

# Algoritmos com VisuAlg



Prof. Gerson Volney Lagemann  
Depto. Eng. de Produção e Sistemas  
UDESC - CCT

---

# Algoritmos com VisuAlg

## Introdução

A linguagem VisuAlg é simples, seu objetivo é disponibilizar um ambiente de programação no idioma português do Brasil utilizando para tal o “Portugol” muito usado na bibliografia da área.

Entre as principais características estruturais da linguagem vale destacar:

- O VisuAlg permite apenas um comando por linha;
- Todas as palavras-chave do VisuAlg foram implementadas sem acentos, cedilha, etc.
- O VisuAlg não distingue maiúsculas e minúsculas no reconhecimento de palavras-chave e nomes de variáveis.
- É permitida a inclusão de comentários: qualquer texto precedido de "//" é ignorado, até se atingir o final da sua linha.
- O formato básico do pseudocódigo é o seguinte:

```
algoritmo "semnome"  
// Função:  
// Autor:  
// Data:  
//Seção de Declarações  
inicio  
// Seção de Comandos  
fimalgoritmo
```

## Tipos de Dados

- inteiro: define variáveis numéricas do tipo inteiro, ou seja, sem casas decimais.  
Ex. idade, número de filhos, quantidade de estados do Brasil.
- real: define variáveis numéricas do tipo real, ou seja, com casas decimais.  
Ex. salário, peso, temperatura.
- caractere: define variáveis do tipo *string*, ou seja, cadeia de caracteres.

Ex. nome, endereço, frase

- logico: define variáveis do tipo *booleano*, ou seja, com valor VERDADEIRO ou FALSO.

O VisuAlg permite também a declaração de variáveis estruturadas através da palavra-chave **vetor**.

## Nomes de Variáveis

Os nomes das variáveis devem obrigatoriamente começar por uma letra. Após a primeira letra poderá conter letras, números ou *underline* ( \_ ), até um limite de 30 caracteres. As variáveis podem ser simples ou estruturadas (vetores de uma ou duas dimensões). Variáveis não podem ter nomes iguais.

Iniciar a declaração de variáveis obrigatoriamente com o termo **var**, seguir com os nomes das variáveis separados por “,”, colocar o sinal “:” e finalmente informar o tipo daquela variável ou lista de variáveis.

**Ex.:**

```
var a: inteiro
    valor1, valor2 : real
    nome_aluno : caractere
```

**Obs:** A indentação da linha 1 para a linha 2 não é obrigatória, mas é recomendável para melhor visualização.

## Constantes

O VisuAlg tem três tipos de constantes:

- **Numéricas:** são valores numéricos escritos na forma usual das linguagens de programação. Podem ser inteiros ou reais. Neste último caso, o separador de decimais é o ponto e não a vírgula. O VisuAlg não suporta separadores de milhares.
- **Caracteres:** qualquer cadeia de caracteres delimitada por aspas duplas (").

- **Lógicos:** admite os valores VERDADEIRO ou FALSO.

### Comando de Atribuição

A atribuição de valores a variáveis é feita com o operador `<-`. Do seu lado esquerdo fica a variável à qual está sendo atribuído o valor, e à sua direita pode-se colocar qualquer expressão (constantes, variáveis, expressões numéricas), desde que seu resultado tenha tipo igual ao da variável.

**Ex.**    `a <- 3`  
          `valor1 <- 1.5`  
          `valor2 <- valor1 + a`  
          `nome_aluno <- "José da Silva"`

### Expressões Aritméticas

Operador	Operação	Exemplo	Resultado
+	Adição	3 + 2	5
-	Subtração	10 - 5	5
*	Multiplicação	3 * 7	21
/	Divisão	25 / 2	12.5
^	Potenciação	5 ^ 2	25

**OBS.:** - Não se usa o "X" para indicar a multiplicação, nem a barra horizontal de fração para indicar a divisão.

- Da mesma forma que a matemática, algumas operações têm prioridade sobre as outras. **A potenciação tem a maior prioridade entre todos os operadores. A Multiplicação e a divisão devem ser operadas antes da adição e da subtração.** Para alterar as prioridades, novamente como na matemática, devemos usar os parênteses "()". Veja alguns exemplos de atribuição e expressões aritméticas:

**Palavras Reservadas do VisuAlg**

aleatório	enquanto	funcao	para
algoritmo	entao	inicio	passo
arquivo	escolha	int	pausa
asc	escreva	inteiro	pos
ate	escreval	interrompa	real
carac	faca	leia	procedimento
caracpnum	falso	limpatela	repita
caractere	fimalgoritmo	logico	retorne
caso	fimenquanto	maiusc	se
compr	fimescolha	minusc	senao
copia	fimfuncao	mod	timer
cronometro	fimpara	nao	var
debug	fimprocedimento	numpcarac	vetor
e	fimrepita	ou	verdadeiro
eco	fimse	outrocaso	xou

**Entrada de Dados no Programa**

Para passar um dado para o programa vamos usar o comando **Leia**. Assim, se você precisa das notas das provas de um aluno para que o programa calcule a média você deve usar esta instrução para que o usuário do programa possa digitá-la e então o programa possa calcular a média. A sintaxe deste comando é:

```
Leia <Nome_De_Variável>
```

Se definíssemos as variáveis:

```
var nota1, nota2, media : real
```

então usaríamos o comando:

```
Leia nota1
Leia nota2

    ou

Leia (nota1, nota2)
```

Supondo que o usuário digitasse “7.5” teclasse ENTER e digitasse “6.5” e teclasse ENTER, o valor “7.5” seria atribuído a variável *nota1* e o valor “6.5” a *nota2*, automaticamente. Quando usar “leia(nota1, nota2)” é necessário que após digitar a “nota1” seja teclado “ENTER” e então digitada a “nota2” e teclado “ENTER” novamente.

### Saída de Dados do Programa

A lógica de qualquer programa indica que pelo menos um resultado ele deve fornecer. Podemos ter programas que não necessariamente necessitem ler algum dado para sua execução, porém não tem sentido um programa executar e não exibir nenhuma informação a respeito do que foi processado. Nos algoritmos usamos o comando **Escreva**. A sintaxe deste comando é a seguinte:

```
Escreva <Expressão1, Expressão2, etc...>
```

Expressão1, Expressão2 são valores de qualquer tipo, separados por vírgula. Estes valores podem ser constantes ou variáveis. Caso a expressão seja uma variável o que será exibido é o seu valor.

<u>Comandos (algoritmo)</u>	<u>Resultado (tela do computador)</u>
Escreva ("Alô, mundo!")	Alô, mundo!
Escreva (20)	20
var nome : character	
var idade : numerico	
nome ← "José"	
idade ← 40	

Escreva (nome)	José
Escreva ("O meu nome é ", nome)	O meu nome é José
Escreva ("Tenho ", idade, " anos")	Tenho 40 anos

Mais exemplos:

	Comando	Comentário
1	var A : inteiro	Definição de uma variável de nome A do tipo inteiro
	A ← 3	O valor de A passa a ser 3
	Escreva (A * 5)	Será mostrado o valor 15, pois A=3 e 3 * 5 = 15
2	var A, B : inteiro	Definição de duas variáveis de nome A e B do tipo inteiro
	A ← 10	O valor de A passa a ser 10
	B ← A * 3	O valor de B passa a ser 30
	Escreva (B - A + 1)	Será mostrado o valor 21 (30 - 10 + 1)
	B ← 40	O valor de B passa a ser 40
	Escreva (B - A + 1)	Será mostrado o valor 31 (40 - 10 + 1)
3	Var x inteiro	Definição de uma variável de nome x do tipo inteiro
	Leia x	Será obtido o número que será digitado pelo usuário.
	Escreva (x * 2)	Será mostrado o dobro do valor digitado pelo usuário.

## Operadores Relacionais

Nome	Símbolo
Igual	=
Diferente	≠
Menor	<
Maior	>
Menor ou igual	<=
Maior ou igual	>=

**Instrução de Decisão “SE”**

**Forma Geral:**

```
se <condição>          entao
    instrução_1
    instrução_2
    .
    .
    instrução_n
senao
    instrução_1
    instrução_2
    .
    .
    instrução_n
fimse
```

O resultado da “<condição>” é obtida pela interpretação lógica a partir da utilização de um dos operadores relacionais acima ou ainda de uma combinação entre eles.

Veja o exemplo a seguir:

```
algoritmo "media"
// Função : media
// Autor : Gerson
// Data : 09/08/2010
// Seção de Declarações
var
    n1, n2: inteiro
    m : real
inicio
// Seção de Comandos
Escreva ("Informe a nota1: ")
leia (n1)
escreva ("Informe a nota 2: ")
leia (n2)
```



```
m <- (n1 + n2) / 2
escreva ("Media = ", m)
se m < 7.0 entao
    Escreva ("Em EXAME")
Senao
    Escreva ("APROVADO")
fimse
fimalgoritmo
```

**OBS:** - O deslocamento de espaços usado no exemplo acima na instrução “se” não é obrigatória, serve somente para deixar o programa mais claro.

- É conveniente antes de ler algum dado do usuário exibir o que está sendo pedido que seja digitado ( Escreva ("Informe a nota1: ")).

A instrução “se” pode ainda aparecer sem o complemento “senao”. Veja:

**Forma Geral:**

```
se    <condição>    entao
    instrução_1
    instrução_2
    .
    .
    instrução_n
fimse
```

## Instruções de Repetição

### A Instrução de Repetição Enquanto\_Faça

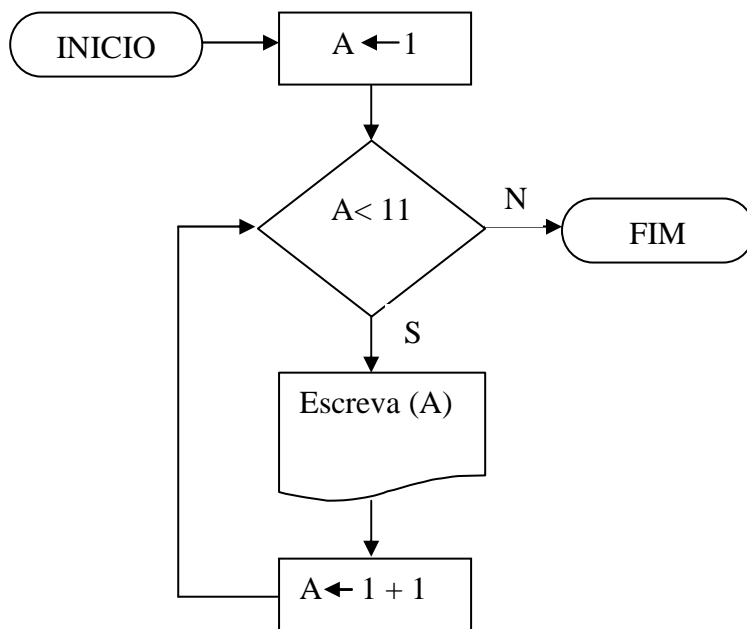
As instruções de repetição servem para que uma determinada sequência de instruções seja repetida um número finito de vezes. O número de repetições é controlado por uma instrução lógica que caso VERDADEIRA execute as instruções do laço e caso FALSA abandona o laço, seguindo para a primeira instrução após o “fimenquanto” .

**Forma Geral:**      **enquanto** <expressão\_lógica> **faca**

instrução_1	}	<b>LAÇO</b>
instrução_2		
.		
.		
instrução_n		

**fimenquanto**

Veja exemplo abaixo



```

algoritmo "lista"
// Função : lista
// Autor : Gerson
// Data : 09/08/2010
// Seção de Declarações
var
    A: inteiro
inicio
// Seção de Comandos
A ← 1
enquanto a < 11 faca
    escreva (A)
    A ← A + 1
fimenquanto
fimalgoritmo
  
```

### A Instrução de Repetição Para\_Faça

Similar a instrução “enquanto\_faça” a “**para\_faça**” serve para que uma determinada sequência de instruções seja repetida um número finito de vezes. O número de repetições é controlado por uma instrução lógica que caso VERDADEIRA executa as instruções do laço e caso FALSA abandona o laço, seguindo para a primeira instrução após o “**fimpara**”. A diferença entre as duas instruções está no incremento automático que a instrução “fimpara” apresenta.

```

Forma Geral:      para  <variável>    de    <valor_inicial>    ate
<valor_final>  <passo>  faca
                    instrução_1
                    instrução_2
                    .
                    .
                    instrução_n
                    }
fimpara          LAÇO

```

Veja exemplo abaixo.

```
algoritmo "lista"
// Função : lista
// Autor : Gerson
// Data : 09/08/2010
// Seção de Declarações
var
    A: inteiro
inicio
// Seção de Comandos
para A de 1 ate 10 passo 1 faca
    escreva (A)
fimpara
finalgoritmo
```

## Referências Bibliográficas

MAZIOLI, J. R. Apostila: **Lógica de Programação com VisuAlg**. CEET: Vila Velha – ES, 2008.

SOUZA, C. M. **VisuAlg**. <http://www.apoioinformatica.inf.br/programas> - acessado em 09/08/2010.