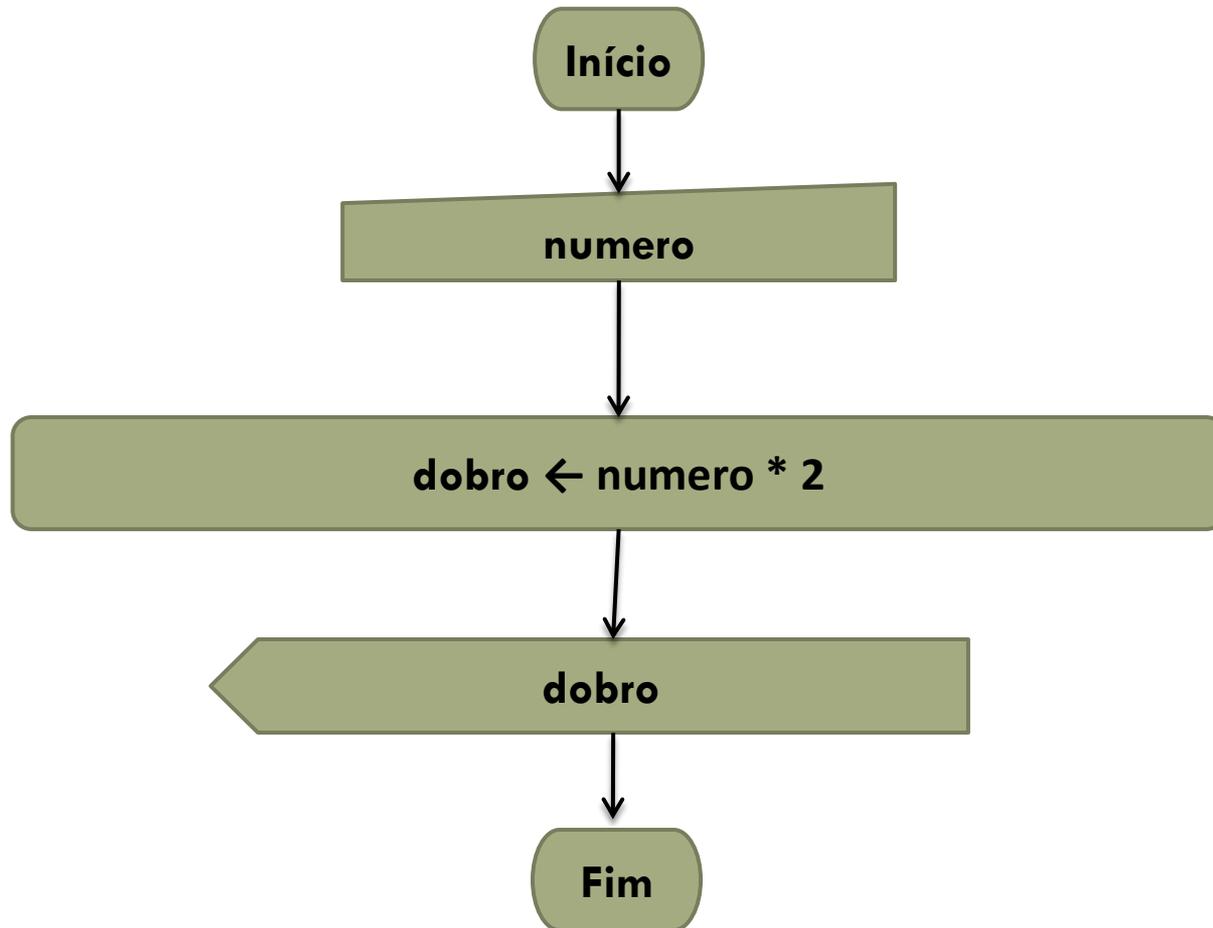
# Exemplo

5

- Escrever um algoritmo que leia um número, calcule e mostre o seu dobro.
  - Informe o número
  - O dobro é igual ao número multiplicado por 2.
  - Mostre o dobro.

# Exemplo

6



# Exemplo

7

**algoritmo** “dobro”

**var**

numero: inteiro;

dobro: inteiro;

**inicio**

leia (numero);

dobro  $\leftarrow$  numero \* 2;

escreva (dobro);

**fim**

# Comandos do Portugol

- Precisamos obter dados de entrada.
- O comando `leia()` é o comando responsável por ler os dados de entrada fornecidos pelo usuário.
  - `leia (variavel);`
  - Obter o dado informado pelo usuário e atribuir o valor para a variável.
- Outro comando importante é o de mostrar valores para o usuário.
- O comando `escreva()` exibe alguma informação para o usuário no monitor.

# Comandos do Portugol

9

- O comando `escreva()` pode ser utilizado de 3 formas:
  - `escreva(variavel);`
  - `escreva("Uma mensagem!!");`
  - `escreva("Seja Bem-vindo, ", variavel);`
- OBS: O texto deve estar sempre entre aspas.
- Existe uma variação do comando `escreva`.
  - `escreval("Mensagem!!");`
- Enquanto o comando `escreva()` vai escrevendo as mensagens numa única linha, sua variação `escreval()` vai sempre pular uma linha.

# Comandos do Portugol

10

## □ Exemplo:

- escreva("Opa!");
- escreva("Este é o meu primeiro programa!");

**Opa!Este é o meu primeiro programa!**

- escreval("Opa!");
- escreval("Este é o meu primeiro programa!");

**Opa!**

**Este é o meu primeiro programa!**

# Variáveis

- As variáveis são utilizadas para armazenar um determinado valor e que pode ser utilizada durante a execução de todo o programa.
- Cada variável precisa de um nome para identificá-la.
- Existem algumas regras para a nomenclatura das variáveis:
  - Iniciar sempre com letra;
  - Não pode possuir espaços em branco;
  - Não pode conter caracteres especiais e acentos.

# Declaração de variáveis

12

- A declaração de variáveis é feita na seção de variáveis no programa (**var**).
- Cada variável possui um nome e um tipo.
- Exemplo:

**algoritmo** “dobro”

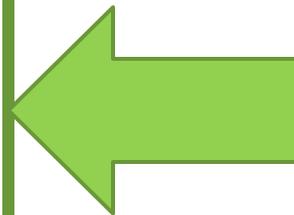
**var**

numero: inteiro;

dobro: inteiro;

**inicio**

...



# Tipos de variáveis

13

- Existem 4 tipos de variáveis:
  - inteiro: define variáveis numéricas do tipo inteiro (sem casas decimais).
  - real: define variáveis numéricas do tipo real (com casas decimais).
  - caractere: define variáveis do tipo sequência de caracteres (letras e números).
    - O valor deste tipo de variável **sempre** deve vir entre aspas.
  - logico: define variáveis do tipo lógico, ou seja, só aceita valores **VERDADEIRO** e **FALSO**.
- Quando declaramos uma variável, um espaço na memória é reservado.
- Cada tipo armazena um espaço com tamanho diferente.

# Atribuindo valores

14

- Uma variável, para ser útil, precisa de um valor.
- A operação de dar um valor para uma variável é chamada **atribuição**.
- Podemos utilizar dois operadores de atribuição:
  - ←
  - :=
- Outra forma de atribuição é através do comando de entrada.
  - `leia(variavel);`

# Atribuindo valores

15

□ Exemplo:

**algoritmo** “dobro”

**var**

numero: inteiro;

dobro: inteiro;

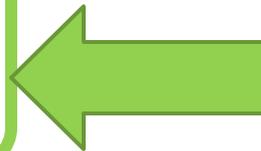
**inicio**

leia (numero);

dobro  $\leftarrow$  numero \* 2;

escreva (dobro);

**fim**



# Operadores aritméticos

16

- Operadores utilizados em cálculos matemáticos.

Sinal	Descrição
+	Operador aritmético tradicional de soma.
-	Operador aritmético tradicional de subtração.
*	Operador aritmético tradicional de multiplicação.
/	Operador aritmético tradicional de divisão. O resultado pode ser um número real.
\	Operador de divisão inteira. O resultado sempre é um inteiro.
%	Operador de módulo ou resto da divisão.
^	Operador de exponenciação.

# Operadores aritméticos

17

```
resultado := 2 + 2 * 3;  
escreva(resultado);
```

```
resultado := (2 + 2) * 3;  
escreva(resultado);
```

Utilize os parênteses para evitar ambiguidade.

# Exercício

18

1. Desenvolva um algoritmo, utilizando pseudo-código, que leia um número inteiro e imprima o seu antecessor e seu sucessor.
2. Elabore um algoritmo que leia uma velocidade em Km/h (valor inteiro) e transforme em m/h.
  - ▣ Resultado =  $V * 0,62$
3. Crie um algoritmo que leia o valor do salário mínimo e o valor do salário de um usuário, calcule a quantidade de salários mínimos esse usuário ganha e imprima o resultado. (1 SM = R\$788,00)

# Comandos de desvio

- Também conhecidos como comandos de decisão, são técnicas bastante utilizadas dentro da programação.
- Alteram o fluxo do programa.
- Fazem com que o programa proceda de uma forma ou de outra.
- Os desvios dependem de **decisões lógicas** tomadas anteriormente.

# Comandos de desvio

20

- Os principais comandos de desvio são:
  - “se entao”
  - “se entao senao”
- O comando condicional “se” executa um bloco de instruções caso uma expressão lógica seja **VERDADEIRA**.
- Já o comando “senao” executa um bloco de instruções caso uma expressão lógica seja **FALSA**.

# Comandos de desvio

21

Sintaxe:

**se** *<expressão lógica>* **entao**  
    *<sequencia de comandos>*  
**fimse**

---

**se** *<expressão lógica>* **entao**  
    *<sequencia de comandos>*  
**senao**  
    *<sequencia de comandos>*  
**fimse**

# Comandos de desvio

22

- Cada comando “se” só pode ter um “senao”.

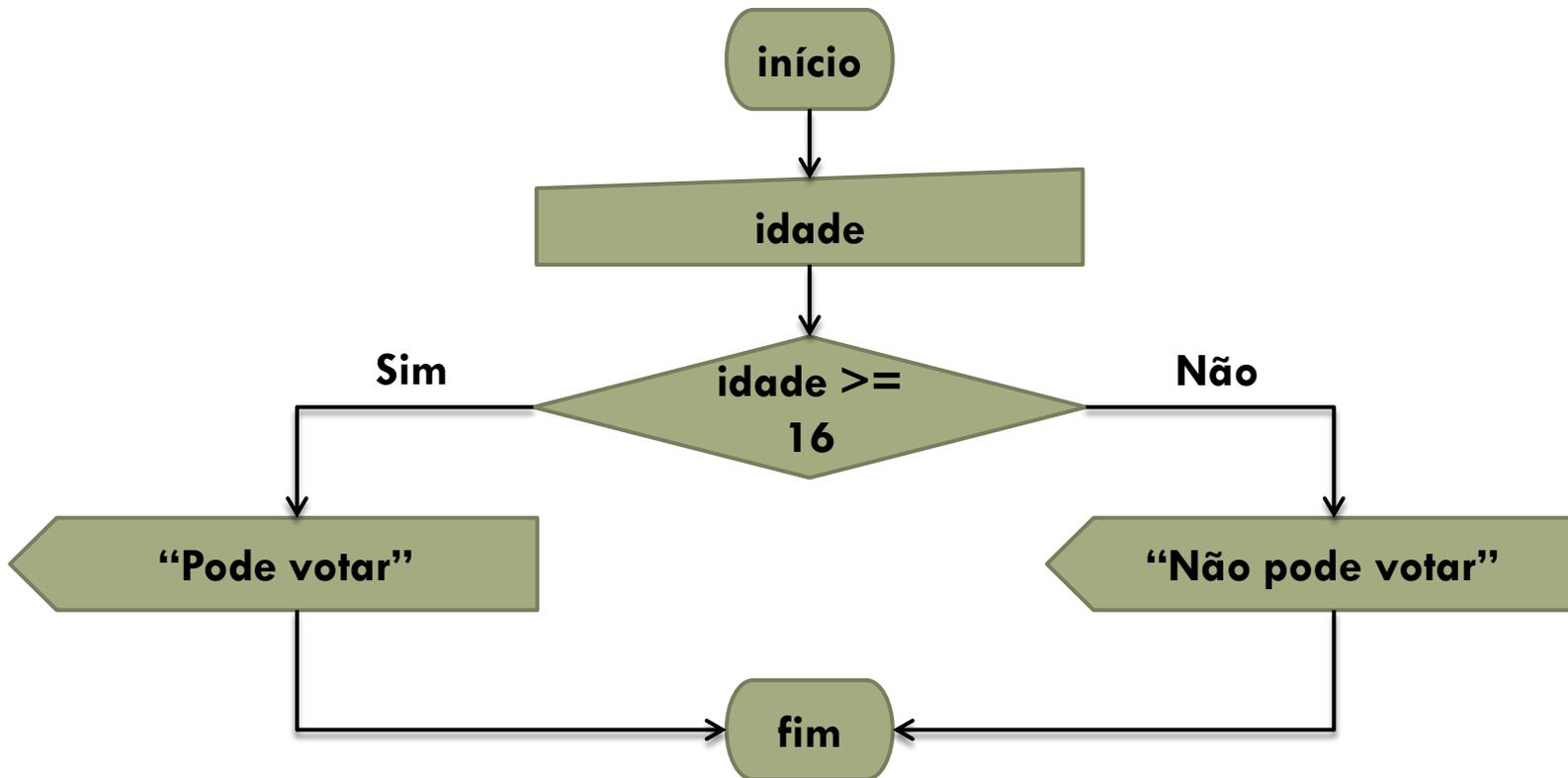
**Considere o exemplo:** como determinar se uma pessoa pode votar ou não?

- Se a pessoa tiver mais de 16 anos ela pode votar...
- Senão, ela não pode votar

# Comandos de desvio

23

- Utilizando fluxograma...



# Comandos de desvio

24

- E em português...

```
algoritmo "verifica_votacao"  
var  
    idade : inteiro  
inicio  
    escreva ("Digite a sua idade: ")  
    leia (idade)  
    se (idade >= 16) entao  
        escreva ("Pode votar")  
    senao  
        escreva ("Não pode votar")  
    fimse  
fim
```

# Encadeamento de comandos

25

- Os comandos “se” e “senao” podem estar encadeados ou aninhados, ou seja, um comando dentro do outro.

- **Exemplo:**

**se** <expressão lógica> **entao**

**se** <expressão lógica> **entao**

**<sequencia de comandos>**

**fimse**

**fimse**

# Exercício

26

1. Crie um algoritmo que leia a nota de um aluno e mostre se ele está: aprovado (acima de 60), em recuperação (abaixo de 60) ou reprovado por média (abaixo de 20).
2. Crie um algoritmo que leia a idade do usuário e imprima se ele é maior de idade ou não.