

VisualG

É uma linguagem de programação de alto nível criada para falantes do português para fins didáticos. É case-insensitive (não diferencia maiúsculas de minúsculas).

Um programa segue a estrutura:

```
Algoritmo "nome do algoritmo --- obrigatório!"  
    //Aqui se colocam informações sobre o programa (autor, data etc.)
```

```
Var  
    //Aqui se declaram variáveis
```

```
Inicio  
    // Aqui se escreve o programa
```

```
FimAlgoritmo
```

Imprimir

Para imprimir, utilizamos `escreva`("Escreva aqui dentro o que vc quer que seja impresso"), ou `escreval`("Escreva aqui para imprimir e pular uma linha").

Exemplo:

```
Algoritmo "Imprimir e pular linha"
```

```
Var
```

```
Inicio  
    escreva("Olá!")  
    escreva("Pulou linha? ")  
    escreva("E agora, pulou linha?")
```

```
FimAlgoritmo
```

Variáveis. Operações Aritméticas

Para declarar, especifique o tipo: inteiro (inteiro), real (real), caracter (letra ou palavra). Para inicializar, utilize `:=` (veja exemplos na seção sobre laços de repetição).

- Exemplo:

Algoritmo "LerNomeIdadeAltura"

Var

nome: caracter
idade: inteiro
altura: real

Inicio

escreval("Digite seu nome: ")
leia(nome)

escreval("Digite sua idade: ")
leia(idade)

escreval("Digite sua altura: ")
leia(altura)

escreva("Olá, ", nome, " você tem ", idade, " anos e ", altura, " de altura")

Fimalgoritmo

Para fazer operações aritméticas, utilizamos +, -, * e / para adição, subtração, multiplicação e divisão, respectivamente.

- Exemplo:

Algoritmo "Operações"

Var

a,b : real

Inicio

escreval("Digite um número: ")
leia(a)

escreval("Digite outro número: ")
leia(b)

escreval(a, " + ", b, " = ", a+b)

escreval(a, " - ", b, " = ", a-b)

escreval(a, " x ", b, " = ", a*b)

escreval(a, " / ", b, " = ", a/b)

Fimalgoritmo

Para divisão inteira e resto da divisão, utilizamos div e mod, respectivamente.

- Exemplo:

Algoritmo "Div e Mod"

Var

a, b: inteiro

Inicio

escreval("Digite um número inteiro: ")
leia(a)

escreval("Digite outro número inteiro: ")
leia(b)

escreval("Algoritmo de Euclides: ", a, " = ", b, " x ", a div b, " + ", a mod b)

Fimalgoritmo

Operadores e Desvios Condicionais

Para comparar dois valores, podemos utilizar: = (igual), <> (diferente), < (menor), <= (menor ou igual), > (maior), >= (maior ou igual). Também temos o desvio condicional **se e se senao** para verificar se uma condição está sendo satisfeita ou não.

Exemplo:

Algoritmo "Operadores e Desvios"

Var

a, b: inteiro

Inicio

escreval("Digite um número inteiro: ")
leia(a)

escreval("Digite outro número inteiro: ")
leia(b)

```
//teste de paridade de a
se a mod 2 = 0 entao
    escreval("O número ", a, " é par!!!")
senao
    escreval("O número ", a, " é ímpar!!!")
fimse
```

```
//teste de paridade de b
se b mod 2 = 0 entao
    escreval("O número ", b, " é par!!!")
senao
    escreval("O número ", b, " é ímpar!!!")
fimse
```

```
//comparação entre a e b
se a <> b entao
    se a > b entao
        escreva(a, " é maior do que ", b)
    senao
        escreva(a, " é menor do que ", b)
    fimse
senao
    escreva(a, " é igual a ", b)
fimse
```

Fimalgoritmo

Laços de Repetição

Para facilitar coisas repetitivas, podemos utilizar **para**, **enquanto** e **repita até**.

Algoritmo "Repetição"

Var

numero, i, j, n: inteiro

Inicio

```
escreval("Digite um número e exibiremos sua tabuada: ")
leia(numero)
```

```
escreval("Resultado com o para:")
para i de 0 até 10 faca
    escreval(numero, " x ", i, " = ", numero*i)
fimpara
```

```
j := 0
```

```
escreval("Resultado com o enquanto: ")
enquanto j <=10 faca
```

```
        escreval(numero, " x ", j, " = ", numero*j)
    j := j+1
fimenquanto
```

```
n := 0
```

```
escreval("Resultado com o repita até: ")
```

```
repita
```

```
    escreval(numero, " x ", n, " = ", numero*n)
```

```
    n := n+1
```

```
até n = 11
```

Fimalgoritmo