



**UNIPAC**

Universidade Presidente Antônio Carlos

*Bacharelado em Ciência da Computação*

---

# *Introdução a Programação*

## **Material de Apoio**

*Parte II*

*Prof. Nairon Neri Silva*

naironsilva@unipac.br

*1º sem / 2020*

*Material cedido pela professora Livia*

# Linguagem algorítmica Portugol

---

- O PORTUGOL é uma pseudolinguagem de programação (simbiose do Português com o ALGOL e PASCAL).

# Linguagem algorítmica Portugol

---

- A ideia é permitir que com um conjunto básico de primitivas seja possível ao projetista pensar no problema e não na máquina que vai executar o algoritmo e, por outro lado, não fique muito distante desta mesma máquina. Em outra perspectiva, que o projetista possa pensar na solução do problema e que esta solução seja facilmente implementada no computador.  
(Guimarães e Lages)

# Portugol

---

- Objetivo: obter uma notação para algoritmos, a ser utilizada na definição, na criação, no desenvolvimento e na documentação de um programa.

# Identificadores e Palavras reservadas

---

- Identificadores:
  - São nomes únicos definidos pelos programadores para identificar/distinguir os elementos de um algoritmo
- Palavras Reservadas
  - São instruções primitivas que têm significados pré-determinados e fazem parte da estrutura de qualquer linguagem de programação

# Nomes de Identificadores

---

- Algumas regras para os nomes dos identificadores:
  - 1) Devem começar por um caractere alfabético;
  - 2) Podem ser seguidos por mais caracteres alfabéticos e/ou numéricos;
  - 3) Não é permitido o uso de espaço em branco ou de caracteres especiais, como: @, #, &, \*, +, ?, \$ (exceto o \_ );
  - 4) Não poderá ser uma palavra reservada a uma instrução do algoritmo;
  - 5) Devem ser significativos.
  - 6) Não pode ser repetido dentro de um mesmo algoritmo/sub-algoritmo

# Variáveis

---

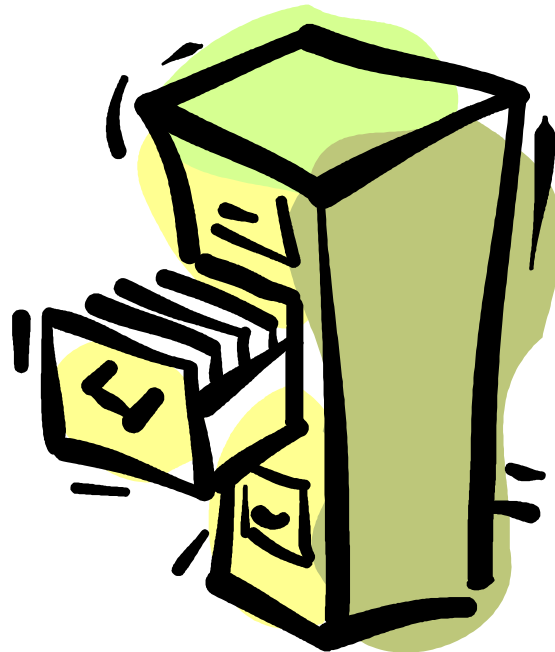
- Variável é um endereço físico da memória principal que é representado por um identificador ao qual, ao longo do seu tempo de existência, é usada para guardar um valor que pode ser modificado pelo programa.

| Endereço Físico | Identificador | Conteúdo | Tipo     |
|-----------------|---------------|----------|----------|
| 1000:2000       | Nome          | “João”   | Caracter |
| 2001:3000       | RG            | 12345    | Inteiro  |
| 3001:4000       | Salário       | 999,99   | Real     |
| 4001:5000       | Fumante       | F        | Lógico   |

# Variáveis

---

- Simplificando: Considere que a memória principal do seu computador é um armário, onde cada gaveta é a uma variável.





# Declaração de variáveis – Tipos básicos

- No Portugal, temos quatro tipos básicos, isto é, tipos básicos de dados que podem ser utilizados:

| Tipo Primitivo | Descrição                                   |
|----------------|---|
| Inteiro        | Representa o conjunto de números inteiros   |
| Real           | Representa o conjunto de números reais      |
| Caracter       | Representa um ou mais caracteres do teclado |
| Lógico         | Representa um valor lógico (V ou F).        |

- Obs: Um Caracter **SEMPRE** devem estar entre “ ”
  - EX: “A”, “Fone 3333-33333”, “1”,

# Declaração de variáveis

---

- Sintaxe para declaração de variáveis:

`<tipo de dado> <var1>;`

**ou**

`<tipo de dado> <var1, ..., varN>;`

- onde `<var1>` é o nome da variável a ser declarada (deve ser um identificador válido), e `<var1, ..., varN>` é uma lista de variáveis a serem declaradas.

# Exercícios

---

4) Identifique os erros e reescreva os identificadores abaixo:

- 13salário
- salário\$
- salário\_mínimo
- salário+reajuste
- novoSalário
- fumante?
- preço médio
- %desconto
- km/h

# Exercícios

---

5) Classifique os dados de acordo com o seu tipo, sendo (I = Inteiro, R = Real, C = Caracter e L = Lógico)

a( ) 0

b( ) + 36

c( ) 0,3257

d( ) F

e( ) 1

f( ) "F"

g( ) "+3257"

h( ) -1

i( ) 0,0

j( ) - 0,001

k( ) "-0,0"

l( ) ".F."

m( ) "o"

n( ) + 0,05

o( ) ".V."

p( ) 7/2

q( ) 32

r( ) + 3257

s( ) V

t( ) -32

u( ) "A"

v( ) "abc"

x( ) -1,9E123

z( ) "0"

# Algoritmo

---

- Exemplo:

```
inteiro idade;  
caracter nome_aluno;
```

*Com estas linhas de comando, estamos criando um espaço na memória com o nome de idade, onde poderemos armazenar valores do tipo inteiro; e um espaço com o nome de nome\_aluno onde iremos armazenar caracteres.*

# Algoritmo

---

- Sintaxe para criação de algoritmos:

`inicio`

`<declaração de variáveis>`

`<corpo do programa>`

`fim.`

# Exemplos

---

```
inicio  
  
inteiro a,b;  
  
fim.
```

```
inicio  
  
inteiro a, b;  
inteiro b; ERRO!  
  
fim.
```

```
inicio  
  
inteiro a;  
inteiro b;  
  
fim.
```

```
inicio  
  
inteiro a;  
inteiro b, c;  
  
fim.
```

# Constantes

---

- As constantes são criadas com base nas mesmas regras e tipos já vistos em variáveis.
- Diferem apenas no fato de armazenar um valor constante, ou seja, que não se modifica durante a execução de um programa.
- Assim, uma constante é uma variável com valor pré-definido que não pode ser modificado por nenhuma função de um programa.



# Declaração de Constantes

---

- Sintaxe para declaração de constantes:

```
const <identificador> = <valor>;
```

- onde <identificador> é o nome da constante a ser declarada (deve ser um identificador válido), e <valor> é o valor a ser atribuído a essa constante (que não poderá ser modificado ao longo do programa).

# Exemplos

---

```
inicio  
  
const x = 10;  
  
fim.
```

```
inicio  
  
const b = 20;  
inteiro b; ERRO!  
  
fim.
```

```
inicio  
  
const x = "Carro";  
inteiro b;  
  
fim.
```

```
inicio  
  
const b; ERRO!  
  
fim.
```

# Comentários

---

- Comentários são escritos para facilitar o entendimento posterior do algoritmo, ou parte dele. Podem aparecer em qualquer parte do algoritmo e não influenciam em nada na execução do algoritmo.

# Comentários

---

- Dica: Escreva comentários claros e objetivos no momento em que estiver escrevendo o algoritmo. Isto também facilita o entendimento do mesmo. Os comentários deverão acrescentar alguma coisa; não apenas frasear as instruções.

# Comentários

---

- Os comentários podem ser de múltiplas linhas, utilizando os caracteres especiais: /\* para iniciar um comentário e \*/ para finalizá-lo.
- Ou de uma linha utilizando os caracteres // (duas barras), neste caso até ao final da linha será comentário.
- Exemplos:

```
inteiro x;    /* isto é um  
Comentário*/
```

```
inteiro y; //isto também é um comentário
```

# Comando de atribuição

---

- Serve para atribuir um valor a uma variável.
- Operador de atribuição: ←

- Sintaxe:

`<variável> ← <expressão ou valor>;`

- A expressão do lado direito do operador é avaliada e seu resultado é armazenado na variável à esquerda, ela deve retornar o mesmo tipo da variável

# Comando de atribuição

---

- Exemplos:

```
inicio  
  
inteiro a,b;  
  
a ← 5;  
  
b ← "g"; {ERRO}  
  
fim.
```

# Expressões aritméticas

---

- Denominamos expressão aritmética aquela cujos operadores são aritméticos e cujos operandos são constantes do tipo numérico (inteiro ou real).
- Operadores aritméticos: símbolos que representam as operações básicas da matemática.



# Operadores aritméticos

---

| Operador        | Operação                            | Exemplo          | Resultado   |
|-----------------|-------------------------------------|------------------|-------------|
| <b>+</b>        | <b>Adição</b>                       | <b>4+3</b>       | <b>7</b>    |
| <b>-</b>        | <b>Subtração</b>                    | <b>4-3</b>       | <b>1</b>    |
| <b>*</b>        | <b>Multiplicação</b>                | <b>4*3</b>       | <b>12</b>   |
| <b>/</b>        | <b>Divisão</b>                      | <b>4/3</b>       | <b>1,33</b> |
| <b>MOD</b>      | <b>Resto da Divisão Inteira</b>     | <b>4 MOD 3</b>   | <b>1</b>    |
| <b>DIV</b>      | <b>Quociente da Divisão Inteira</b> | <b>4 DIV 3</b>   | <b>1</b>    |
| <b>POT(x,y)</b> | <b>Potênciação</b>                  | <b>POT (4,3)</b> | <b>64</b>   |
| <b>RAD(x)</b>   | <b>Radiciação</b>                   | <b>RAD(4)</b>    | <b>2</b>    |
| <b>+</b>        | <b>Manutenção do Sinal</b>          | <b>+- 4</b>      | <b>- 4</b>  |
| <b>-</b>        | <b>Inversão do Sinal</b>            | <b>-- 4</b>      | <b>+4</b>   |

# Prioridades

---

| Operador | Tipo    | Operação                     | Prioridade |
|----------|---------|------------------------------|------------|
| -        | Unário  | Inversão do Sinal            | 1          |
| +        | Unário  | Manutenção do Sinal          | 1          |
| RAD(x)   | Binário | Radiciação                   | 2          |
| POT(x,y) | Binário | Potênciação                  | 2          |
| DIV      | Binário | Quociente da Divisão Inteira | 3          |
| MOD      | Binário | Resto da Divisão Inteira     | 3          |
| /        | Binário | Divisão                      | 3          |
| *        | Binário | Multiplicação                | 3          |
| -        | Binário | Subtração                    | 4          |
| +        | Binário | Adição                       | 4          |

↑  
Ordem prioridade  
↑

Maior

Menor

# Exercícios

---

6) Indique qual o resultado será obtido das seguintes expressões:

- a)  $1 / 2$
- b)  $1 \text{ DIV } 2$
- c)  $1 \text{ MOD } 2$
- d)  $( 200 \text{ DIV } 10 ) \text{ MOD } 4$
- e)  $\text{POT}(5,2) + 3$
- f)  $\text{RAD}(25)+19-23$
- g)  $3,0 * 5,0 + 1$
- h)  $1/4+2$
- i)  $28,0/7+4$
- j)  $3/6,0-7$

# Exercícios

---

7) Escreva o comando de atribuição e resolva a expressão das seguintes fórmulas matemáticas:

a) 
$$X = \frac{A + \frac{B}{C}}{D - \frac{E}{F}}$$
 onde A= 2, B= 6, C = 3, D=4, E=8, F=4

b) 
$$Y = \frac{\frac{2X^2 - 3X^{(X+1)}}{2} + \frac{\sqrt{X+2}}{X}}{2^X}$$
 onde x = 2

# Exercícios

---

8) Se X possui o valor 15 e foram executadas as seguintes instruções:

$X \leftarrow X + 3;$

$X \leftarrow X - 6;$

$X \leftarrow X / 2;$

$X \leftarrow 3 * X;$

$X \leftarrow X - X + X * X / X;$

Qual será o valor armazenado em X?

# Exercícios

---

9) Dado o seguinte algoritmo, qual o conteúdo das variáveis x, y e z após a execução do mesmo?

Início

```
inteiro X, Y, Z;  
X ← 5;  
Y ← 1;  
Z ← 3;  
X ← 5 + Y * Z;  
Z ← X;  
Y ← Z ;  
Y ← X ;  
Z ← X + Y / 2 * 2;  
X ← Z;
```

fim.

# Operadores relacionais

---

- Os operadores relacionais realizam comparações entre variáveis e/ou expressões. São eles:

| Operador | Operação              | Exemplo | Resultado |
|----------|-----------------------|---------|-----------|
| =        | Igual                 | 4=3     | F         |
| >        | Maior                 | 4>3     | V         |
| <        | Menor                 | 4<3     | F         |
| >=       | Maior ou igual        | 4>=3    | V         |
| <=       | Menor ou igual        | 4<=3    | F         |
| !=       | Não igual (diferente) | 4 != 3  | V         |

# Operadores lógicos

---

- Os operadores lógicos retornam V (Verdadeiro) ou F (Falso), de acordo com seus operandos. São eles:

| Operador |
|----------|
| NÃO      |
| E        |
| OU       |



# Exemplos

- Considerando o seguinte algoritmo:

```
inicio  
  
    inteiro X, Y, Z;  
    X ← 5;  
    Y ← 1;  
    Z ← 3;  
  
fim.
```

- As seguintes expressões possuem os valores:

- $X + Y > 10$

$$5 + 1 > 10$$

$$6 > 10 \Rightarrow \textbf{(F)}$$

- $(Z > 6) \text{ E } (X - Y \geq 5)$

$$(3 > 6) \text{ E } (5 - 1 \geq 5)$$

$$\textbf{F} \text{ E } \textbf{F} \Rightarrow \textbf{(F)}$$

# Exercícios

---

10) Avalie as expressões abaixo:

- $x > y$
- $x < z$
- $x = z$
- $x + z > y$
- $x * y < z$
- $y - x \neq z$
- $(x - y > 8) \text{ E } (z + 2 = 10)$
- $(y * z \geq 30) \text{ OU } (x - y < 10)$
- $(y - z \leq 10) \text{ E } (z + y > 5) \text{ OU } (z > 9)$

```
inicio  
  
    inteiro X, Y, Z;  
    X ← 5;  
    Y ← 1;  
    Z ← 3;  
  
fim.
```

# Comandos de entrada e saída

---

- Os algoritmos precisam ser “alimentados” com dados para efetuarem as operações e cálculos que são necessários a fim de alcançar o resultado desejado. Com esta finalidade utilizaremos os comandos de entrada e saída.
- Precisamos de instruções básicas que efetuam tarefas essenciais para o recebimento e apresentação dos dados.

# Entrada de dados

---

- A instrução de entrada de dados permite que informações dos usuários sejam transferidas para a memória do computador (variáveis).
- Sua função é atribuir o dado a ser fornecido à variável identificada.
- Sintaxe:  

```
leia (<variável>);  
ou  
leia (<var1, ..., varN>);
```

# Entrada de dados

---

- Os dados são fornecidos ao computador por meio de um dispositivo de entrada (ex: teclado e mouse ) e armazenados nas posições de memória das variáveis.
- Exemplos:

```
leia (nome);
```

```
leia (nome, endereco);
```

# Exemplo

---

- Crie um algoritmo que leia o preço unitário de um produto e também a quantidade desse produto em estoque. Calcule, também, o preço total do estoque.

inicio

```
real precoUnit, precoTot;  
inteiro qtd ;
```

```
leia(precoUnit, qtd);  
precoTot ← precoUnit * qtd;
```

fim.

# Saída de dados

---

- A instrução de saída de dados é o meio pelo qual variáveis, constantes e expressões têm seus dados exibidos pelos dispositivos de saída de um computador (ex: vídeo e impressora).
- Finalidade de fornecer dados ao ambiente exterior ao algoritmo.

# Saída de dados

---

- Sintaxe:

`imprima (<variável>);`

**ou**

`imprima (<var1, ..., varN>);`

**ou**

`imprima (<texto>);`

**ou**

`imprima (<texto, var1, ..., texto>);`

- Usa-se “,” (vírgula) para concatenar (juntar) o valor de uma variável com um texto explicativo.



# Exemplo

---

inicio

```
real precoUnit, precoTot;  
inteiro qtd ;
```

```
leia(precoUnit, qtd);  
precoTot ← precoUnit * qtd;  
imprima(precoTot);
```

fim.

# Exemplos

---

- Melhorando a interface com o usuário do algoritmo.

inicio

```
real precoUnit, precoTot;  
inteiro qtd ;
```

```
imprima("Informe o preço e a quantidade: ");  
leia(precoUnit, qtd);  
precoTot ← precoUnit * qtd;  
imprima("Preço total: ", precoTot);
```

fim.

# Exercícios

---

Escreva algoritmos que:

- 11) Leia nome, idade e salário de um funcionário e exiba os mesmos dados, contudo o salário deve ser reajustado em 12%.
- 12) Leia a base e a altura de um triângulo. Em seguida, escreva a área do mesmo.
- 13) Leia 2 notas, calcule e exiba a média ponderada dessas notas. Considere nota1 com peso 6 e nota2 com peso 4.
- 14) Transforme de um valor em dólar para reais. Considere o dólar a R\$ 3,18.
- 15) Leia um valor inteiro em segundos e, depois converte-o no formato hh:mm:ss.

# Exercícios

---

- 16) Leia o preço de fábrica de um automóvel e exiba seu preço final. Considere que o preço final é igual ao preço de fábrica mais o preço dos impostos (45% do preço de fábrica) mais a percentagem do revendedor (28% do preço de fábrica).
- 17) Leia as variáveis inteiras  $n1$  e  $n2$  e troque o valor destas variáveis. Isto é,  $n1$  deve ficar com o valor de  $n2$  e  $n2$  deve ficar com o valor de  $n1$ .

# Comando SE

---

- Sintaxe:

```
SE <condição> ENTÃO  
    <comandos>  
FIM-SE
```

# Comando SE

---

- Exemplo:

```
a<-4;
```

```
b<-2;
```

```
SE a>b ENTÃO
```

```
    imprima("A é maior")
```

```
FIM-SE
```

# Comando SENÃO

---

- Sintaxe:

```
SE <condição> ENTÃO
    <comandos>
SENÃO
    <comandos>
FIM-SE
```

# Comando SE

---

- Exemplo:

```
a<-4;
```

```
b<-2;
```

```
SE a>b ENTÃO
```

```
    imprima("A é maior")
```

```
SENÃO
```

```
    imprima("A é menor")
```

```
FIM-SE
```



# Exercícios

---

- 18) Ler um valor e escrever a mensagem É MAIOR QUE 10! se o valor lido for maior que 10, caso contrário escrever NÃO É MAIOR QUE 10!
- 19) Ler dois números inteiros (considere que não serão lidos valores iguais) e escrever o maior deles.
- 20) Ler dois números inteiros (considere que não serão lidos valores iguais) e escrevê-los em ordem crescente.

# Exercícios

---

- 21) Ler o ano atual e o ano de nascimento de uma pessoa. Escrever uma mensagem que diga se ela poderá ou não realizar o exame para obter a carteira nacional de habilitação (não é necessário considerar o mês em que a pessoa nasceu).
- 22) Faça um algoritmo para ler: número da conta do cliente, saldo, débito e crédito. Após, calcular e escrever o saldo atual (saldo atual = saldo – débito + crédito). Também testar se o saldo atual for maior ou igual a zero escrever a mensagem 'Saldo Positivo', senão escrever a mensagem 'Saldo Negativo'.
- 23) Ler o nome de 2 times e o número de gols marcados na partida (para cada time). Escrever o nome do vencedor. Caso não haja vencedor deverá ser impressa a palavra EMPATE

# Referências Bibliográficas

---

- FORBELLONE, André Luiz Villar; EBERSPACHER, Henri Frederico. *Lógica de Programação*. Makron books.
- GUIMARAES, Angelo de Moura; LAGES, Newton Alberto Castilho. *Algoritmos e estruturas de dados*. LTC Editora.
- FIDALGO, Robson. *Material para aulas*. UFRPE.
- LOPES, Anita; GARCIA, Guto. *Introdução à programação – 500 algoritmos resolvidos*. Elsevier.
- <http://www.ime.usp.br/~pf/algoritmos/>