

# Noções de Programação

**Faculdade Professor  
Miguel Ângelo da Silva Santos  
FeMASS  
Prof. Sérgio Netto**

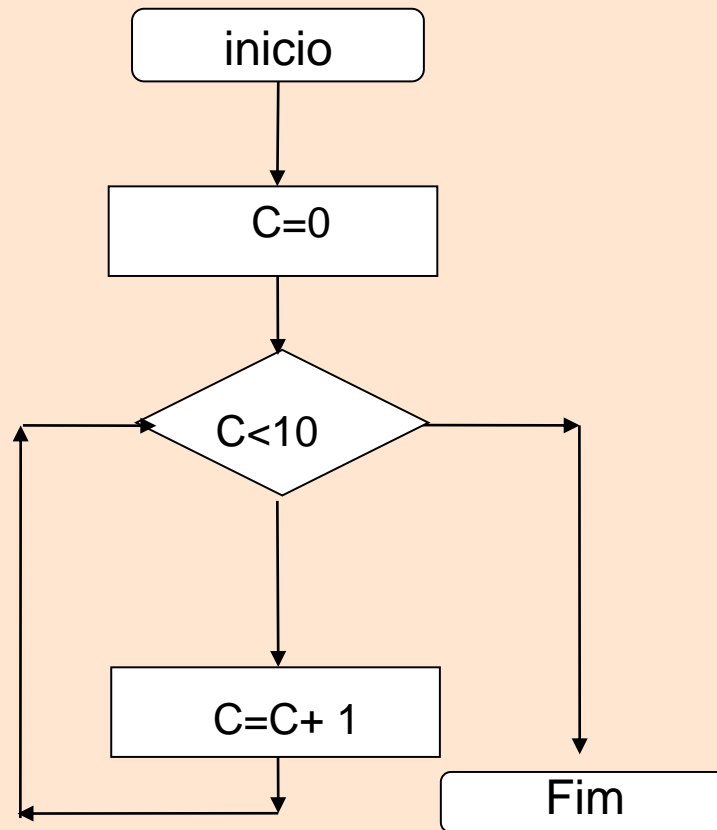
# Noções de Programação

## Estrutura de Repetição.

**Quando vamos resolver um problema no computador quase sempre vamos precisar ler um conjunto de dados, para isso utilizamos as estruturas de repetição.**

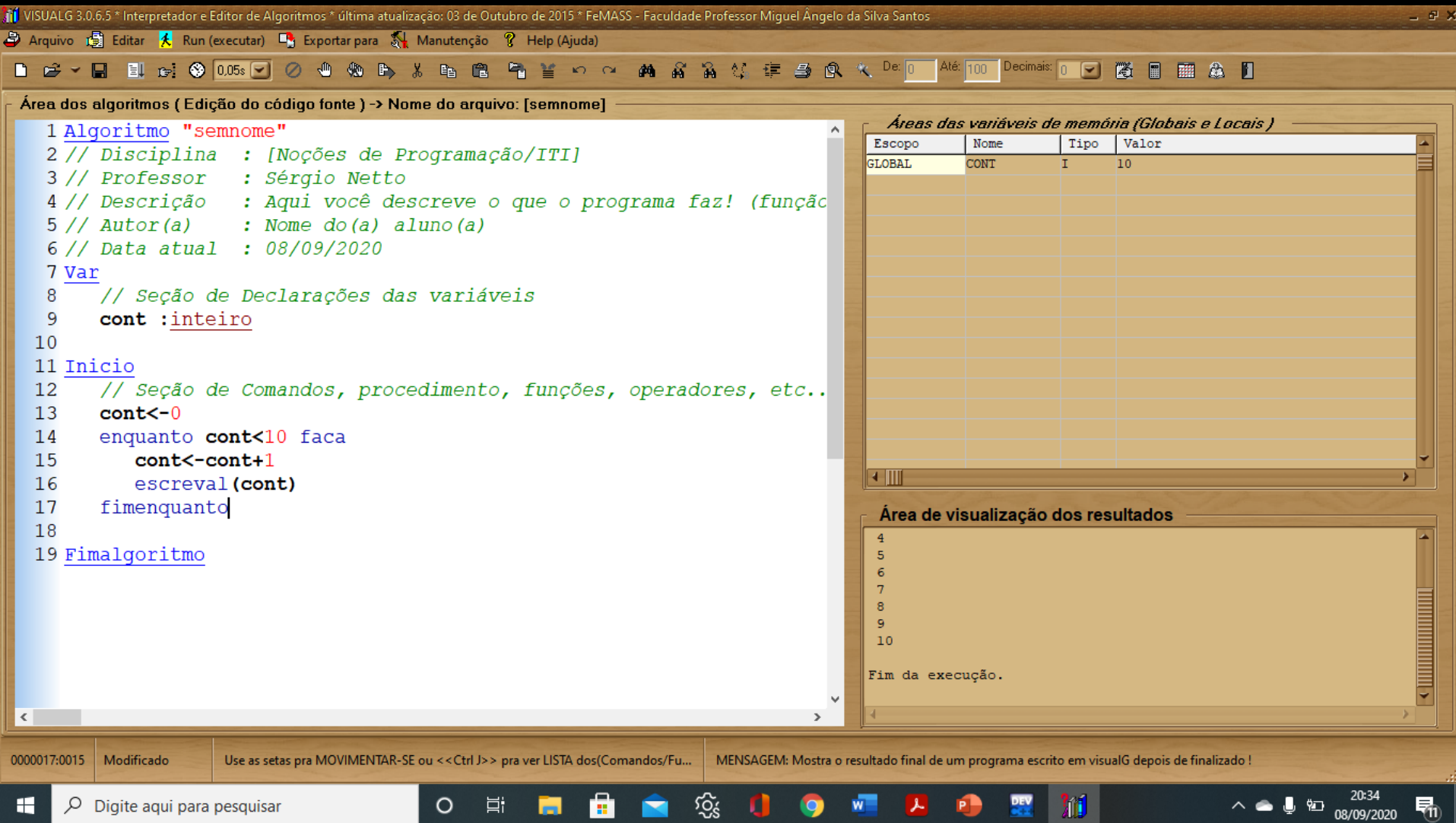
# Noções de Programação

## Estruturas de Repetição



# Noções de Programação

## Estrutura de Repetição



# Estrutura de Repetição

- Exercícios:

- 1) Fazer um programa que apresente uma sequência de números entre 20 e 50 inclusive.
- 2) Fazer um Algoritmo que apresente todos os múltiplos de 4 entre 1 e 100 inclusive.
- 3) Fazer um Algoritmo que apresente quantos números são múltiplos de 7 entre 100 e 500.
- 4) Fazer um Algoritmo que receba o intervalo (a, b) e um número c, apresente quantos são os múltiplos de c no intervalo.
- 5) Fazer um Algoritmo que apresente os N primeiros números da série de Fibonacci (0-1-1-2-3-5...)

# Operadores Aritméticos

## Operadores Aritméticos

$+, -$	Operadores unários, isto é, são aplicados a um único operando. São os operadores aritméticos de maior precedência. Exemplos: $-3$ , $+x$ . Enquanto o operador unário $-$ inverte o sinal do seu operando, o operador $+$ não altera o valor em nada o seu valor.
$\backslash$	Operador de divisão inteira. Por exemplo, $5 \backslash 2 = 2$ . Tem a mesma precedência do operador de divisão tradicional.
$+, -$ $, *, /$	Operadores aritméticos tradicionais de adição, subtração, multiplicação e divisão. Por convenção, $*$ e $/$ têm precedência sobre $+$ e $-$ . Para modificar a ordem de avaliação das operações, é necessário usar parênteses como em qualquer expressão aritmética.
MOD ou $\%$	Operador de módulo (isto é, resto da divisão inteira). Por exemplo, $8 \text{ MOD } 3 = 2$ . Tem a mesma precedência do operador de divisão tradicional.
$^$	Operador de potenciação. Por exemplo, $5 ^ 2 = 25$ . Tem a maior precedência entre os operadores aritméticos binários (aqueles que têm dois operandos).

# Operadores Relacionais e Lógicos

## Operadores Relacionais

<code>=, &lt;, &gt;, &lt;=, &gt;=, &lt;&gt;</code>	Respectivamente: igual, menor que, maior que, menor ou igual a, maior ou igual a, diferente de. São utilizados em expressões lógicas para se testar a relação entre dois valores do mesmo tipo. Exemplos: <code>3 = 3</code> (3 é igual a 3?) resulta em VERDADEIRO ; <code>"A" &gt; "B"</code> ("A" está depois de "B" na ordem alfabética?) resulta em FALSO.
--	---

**Importante:** No VisuAlg, as comparações entre *strings* não diferenciam as letras maiúsculas das minúsculas. Assim, `"ABC"` é igual a `"abc"`. Valores lógicos obedecem à seguinte ordem: FALSO < VERDADEIRO.

## Operadores Lógicos

<code>nao</code>	Operador unário de negação. <code>nao VERDADEIRO = FALSO</code> , e <code>nao FALSO = VERDADEIRO</code> . Tem a maior precedência entre os operadores lógicos. Equivale ao NOT do Pascal.
<code>ou</code>	Operador que resulta VERDADEIRO quando um dos seus operandos lógicos for verdadeiro. Equivale ao OR do Pascal.
<code>e</code>	Operador que resulta VERDADEIRO somente se seus dois operandos lógicos forem verdadeiros. Equivale ao AND do Pascal.
<code>xou</code>	Operador que resulta VERDADEIRO se seus dois operandos lógicos forem diferentes, e FALSO se forem iguais. Equivale ao XOR do Pascal.



# Função Randi()

Agora que estamos utilizando estruturas de repetição, iremos aumentar a quantidade e variedade de dados ou valores que precisaremos gerar para serem analisados por nossos programas.

Para isso utilizaremos a função randi() que gera valores aleatórios, evitando assim a necessidade de digitarmos muitos dados .

Segue o código com exemplos.



# Função Randi()

VISUALG 3.0.6.5 \* Interpretador e Editor de Algoritmos \* última atualização: 03 de Outubro de 2015 \* FeMASS - Faculdade Professor Miguel Ângelo da Silva Santos

Arquivo Editar Run (executar) Exportar para Manutenção Help (Ajuda)

0,05s Dec: 0 Até: 100 Decimais: 0

Área dos algoritmos ( Edição do código fonte ) -> Nome do arquivo: [semnome]

```
5 // Autor(a) : Nome do(a) aluno(a)
6 // Data atual : 15/03/2021
7 Var
8 // Seção de Declarações das variáveis
9 a, b ,c: inteiro
10 f :real
11
12 Inicio
13 // Seção de Comandos, procedimento, funções, operadores, etc..
14 a<-0
15 enquanto a<100| faça
16 // randi(10) gera valores aleatorios entre 0 e 9 inclusive
17 b<- randi(10)
18 // gerando valores decimais
19 f<- randi(100)/10
20 // gerando valores descolados de 10 exemplo valores entre
21 //15 e 30, primeiro temos que saber quantos são os números
22 //possíveis no caso (16)+15 que é o primeiro número.
23
24 c<- randi(16)+15
25 escreval(b, " ",f:4:1, " ",c)
26 a<-a+1
27 fimenquanto
28
29 Fimalgoritmo
```

Áreas das variáveis de memória (Globais e Locais)

Escopo	Nome	Tipo	Valor
GLOBAL	A	I	100
GLOBAL	B	I	5
GLOBAL	C	I	23
GLOBAL	F	R	9,300000000000000

Área de visualização dos resultados

```
7 8.8 17
3 8.7 21
1 7.1 29
0 6.9 18
2 8.3 27
0 2.8 16
5 9.3 23
```

Fim da execução.

0000015:0018

Modificado

Use as setas pra MOVIMENTAR-SE ou << Ctrl J >> pra ver LISTA dos(Comandos/Fu...

MENSAGEM: Permite CRIAR, ALTERAR, EXCLUIR, CONSULTAR e VISUALIZAR o código fonte escrito em VISUALG

# Estrutura de Repetição

- Exercícios Continuação:

- 6) Faça um Algoritmo que gere aleatoriamente dez números entre 20 e 50 e apresente a soma destes números.
- 7) Gere aleatoriamente 30 números entre -10 e 30, em seguida apresente a sua média.
- 8) Faça um Algoritmo que gere aleatoriamente 5 números entre 0 e 10 e seus respectivos pesos (1, 2 ou 3) ao final calcule sua média ponderada.
- 9) Escreva um Algoritmo que gere 10 números inteiros entre 0 e 100 e escreva o menor e o maior valor .
- 10) Faça um Algoritmo que gere um número inteiro positivo par N menor que 100 e imprima todos os números pares de 0 até N em ordem crescente.