

Vetores ou Arrays

Já entendemos que as variáveis são os recursos que usamos para armazenar informações durante o desenvolvimento do software, e como podemos manipulá-las. Agora, vamos explorar sobre os tipos de dados podemos utilizar dentro dessas variáveis.

- **Caractere:** são sequências de caracteres que devem ser mostradas entre aspas simples ou duplas. Podemos dizer que as strings são textos que desejamos imprimir no código, por isso usamos aspas para que a cadeia de caracteres que desejamos seja impressa
- **Inteiro:** são os dados numéricos positivos ou negativos inteiros. Quando usamos o inteiro, não precisamos utilizar aspas pois o nosso código aceita o número da forma como declaramos.
- **Real:** são dados numéricos positivos ou negativos decimais. Ao contrário das regras matemáticas que aprendemos na escola, aqui utilizamos o ponto em números decimais
- **Vetor (Array):** são listas ou conjuntos capazes de guardar diversos dados ao mesmo tempo em uma única variável

```
Área dos algoritmos ( Edição do código fonte ) -> Nome do arquivo: [semnome]
1 Algoritmo "exemplo"
2 // Disciplina   : [Lógica de Programação]
3 // Professor    : Andrea Garcia Trindade
4 Var
5 nome, cidade: caractere
6 idade, quantidade: inteiro
7 altura: real
8 listadecomidas: vetor [1..5] de caractere
9
```

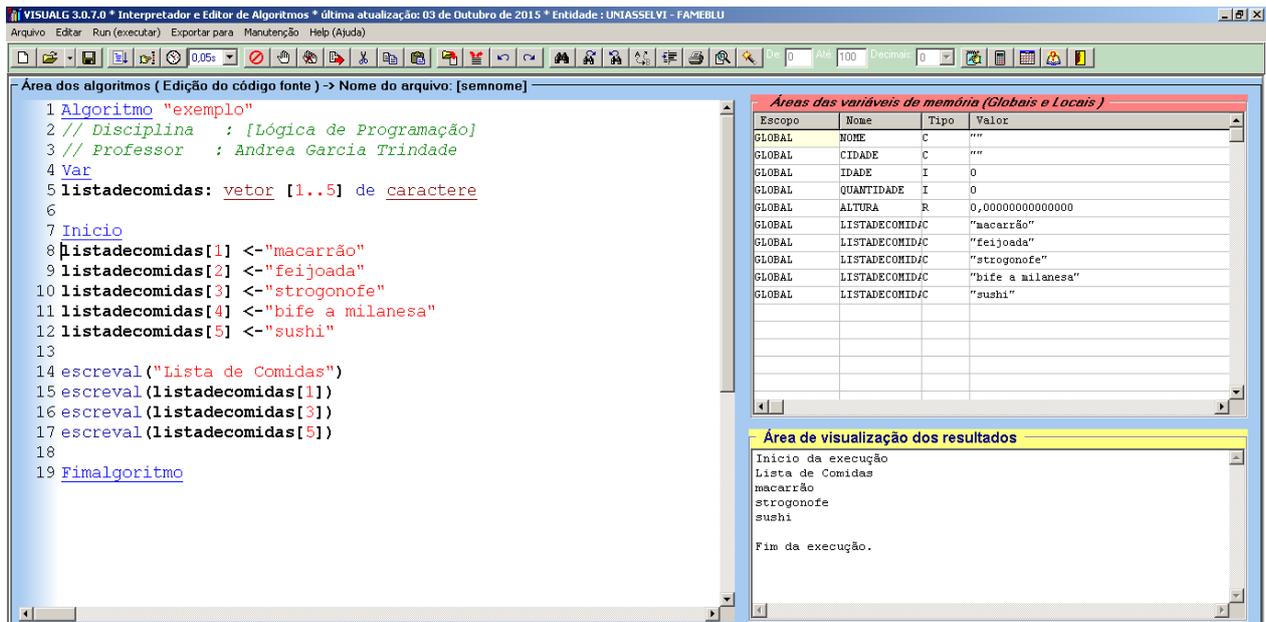
Vetores

A principal vantagem da utilização de variáveis vetorizadas ou arrays é que é definida uma variável e ele pode armazenar um conjunto de valores, evitando inúmeras variáveis no programa.

No Visualg, a definição deste tipo de variável ocorre definindo-a como vetor, seu tamanho e o tipo de informação que será colocada: caractere, inteiro ou real. Por exemplo, abaixo é criado um vetor com 5 posições caractere:

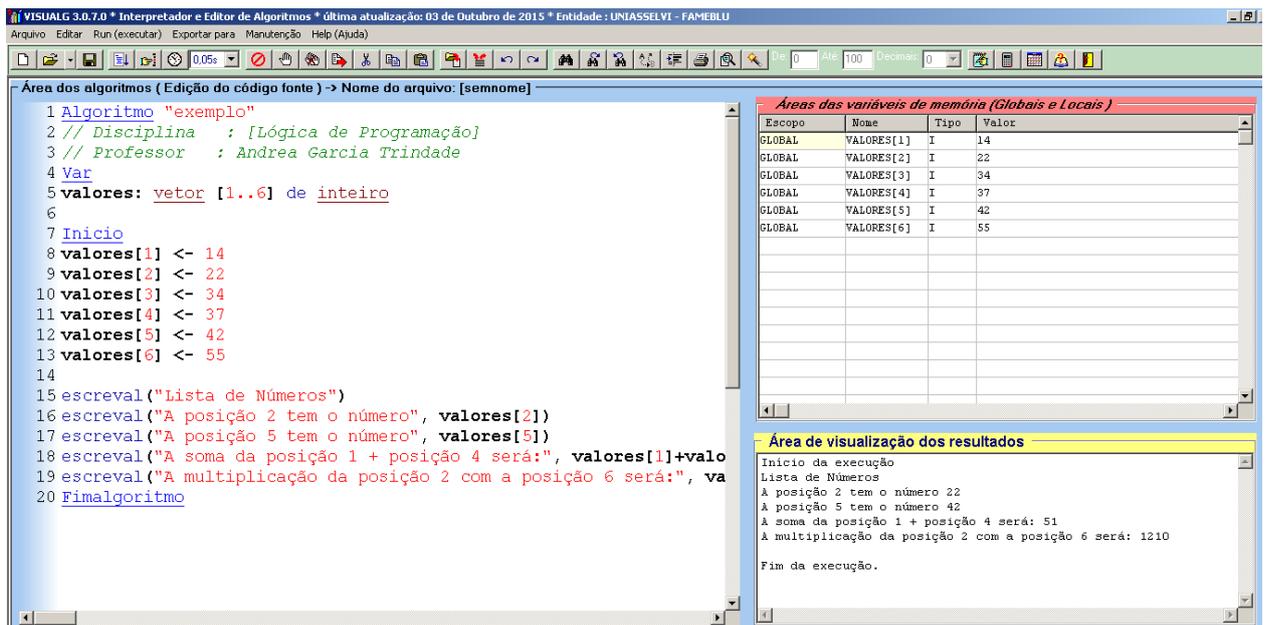
```
listadecomidas: vetor [1..5] de caractere
```

Para inserir conteúdo dentro deste tipo de variável devemos indicar a posição de cada informação. Um exemplo na programação do Visualg:



Note que neste exemplo temos a entrada de 5 posições, mas na hora de exibir eu quis exibir somente: macarrão, strogonofe e sushi. Então, foi mostrada somente as posições listadecomidas[1], listadecomidas[3] e listadecomidas[5].

Vetores podem ser utilizados com valores numéricos:



A programação este exemplo:

```

Algoritmo "exemplo"
// Disciplina : [Lógica de Programação]
// Professor : Andrea Garcia Trindade
Var
valores: vetor [1..6] de inteiro

Inicio
valores[1] <- 14
valores[2] <- 22
valores[3] <- 34
valores[4] <- 37
valores[5] <- 42
valores[6] <- 55

escreval("Lista de Números")
escreval("A posição 2 tem o número", valores[2])
escreval("A posição 5 tem o número", valores[5])
escreval("A soma da posição 1 + posição 4 será:", valores[1]+valores[4])
escreval("A multiplicação da posição 2 com a posição 6 será:", valores[2]*valores[6])
Fimalgoritmo

```

Mas da forma como está apresentada acima pode não parecer vantajoso trocar a forma como estávamos trabalhando para esta, pois todas as variáveis estão com sua entrada de dados linha a linha.

Para uma performance eficiente, os vetores devem ser trabalhados com estruturas de repetição, como por exemplo PARA...FIMPARA.

O looping deverá ser sempre na quantidade de posição que o vetor que será percorrido tenha, portanto **um vetor definido como [1..15]**, deverá ter:

Para x de 1 ate 15 faça , onde ele percorrerá todas as posições

No exemplo abaixo, o usuário deverá digitar 6 números e em seguida o programa exibirá os números digitados:

The screenshot shows the VISUAL 6.3.0.7.0 IDE interface. The main window displays the source code of the algorithm, which has been modified to use a loop for input and output. The code is as follows:

```

1 Algoritmo "exemplo"
2 // Disciplina : [Lógica de Programação]
3 // Professor : Andrea Garcia Trindade
4 Var
5 valores: vetor [1..6] de inteiro
6 x: inteiro
7
8 Inicio
9 escreval("Digite 6 Números")
10 // uma estrutura de repetição para entrada de dados
11 para x de 1 ate 6 faça
12 leia (valores[x])
13 fimpara
14 // uma estrutura de repetição para mostrar os dados cadastrados
15 para x de 1 ate 6 faça
16 escreval("Os valores digitados foram: ", valores[x])
17 fimpara
18
19
20 Fimalgoritmo

```

On the right side, the 'Áreas das variáveis de memória (Globais e Locais)' window shows the state of memory during execution:

Escopo	Nome	Tipo	Valor
GLOBAL	VALORES[1]	I	10
GLOBAL	VALORES[2]	I	20
GLOBAL	VALORES[3]	I	30
GLOBAL	VALORES[4]	I	40
GLOBAL	VALORES[5]	I	50
GLOBAL	VALORES[6]	I	60
GLOBAL	X	I	6

Below this, the 'Área de visualização dos resultados' window shows the output of the program:

```

60
Os valores digitados foram: 10
Os valores digitados foram: 20
Os valores digitados foram: 30
Os valores digitados foram: 40
Os valores digitados foram: 50
Os valores digitados foram: 60

Fim da execução.

```

```
Algoritmo "exemplo"
// Disciplina : [Lógica de Programação]
// Professor : Andrea Garcia Trindade
Var
valores: vetor [1..6] de inteiro
x: inteiro

Inicio
escreval("Digite 6 Números")
// uma estrutura de repetição para entrada de dados
para x de 1 ate 6 faça
    leia (valores[x])
fimpara
// uma estrutura de repetição para mostrar os dados cadastrados
para x de 1 ate 6 faça
    escreval("Os valores digitados foram: ", valores[x])
fimpara
Fimalgoritmo
```

Exercícios:

1. Crie um vetor com o nome meses com 12 posições e coloque dentro todos os meses do ano, não precisa ser digitado pelo usuário. Ao final mostrar, o nome dos meses 5, 8 e 10.
2. Crie uma lista com 6 números, some os elementos das posições 1, 3 e 5 e mostre o resultado.
3. Uma agência de carros quer exibir seus estoque para seus clientes. Eles colocaram os nomes dos carros em uma variável chamada listadecarros:
Listadecarros = ["Gol", "Fiesta", "Fox", "Uno", "Sander", "Onix", "Celta"]
Os dados devem ser exibidos no formato, use estrutura de repetição para mostrar:
Nome do Carro: Gol
Nome do Carro: Fiesta e assim até o último veículo
4. Uma empresa separou em uma lista, os valores dos lucros mensais dos últimos semestre. Crie um vetor chamado Listadelucro com 6 posições, contendo em cada posição o valor de recebido de cada mês (usuário digitará este valor). Seu trabalho será criar um looping (Para...fimpara) que calcule o valor total e mostre ao final do programa.