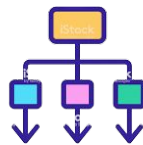


Estrutura Sequencial

Operações Básicas e Expressões



Conceito



Para enfatizar que os comandos do algoritmo executa em sequência, de cima para baixo

Um algoritmo deve obedecer uma sequência lógica adequada para cumprir seu papel.

```
x <- 10  
y <- 20  
soma <- x + y
```

correto

```
soma <- x + y  
x <- 10  
y <- 20
```

errado

Operações Básicas

Durante a construção do algoritmo, existe a necessidade de realizar operações sobre as diversas variáveis que o compõem.

Operações como atribuição, alteração e consulta sobre as variáveis são realizadas freqüentemente em toda extensão de um algoritmo.

Operações Básicas

Para realizar isso utilizamos diversos tipos de operações, como por exemplo: aritméticas, lógicas, relacionais, entre outras.

Cada tipo de operação utilizará seus respectivos conjuntos de operadores.



Expressões Aritméticas



Operadores Aritméticos

Os operadores aritméticos são usados para desenvolver operações matemáticas.

São operações que envolvem operandos do tipo inteiro ou real.

Operador	Significado
+	adição
-	subtração
*	multiplicação
/	divisão
\	divisão inteira
% ou mod	resto da divisão (inteiro)
^	potenciação (real)

Operadores Aritméticos

Operação	Símbolo	Descrição
Adição	+	Realiza a soma entre dois operandos
Subtração	-	Realiza a subtração entre dois operandos
Multiplicação	*	Realiza a mutiplicação entre dois operandos
Divisão	/	Realiza a divisão entre dois operandos
Resto	% ou MOD	Retorna o resto de uma divisão
Quociente	\	Retorna o quociente de uma divisão
Potenciação	^	Retorna o resultado de uma base elevada ao seu expoente

Operadores Aritméticos

Operação	Exemplo	Resultado
Adição	$9 + 11$	20
Subtração	$30 - 15$	15
Multiplicação	$10 * 3$	30
Divisão	$21 / 2$	10,5
Resto	$31 \% 15$	1
Quociente	$31 \setminus 15$	2
Potenciação	$3 ^ 2$	9



Precedência:

1. $^$
2. $/$, $*$, $\%$ ou Mod , \setminus
3. $+$, $-$

Operadores Aritméticos

Exemplo	Resultado
$5 + 4 * 10$	45
$(5 + 4) * 10$	90
$2 + 10 * 3 + 3$	35
$(2 + 10) * (3 + 3)$	72
$3 * 2 ^ 2$	12
$9 \% 5 \setminus 2 / 2$	1
$9 \% (5 \setminus 2) / 2$	0,5

Estando os operadores na mesma precedência, a avaliação é feita sempre da esquerda para direita.

Para modificar a ordem de avaliação das operações, é necessário usar parênteses como em qualquer expressão aritmética.

Exemplo de Expressões Aritméticas

$$2 * 6 / 3$$

Resultado = 4

$$3 + 2 * 4$$

Resultado = 11

$$(3 + 2) * 4$$

Resultado = 20

$$2 * 3 ^ 4$$

Resultado = 162

$$60 / (3 + 2) * 4$$

Resultado = 48

$$60 / ((3 + 2) * 4)$$

Resultado = 3

Operador MOD ou %

Exemplos com o operador "mod" ou %

$$14 \% 3$$

Resultado = 2

$$19 \% 5$$

Resultado = 4

Pois:

$$\begin{array}{r|l} 14 & 3 \\ \hline 2 & 4 \end{array}$$

$$\begin{array}{r|l} 19 & 5 \\ \hline 4 & 3 \end{array}$$

Operações com Caracter

São operações utilizadas para concatenar (unir) duas variáveis ou valores (constantes) do tipo texto.

“Linguagem de ” + “programação” = “Linguagem de programação”

“José ” + “ da ” + “ Silva ” = “ José da Silva”

Sejam as variáveis:

Nome= “Pedro”

Sobrenome = “Silva”

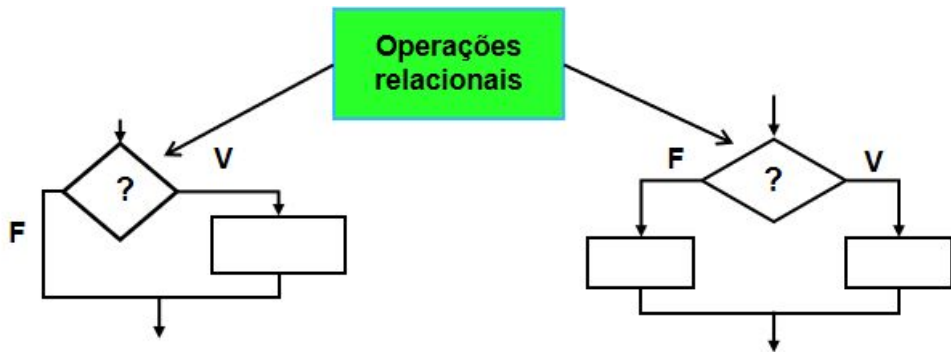
A operação Nome + “ ” + Sobrenome resultará em um conjunto de caracteres igual a “Pedro Silva”

Operações Relacionais

Os operadores relacionais são sinais que identificam uma relação entre valores, variáveis ou expressões.

O resultado de uma operação relacional, fornece apenas dois valores possíveis. O valor verdadeiro (V) ou o valor falso (F).

A utilização destes operadores permite o controle do fluxo do algoritmo.



Operadores Relacionais

Relação	Símbolo	Exemplo	Descrição
Igual	=	$a = b$	Verifica se a é igual a b
Diferente	\neq	$a \neq b$	Verifica se a é diferente de b
Maior	>	$a > b$	Verifica se a é maior que b
Menor	<	$a < b$	Verifica se a é menor que b
Maior ou Igual	\geq	$a \geq b$	Verifica se a é maior ou igual a b
Menor ou Igual	\leq	$a \leq b$	Verifica se a é menor ou igual a b

Operadores Lógicos

São operações que utilizam os operadores lógicos (verdadeiro ou falso).

O resultado de uma operação lógica, também fornecerá apenas dois valores possíveis (verdadeiro ou falso).

Operação	Representação	Resultado
Conjunção	e	Será verdadeiro (V), quando todos os operandos forem verdadeiro, caso contrário será falso (F)
Disjunção	ou	Será falso (F), quando todos os operandos forem falsos, caso contrário será verdadeiro (V)
Negação	não	Inverte o valor da expressão lógica (verdadeiro passa a ser falso e falso passa a ser verdadeiro)

Operações Lógicas

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F

p	Não p
V	F
F	V

Operações Lógicas


- `Media >= 6` **e** `Faltas <= 18`
- `Media < 6` **ou** `Faltas > 18`
- **não** `Media >= 6`

As operações acima, resultarão em **Verdadeiro** ou **Falso** a depender dos valores das variáveis (`Media` e `Faltas`)

Exercício

Sendo os valores das variáveis $X = 2$, $Y = 5$, $Z = 7$ e $Achou = \text{Falso}$, calcule o resultado das expressões:

Expressão	Resultado
$(X + 5) / Z$	
$X > Y$	
$(X > 0) \text{ ou } (Y < 7)$	
não $((X > 0) \text{ e } (Y < 7))$	
Não Achou	
$Y \setminus X$	
$Z \% X$	
$(Y \setminus X) / X$	
$(Y \% X) = Y - (Y \setminus X) * X$	



Expressão	Resultado
$(X + 5) / Z$	1
$X > Y$	Falso
$(X > 0) \text{ ou } (Y < 7)$	Verdade
não $((X > 0) \text{ e } (Y < 7))$	Falso
Não Achou	Verdade
$Y \setminus X$	2
$Z \% X$	1
$(Y \setminus X) / X$	1
$(Y \% X) = Y - (Y \setminus X) * X$	Verdade

Variável - Atribuição de Valores



Atribuição

Uma vez que você tenha declarado uma variável de qualquer tipo, um valor poderá ser armazenado na mesma, através de um comando de atribuição.

O sinal de atribuição em algoritmo “<-”.

Sintaxe

Variável <- Valor, Variável ou Expressão

Atribuição

A variável deve ter o tipo compatível com o valor, variável ou expressão que está sendo atribuída.

Lembrando que uma outra forma para atribuir (armazenar) valores a variáveis é utilizando o comando `leia (Nome_Variavel)`.

Variáveis

Variáveis

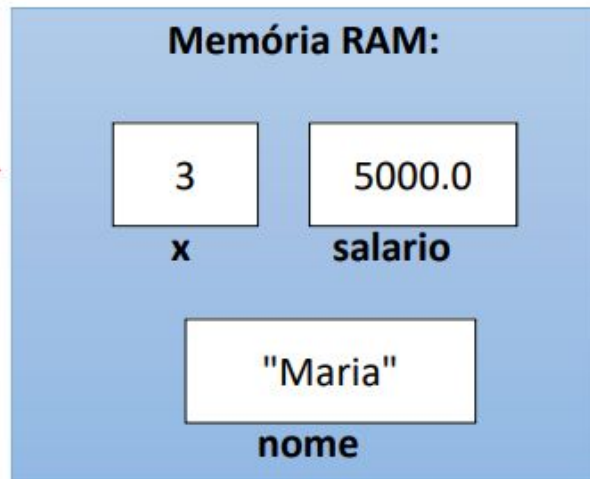
Definição informal:

Em programação, uma variável é uma porção de memória (RAM) utilizada para armazenar dados durante a execução dos programas.



(8 Giga Bytes)

Desenho
esquemático:



Variáveis

Declaração de variáveis

Sintaxe:

`<nome> : <tipo>`

Exemplos:

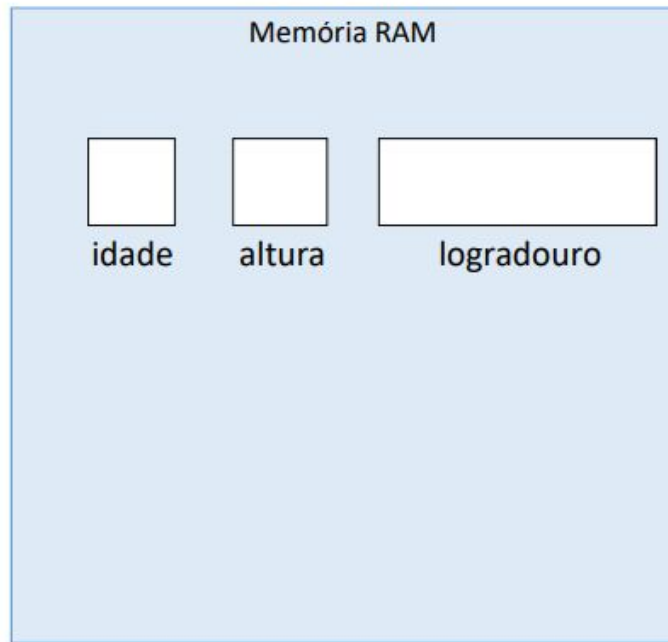
`idade : inteiro`

`altura : real`

`logradouro : caractere`

Uma variável possui:

- Nome (ou identificador)
- Tipo
- Valor
- Endereço

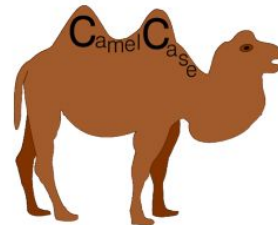


Variáveis

TIPO	DESCRIÇÃO	VALOR PADRÃO	VALORES POSSÍVEIS
inteiro	número inteiro	0	-2147483648 a 2147483647
real	número com ponto flutuante	0	-1,4024E-37 a 3,4028E+38
caractere	texto	""	(textos)
logico	valor verdade	FALSO	FALSO, VERDADEIRO

Nomes de variáveis

- Não pode começar com dígito: use uma letra ou _
- Não pode ter espaço em branco
- Não usar acentos ou til
- Sugestão: use o padrão "camel case"



Errado:

```
5minutos : inteiro  
salário : inteiro  
salário do funcionário : inteiro
```

Correto:

```
_5minutos : inteiro  
salario : inteiro  
salarioDoFuncionario : inteiro
```



Atribuição

Comando de atribuição.

Sintaxe:

<variável> <- <expressão>

Lê-se “recebe”



REGRA:

- 1) A expressão é calculada
- 2) O resultado da expressão é armazenado na variável

Atribuição

Exemplo 1

```
x, y : inteiro
```

```
x <- 5
```

```
y <- 2 * x
```

```
escreval(x)
```

```
escreval(y)
```

Exemplo 2

```
x : inteiro
```

```
y : real
```

```
x <- 5
```

```
y <- 2 * x
```

```
escreval(x)
```

```
escreval(y)
```

Atribuição

Exemplo 3

```
b1, b2, h, area : real
```

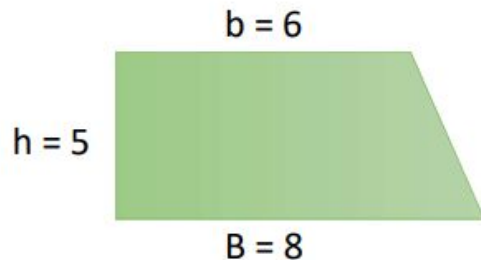
```
b1 <- 6.0
```

```
b2 <- 8.0
```

```
h <- 5.0
```

```
area <- (b1 + b2) / 2.0 * h;
```

```
escreval(area)
```



$$area = \frac{(b + B)}{2} \times h$$

No exemplo:

$$\begin{aligned} area &= \frac{(6 + 8)}{2} \times 5 \\ &= \frac{14}{2} \times 5 = 7 \times 5 = 35 \end{aligned}$$

Atribuição

Exemplo 4

a, b, resultado : inteiro

a <- 5

b <- 2

resultado <- a / b // ERRO!

escreval(resultado)

SOLUÇÃO:

resultado <- a \ b

Retorna o quociente - que é inteiro

Atribuição

Exemplo 5

```
a : real  
b : inteiro
```

```
a <- 5.0  
b <- a      // ERRO
```

```
escreval(b)
```

SOLUÇÃO:

```
b <- Int(a)
```

Transforma um número em inteiro

• Exemplo

algoritmo "Atribuição de variáveis"

var

Nota1, Nota2, Nota3 : real
Media : real

inicio

leia(Nota1)

Nota2 <- 8.5

Nota3 <- Nota2

Media <- (Nota1+Nota2+Nota3)/3

escreva(Media)

fimalgoritmo

Atribuindo um valor a uma
variável

Atribuindo uma variável a uma
outra variável

Atribuindo o resultado de uma
expressão a uma variável

Entrada - Processamento - Saída



Um programa de computador é capaz de realizar essencialmente três operações:

Entrada
de dados



Processamento
de dados



Saída
de dados



Entrada de Dados

Entrada de dados

Usuário → **Programa**
(dentro de variáveis)



Dispositivo de ENTRADA



Também chamada de
LEITURA:

"O programa está lendo dados."

Processamento de Dados

Processamento de dados

É quando o programa realiza os cálculos



O processamento de dados se dá por um comando chamado **ATRIBUIÇÃO**

```
media <- (x + y) / 2.0
```

Saída de Dados

Saída de dados

Programa → Usuário



Dispositivo de SAÍDA



Também chamada de
ESCRITA:

"O programa está escrevendo dados."

Exercício Terreno



10.0 metros

30.0 metros

$$\text{área} = \text{largura} \times \text{comprimento}$$

$$\text{preço} = \text{área} \times \text{preço do metro quadrado}$$

Digite a largura do terreno: **10.0**

Digite o comprimento do terreno: **30.0**

Digite o valor do metro quadrado: **200.00**

Area do terreno = 300.00

Preco do terreno = 60000.00

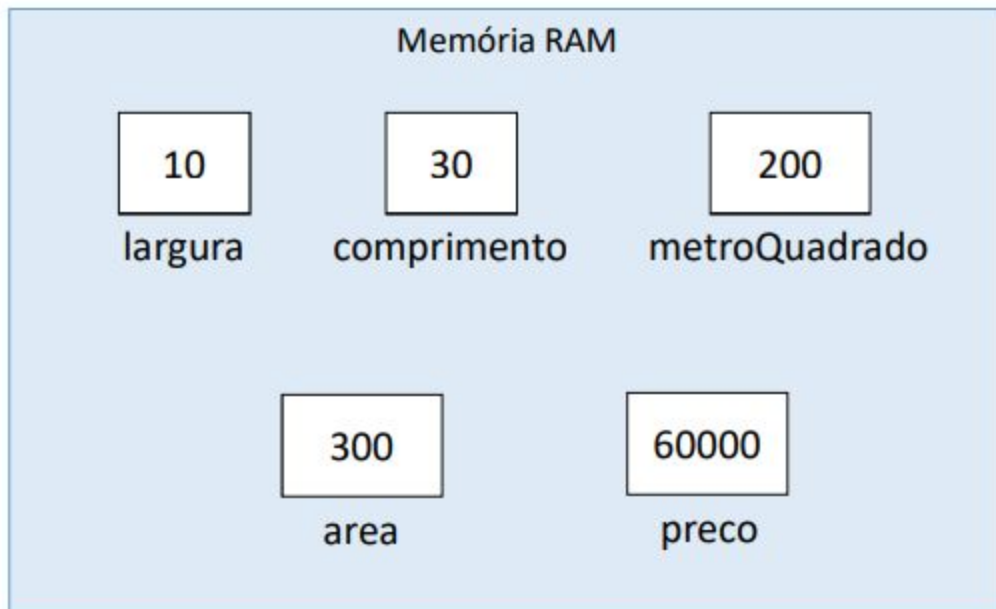
Digite a largura do terreno: **10.0**

Digite o comprimento do terreno: **30.0**

Digite o valor do metro quadrado: **200.00**

Area do terreno = 300.00

Preco do terreno = 60000.00



Terreno

Algoritmo "terreno"

Var

```
largura, comprimento, metroQuadrado : real  
area, preco : real
```

Inicio

```
escreva("Digite a largura do terreno: ")  
leia(largura)  
escreva("Digite o comprimento do terreno: ")  
leia(comprimento)  
escreva("Digite o valor do metro quadrado: ")  
leia(metroQuadrado)
```

```
area <- largura * comprimento  
preco <- area * metroQuadrado
```

```
escreval("Area do terreno = ", area:10:2)  
escreval("Preco do terreno = ", preco:10:2)
```

Fimalgoritmo