

Resumo PseudoCódigo:

VARIÁVEIS

Uma variável é um espaço na memória do computador destinado a um dado que é alterado durante a execução do algoritmo. Para funcionar corretamente, as variáveis precisam ser definidas por nomes e tipos

Variáveis no VisuAlg	
INTEIRO	Utilizado para definir números inteiros, ou seja, que não possuem pontos.
REAL	Utilizado para definir números fracionados, ou seja, possuem pontos, exemplo 35.50
CARACTERE	Utilizado para definir textos.
LÓGICO	Utilizado para definir os valores de FALSO ou VERDADEIRO
1 de ago. de 2017	

Escreva = escreve textos e variáveis no terminal sem pular a linha

Escreval = escreve textos e variáveis no terminal porem pula a linha

Exemplo:

```
var
x: inteiro
inicio
// Seção de Comandos
Escreval("escreva um número:")
Leia(x)
Escreval("x vale:", x)
```

No Terminal:



```
x vale: 1
Fim
*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

Leia = lê o valor que o usuário digitar e o insere na variável

Exemplo:

```
var
x: inteiro
inicio
// Seção de Comandos
Escreval("escreva um número:")
Leia(x)
Escreval("x vale:", x)
```

No Terminal:



```
escreva um número:
5
x vale: 5

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

ESTRUTURA DE DESIÇÃO

SE / SENÃO = é um recurso que indica quais instruções o sistema deve processar de acordo com uma expressão booleana. Assim, o sistema testa se uma condição é verdadeira e então executa comandos de acordo com esse resultado.

FIMSE = termina a estrutura se / senão

Exemplo:

```
var
x: inteiro
inicio
// Seção de Comandos
x <- 0
se x = 0 entao
    Escreval("x vale 0")
senao
    Escreval("x não vale 0")
Fimse
```

No terminal:



```
x vale 0

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

Escolha Caso = pode ser uma solução para o algoritmo quando se apresentam muitas estruturas de decisão SE-ENTÃO-SENÃO e quando as verificações são feitas caso a anterior falhe. O algoritmo executará um cálculo diferente, dependendo de qual operador o usuário escolher – Utilize o comando caso.

Exemplo:

```
var
x: inteiro
inicio
// Seção de Comandos
Escolha x
caso 0
    Escreval("x vale 0")
caso 1
    Escreval("x vale 1")
outrocaso
    Escreval("x tem o valor de ", x)
fimescolha
```

No terminal:



```
x vale 0

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

Estruturas de repetição

Para = são úteis quando se conhece previamente o número exato de vezes que se deseja executar um determinado conjunto de comandos. É uma estrutura dotada de mecanismos para contar o número de vezes que o corpo do laço (ou seja, a lista de comandos em seu interior) é executado.

Fimpara = termina a estrutura para

Exemplo:

```
var
x: inteiro
inicio
// Seção de Comandos
para x de 1 ate 10 faca
    escreval("x vale: ", x)
fimpara
```

No Terminal: 

```
x vale: 1
x vale: 2
x vale: 3
x vale: 4
x vale: 5
x vale: 6
x vale: 7
x vale: 8
x vale: 9
x vale: 10

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

REPITA ATE = é utilizada quando um conjunto de comandos deve ser executado repetidamente, até que uma seja verdadeira.

Exemplo:

```
var
x: inteiro
inicio
// Seção de Comandos
repita
    Escreval("x vale: ", x)
    x <- x + 1
ate x = 5
```

No Terminal: 

```
x vale: 0
x vale: 1
x vale: 2
x vale: 3
x vale: 4

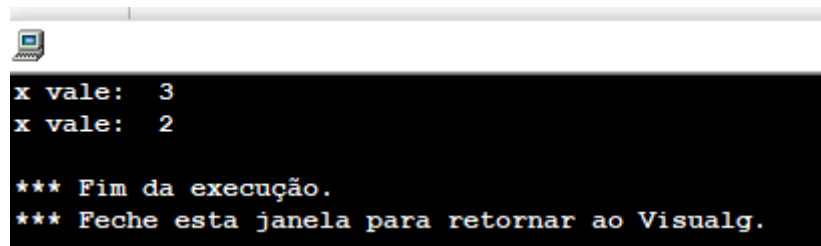
*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

Enquanto = os passos dentro deste bloco, são repetidos enquanto a expressão booleana (VERDADEIRO ou FALSO) resultar VERDADEIRO.

Exemplo:

```
var
x: inteiro
inicio
// Seção de Comandos
x <- 3
Enquanto x <> 1 faca
    escreval("x vale: ", x)
    x <- x - 1
fimenquanto
```

No Terminal:



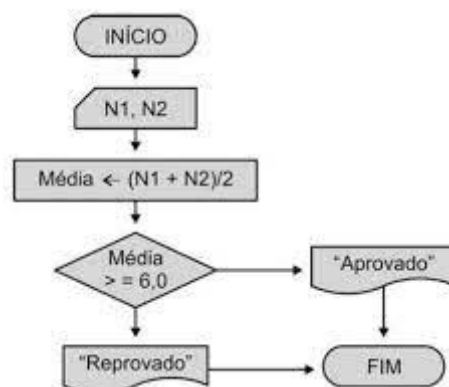
```
x vale: 3
x vale: 2

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

Fluxograma

Um fluxograma é um diagrama que descreve um processo, sistema ou algoritmo de computador. ... Fluxogramas usam retângulos, ovais, diamantes e muitas outras formas para definir os tipos de passos, assim como setas conectoras para definir fluxo e sequência.

Exemplo:



VETOR E MATRIZ

VETOR = Vetor (array uni-dimensional) é uma variável que armazena várias variáveis do mesmo tipo. No problema apresentado anteriormente, nós podemos utilizar um vetor de 50 posições para armazenar os nomes dos 50 alunos.

MATRIZ = Matriz (array multi-dimensional) é um vetor de vetores. No nosso problema, imagine uma matriz para armazenar as 4 notas de cada um dos 50 alunos. Ou seja, um

vetor de 50 posições, e em cada posição do vetor, há outro vetor com 4 posições. Isso é uma matriz.

Exemplo Vetor:

```
var
x: vetor[0..2] de inteiro
inicio
// Seção de Comandos
x[0] <- 1
x[1] <- 2
x[2] <- 3
escreval(x[2], x[0], x[1])
```

No Terminal:



```
3 1 2

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

Exemplo Matriz:

```
var
x: vetor[0..2, 0..1] de inteiro
inicio
// Seção de Comandos
x[0, 1] <- 1
x[1, 0] <- 2
x[2, 1] <- 3
escreval(x[2, 1], x[0, 1], x[1, 0])
```

No Terminal:



```
3 1 2

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

RANDI E TIMER

RANDI = Retorna um número inteiro gerado aleatoriamente, maior ou igual a zero e menor que limite.

TIMER = insere um atraso (que pode ser especificado) antes da execução de cada linha.

Exemplo RANDI:

```
var
x: inteiro
inicio
// Seção de Comandos
x <- randi(99) + 1
escreval(x)
```

No Terminal:



```
88
*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

Exemplo Timer:

```
var
x: inteiro
inicio
// Seção de Comandos
timer(1000) on
x <- 1
timer off
```