

 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PARAÍBA Campus Campina Grande</p>	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA – CAMPUS CAMPINA GRANDE		
	CURSO:	CURSO SUPERIOR ENGENHARIA DA COMPUTAÇÃO	
	PERÍODO:		TURMA:
	DISCIPLINA:	ALGORITMOS E LABORATÓRIO DE PROGRAMAÇÃO	
	PROFESSOR:	CÉSAR ROCHA VASCONCELOS	SEMESTRE LETIVO

Módulo 02 – DEFINIÇÃO DE UMA PSEUDOLINGUAGEM ALGORÍTMICA

É fundamental que o aluno tenha compreendido todos os conceitos apresentados no módulo anterior, haja vista que, a partir deste, iremos migrar nossos algoritmos de uma simples sequência de passos (e sub-passos) numerados para uma linguagem mais próxima do mundo real.

Visando facilitar ainda mais o nosso entendimento e podermos praticar a criação de algoritmos abstraindo toda a complexidade das regras de uma linguagem de alto nível real (como C ou JAVA, etc.), vamos definir, neste módulo, uma **pseudolinguagem** em português que, apesar de não padronizada, conterà diversas regras básicas de uma linguagem de programação real.

A partir de agora, será apresentada uma série de regras básicas que constarão da nossa linguagem algorítmica.

1. DADOS E SEUS TIPOS

Computadores lidam com diversos tipos de dados: **numéricos**, **cadeias de caracteres** e **lógicos**

1.1 Numéricos:

Compreendem os seguintes tipos:

- **Inteiro:** 1, 90, -25, 58, 12589, 23.659.545
 - Exemplos: idade de uma pessoa, ano, etc.
- **Real (Ponto Flutuante):** 2.0, -3.1, 5.698, 56.258, 100.01
 - Exemplos: salário ou altura de uma pessoa, etc.

1.2 Caractere (Cadeia de caracteres)

- **Alfabético:** (A..Z, a..z)
- **Dígitos:** (0..9)
- **Especiais:** ('\$', '#', '&', '>', '=', etc.)

Aspas duplas delimitam o início e o fim da cadeia. Exemplos:

"PROGRAMAÇÃO AVANÇADA" ou

"Rua do Sol, 123 - Centro" ou "MNY-4628" ou "e-mail: cesarocha@ifpb.edu.br"

1.3 Lógico (ou booleano)

- Possuem dois valores apenas: **Verdadeiro** e **Falso**.
- Exemplo: uma locadora de dvd's poderia armazenar um valor booleano associado a um produto qualquer indicando sua disponibilidade para aluguel (verdadeiro ou falso)

2. VARIÁVEL E IDENTIFICADOR

De acordo com [1], uma variável corresponde a uma posição de memória cujo conteúdo pode variar ao longo do tempo durante a execução de um algoritmo. Embora a variável possa assumir diferentes valores ao longo do programa, ela só pode armazenar um valor a cada instante.

De maneira mais específica, uma variável é o local que, durante a execução de um programa ou algoritmo, você poderá guardar valores de um determinado tipo. Estes locais serão alocados em posições de memória para você, automaticamente. O tipo associado a uma variável não poderá sofrer alterações. Apenas o seu conteúdo é dinâmico. O nome dado a uma variável também é conhecido como **identificador**.

Na prática, as linguagens de programação reais (como C ou JAVA) possuem algumas regras claras para atribuir nomes às variáveis:

- Primeiro caractere deve ser uma letra (pode ser maiúscula ou minúscula);
- Não são permitidos espaços em branco no nome da variável;
- Números serão permitidos, desde que não apareçam no primeiro caractere;
- Caracteres especiais não são permitidos (exceto o `_`);
- Nomes de tipos de dados como INTEIRO e REAL (e alguns outros que veremos em módulos posteriores) são reservados e **não devem ser usados para nomear variáveis**.

Nomes Válidos

LADO1
ALTURA
MEDIA_GERAL

Nomes inválidos

1REGISTRO
BASE*ALTURA
CODIGO POSTAL

Lembre-se: variáveis possuem um tipo. Isto significa que você só poderá guardar valores nestas variáveis que sejam do mesmo tipo (ou compatível) daquela variável.

2.1 Declaração de variáveis

Usada para alocar uma ou mais variáveis no programa. Nesta pseudolinguagem, a declaração de variáveis tem a seguinte forma:

- Lista de Identificadores (nomes separados por vírgulas)
- Insira o sinal de dois pontos “:”
- Nome do tipo (inteiro, lógico, real, caractere)

```
ano, j, k : inteiro           (três variáveis inteiras)
titulo, genero : caractere    (duas variáveis do tipo cadeia de caracteres)
preco : real                  (uma variável do tipo real)
disponivel, dvdPossuiExtras : logico (duas variáveis do tipo lógico)
```

2.2 Operações efetuadas sobre uma variável:

- Ler o valor atual da variável.
- Guardar (ou gravar) um novo valor na variável (atribuição).
 - A cada nova atribuição, o valor antigo é perdido.

```
area : real
area ← 2.5
area ← 1055.67 (adeus, 2.5)
```

```
k : inteiro
k ← 27 + area (cuidado com a perda!)
```

```
errou : logico
errou ← VERDADEIRO
```

```
nome : caractere
nome ← "Uma variável do tipo cadeia"
```

3. EXPRESSÕES ARITMÉTICAS

São expressões onde utilizamos os números inteiros ou reais como operandos junto com os operadores aritméticos, dando sempre como resultado valores numéricos[1].

3.1 Operadores aritméticos

Operador	Operação	Operandos	Resultado
+	Adição	Caractere, Inteiro, Real	Caractere, Inteiro, Real
-	Subtração	Inteiro, Real	Inteiro, Real
*	Multiplicação	Inteiro, Real	Inteiro, Real
/	Divisão Real	Inteiro, Real	Inteiro, Real
\	Divisão Inteira	Inteiro	Inteiro
MOD ou %	Resto da divisão	Inteiro	Inteiro

Exemplos:

Expressão	Resultado
1 + 2	3
"Ana" + "Maria"	"AnaMaria"
5.0 - 1	4.0
2 * 1.5	3.0
5 / 2	2.5
5 \ 2	2
5 MOD 2	1
5 % 2	1

Figura 1: Tabela de operadores aritméticos disponíveis em nossa pseudolinguagem. [1]

3.2 Prioridade

Ordem	Operador	Avaliação
1	-, +	Unários (direita para esquerda)
2	*, /	Esquerda para a direita
3	+, -	Esquerda para a direita

Obs: na dúvida, use parênteses. Será que $2+3/2$ resulta no mesmo valor de $(2+3)/2$?

4. FUNÇÕES EMBUTIDAS

São funções matemáticas já prontas à disposição¹ dos usuários para cálculos matemáticos mais comuns:

Função	Finalidade	Tipo do(s) Argumento(s)	Tipo do resultado
ABS (X)	Valor Absoluto	Inteiro ou Real	O mesmo do argumento
TRUNC (X)	Parte Inteira	Real	Inteiro
ROUND (X)	Valor Arredondado	Real	Inteiro
POT (X, Y)	Potenciação	Inteiro ou Real	Real
RAIZ (X, Y)	Radiciação	Inteiro ou Real	Real

EXEMPLOS:

Expressão	Resultado
ABS (-4)	4
ABS (1.5)	1.5
TRUNC (2.78)	2
ROUND (2.78)	3
POT (5, 2)	25.0
POT (1.5, 3)	3.375
RAIZ (64, 3)	4.0
RAIZ (2.25, 2)	1.5

Figura 2: Tabela de funções embutidas prontas em nossa pseudolinguagem. [1]

¹ Segundo a referência da pseudolinguagem definida para a ferramenta Visualg, funções embutidas não são permitidas.

Questão em sala:

1. Suponha que A, B e C sejam variáveis reais e que I, J, K sejam variáveis inteiras.
Dados A = 4, B = 6.0 e I = 3, qual seria o valor final e o tipo dos comandos seguintes:

- a) $C \leftarrow A * B - I$
- b) $K \leftarrow B / 4 * 2$
- c) $C \leftarrow B / A + 1.5$
- d) $K \leftarrow \text{TRUNC}(B / A + 4.7)$
- e) $J \leftarrow \text{ROUND}(A * (5 / I))$
- f) $K \leftarrow \text{ABS}(A - B) * 2 + 1$

5. EXPRESSÕES LÓGICAS

As expressões lógicas são aquelas realizadas utilizando-se operadores relacionais. O retorno de uma expressão lógica será sempre um valor **booleano**: VERDADEIRO ou FALSO

5.1 Operadores relacionais:

Operador	Operação
=	Igual
<	Menor
>	Maior
<=	Menor ou igual
>=	Maior ou igual
<>	Diferente

Exemplos:

Expressão	Resultado
1 = 2	FALSO
"A" <> "B"	VERDADEIRO
5.0 > 2.5	VERDADEIRO
3 <= 3	VERDADEIRO
"A" < "B"	VERDADEIRO
"joao" > "jose"	FALSO
2 + 3 <> 5	FALSO
11 < 4	FALSO
"11" < "4"	VERDADEIRO

5.2 Operadores lógicos

São utilizados para combinar expressões lógicas.

Operador	Operação
não	não (negação)
e	e (conjunção)
ou	ou (disjunção)

A tabela verdade dos operadores é dada da seguinte forma:

A	B	A e B	A ou B	não A	não B
V	V	V	V	F	F
V	F	F	V	F	V
F	V	F	V	V	F
F	F	F	F	V	V

Figura 3: Tabela Verdade resultante.

5.3 Compatibilidade entre variáveis:

- Apenas expressões ou outras variáveis de um tipo T podem ser atribuídas a uma variável do tipo T;
- Reais podem ser atribuídos a um inteiro (com perda da parte decimal) e vice-versa (sem perda alguma).

```
var_int : inteiro
var_real : real
var_int ← 27.28   (é guardado 27)
var_real ← 2      (é guardado 2.0)
```

5.4 Operações Mistas

Envolvem tipos distintos de operadores. Prioridade:

- parênteses e funções embutidas;
- operadores Aritméticos;
- operadores Relacionais;
- operadores Lógicos.

Veja a combinação de constantes e/ou variáveis aplicadas a operadores (aritméticos, lógicos, relacionais); Podem ser declaradas sozinhas ou do lado direito da atribuição. Veja exemplos:

```
UM_INTEIRO ← 1 + 2 + 3           (6 é guardado em UM_INTEIRO)
MEDIA ← (10.0 + 8.5 + 8.5)/3    (9.0 é guardado em MEDIA)
EH_MAIOR ← 9 > 10               (FALSO é guardado em EH_MAIOR)
O_DOBRO ← UM_INTEIRO*2          (12 é guardado em O_DOBRO)
UM_REAL ← UM_INTEIRO*2.0        (12.0 é guardado em UM_REAL)
```

Ao contrário das variáveis, expressões não são declaradas, elas simplesmente ocorrem dentro do algoritmo. O tipo da expressão é o tipo do valor RESULTANTE da avaliação da mesma.

```
4*5 + 8/2 = 20 + 4 = 24          (INTEIRO)
2.0*2-3+8 = 4.0 - 3 + 8 = 1.0 + 8 = 9.0 (REAL)
2 > 3 E 9 > 4 = FALSO E VERDADEIRO = FALSO (LOGICO)
10 > 2 E 9 - 4 = V E 9-4 = ???      ERRO!!! TIPOS INCOMPATÍVEIS
"ABC " + "DEF" = "ABC DEF"        (CARACTERE)
```

Lembre-se: as variáveis de uma expressão **devem** possuir um valor no momento da avaliação da expressão. Exemplos:

```
A ← 32.0
B ← 8.0
DIVISAO ← A/B

A ← 32.0
DIVISAO ← A/B (Erro! Quem é B?)
```

6. COMENTÁRIOS

São trechos escritos dentro do programa que não serão tratados pelo computador. São úteis para potencializar a legibilidade e clareza do código. Escreva-os usando os símbolos //

Ex: // este é um comentário

7. COMANDOS DE ENTRADA E SAÍDA

Comandos de E/S são bastante particulares de uma linguagem para outra.

7.1 Comando de entrada de dados:

Em nossa pseudolinguagem, será usada a seguinte sintaxe:

Leia (*Lista de entrada*)

Onde lista de entrada = variáveis separadas por vírgulas.

Importante: esta operação é destrutiva da mesma forma que a atribuição. Na verdade, é uma atribuição com a diferença de que o valor a ser atribuído a variável será retornado através de um dispositivo de entrada no momento em que o programa for rodado.

Exemplo:

A, B : inteiro	
C : real	
D : caractere	
Leia (A, B, C, D)	
12	Será atribuído a A (é o mesmo que $A \leftarrow 12$)
4	Será atribuído a B (é o mesmo que $B \leftarrow 4$)
-3.0	Será atribuído a C (é o mesmo que $C \leftarrow -3.0$)
"César"	Será atribuído a D (é o mesmo que $D \leftarrow \text{"César"}$)

7.2 Comando de saída de dados:

Escreva (*Lista de saída*)

Onde: lista de saída = variáveis, expressões ou constantes separadas por vírgulas. É mais genérico que o comando Leia. Veja os exemplos finais para maiores detalhes do uso desta rotina.

Questão em sala:

1. Indique qual o conjunto de dados passado através de um dispositivo de entrada é válido para cada um dos comandos "Leia" abaixo. Considere: A e B inteiros; X, Y e Z reais e NOME, SOBRENOME e TIPO cadeias de caracteres.

- | | | | | | | |
|------------|--------|---------|---------|-----|----------|-----|
| a) "JOAO" | 100 | 10 | "SILVA" | 90 | 2.3 | 10 |
| b) "JOSE" | 9.5 | 10 | "JOAO" | 10 | "ALUNOS" | 5.5 |
| c) "PEDRO" | 10 | "JOSE" | 9.5 | 8.5 | "ALUNOS" | |
| d) 100 | "JOAO" | "SILVA" | 10.0 | 5.5 | 4 | |

- (1) Leia (NOME, A, SOBRENOME, X, Y, TIPO)
(2) Leia (NOME, X, Y, SOBRENOME, A, Z, B)
(3) Leia (A, NOME, SOBRENOME, X, Y, Z)
(4) Leia (NOME, X, A, SOBRENOME, B, TIPO, Y)

8. FORMATO DE UM ALGORITMO

Adotar-se-á, para fins de padronização, o seguinte formato para descrever os algoritmos:

```
Algoritmo "<título do algoritmo>"
var
  <declarações de variáveis>
inicio
  <comandos>
finalgoritmo
```

9. ALGUNS EXEMPLOS FINAIS

a) Algoritmo "Média Escolar"

```
// Este algoritmo obtém alguns dados de um único aluno (nome e suas três notas )
// como entrada e procura calcular uma média aritmética.
```

var

```
    NOTA1,NOTA2,NOTA3,MEDIA : real
    NOME : caractere
```

inicio

```
    // entrada de dados
    Leia (NOME,NOTA1,NOTA2,NOTA3)

    // processamento da media
    MEDIA <- ((NOTA1+NOTA2+NOTA3)/3.0)

    // imprimir os resultados
    Escreval("O aluno ", nome, " obteve o seguinte desempenho:" )
    Escreval("Nota 1 =", NOTA1)
    Escreval("Nota 2 =", NOTA2)
    Escreval("Nota 3 =", NOTA3)
    Escreval("Media Final =", MEDIA)
```

fimalgoritmo

Obs: existe alguma vantagem se a última instrução fosse escrita como está abaixo? Ou não?

```
Escreva ("Media final= ", (NOTA1+NOTA2+NOTA3)/3 )
```

b) Algoritmo "Desconto de uma venda"

```
// Dados os valores iniciais do produto e um desconto, calcular o desconto final em reais
// e o preço final a ser pago pelo consumidor
```

var

```
    precoInicial, precoFinal, valorDesconto, descontoCalculado : real
```

inicio

```
    // ler os dados
    Escreva( "Insira o preco do produto: " )
    Leia( precoInicial )
    Escreva( "Insira o valor do desconto (em %): " )
    Leia( valorDesconto )

    //Calcular o valor do desconto.
    //Esse passo facilitará o cálculo do preço final
    descontoCalculado <- precoInicial * (valorDesconto / 100)

    // Finalmente, pode-se calcular o preço final
    precoFinal <- precoInicial - descontoCalculado

    // Exibir os resultados
    Escreval( "-----" )
    Escreval( "Os valores digitados foram: " )
    Escreval( "-----" )
    Escreval( "Preço Inicial (R$)=", precoInicial )
    Escreval( "Desconto (%)=", valorDesconto )
    Escreval( "Preço Final (R$)=", precoFinal )
    Escreval( "Você economizou (R$)=", descontoCalculado )
    Escreval( "-----" )
```

fimalgoritmo

10. CONSIDERAÇÕES FINAIS DA NOTA DE AULA

- Foram apresentados diversos conceitos e regras básicas da nossa linguagem algorítmica;
- Existem vários tipos de dados na nossa linguagem algorítmica: INTEIRO, REAL, LÓGICO e CARACTERE;
- Expressões e variáveis devem pertencer a um dos tipos acima listados;

- Variáveis têm um nome e permitem duas operações básicas: atribuição de um valor do mesmo tipo (constante, expressão ou outra variável) e leitura do valor atualmente guardado. A operação de atribuição é destrutiva.
- O tipo de uma expressão é o tipo do valor resultante da sua avaliação.
- Existem várias funções embutidas. Outras serão apresentadas em momento oportuno;
- Existem dois comandos para tratar a entrada e saída de dados. O comando Leia atribui às variáveis listadas como parâmetro os valores fornecidos através de um dispositivo qualquer de entrada. Por outro lado, o comando Escreva envia para um dispositivo de saída as constantes, expressões e variáveis presentes na sua lista de saída.

11. CRÉDITOS E BIBLIOGRAFIA UTILIZADA PARA A GERAÇÃO DESTA NOTA DE AULA:

[1] EGYPTO, Cândido. **Lógica e Algoritmos**. CEFET-PB, 2003. (Livro-texto)

[2] PEREIRA, Frederico C. G. **Algoritmos**. CEFET-PB, 2004.