**Curso de Lógica de Programação**

Curso de Algoritmos - Gustavo Guanabara

Sumário

[Aula 1 – Introdução a Algoritmos 1](#_Toc111718232)

[Aula 2 – Primeiro Algoritmo 2](#_Toc111718233)

[Algoritmos Computacionais 2](#_Toc111718234)

[ Variáveis 3](#_Toc111718235)

[Aula 3 - Comando de Entrada e Operadores 5](#_Toc111718236)

[Operadores Aritméticos 5](#_Toc111718237)

[Funções Aritméticas 6](#_Toc111718238)

[Aula 4 - Operadores Lógicos e Relacionais 7](#_Toc111718239)

[Operadores Relacionais 7](#_Toc111718240)

[Operadores Lógicos 7](#_Toc111718241)

[Ordem de Procedência Geral 7](#_Toc111718242)

[Aula 5 – Introdução ao Scratch 8](#_Toc111718243)

[Aula 6 – Exercícios de Algoritmo Resolvidos 8](#_Toc111718244)

[Aula 7 - Estruturas Condicionais 1 8](#_Toc111718245)

[- Função Se (expressão) Então ... Senão: 8](#_Toc111718246)

[Aula 8 - Estruturas Condicionais 2 8](#_Toc111718247)

[- Função Escolha (Var) ... Caso (valor) ... 8](#_Toc111718248)

[Aula 9 – Estruturas de Repetição 1 9](#_Toc111718249)

[- Função Enquanto (var) ... Faça 9](#_Toc111718250)

[Aula 10 – Estruturas de Repetição 2 9](#_Toc111718251)

[Aula 11 – Estruturas de Repetição 3 9](#_Toc111718252)

[Função Para (expressão) até (passo) faça ...: 9](#_Toc111718253)

[Aula 12 – Procedimentos 10](#_Toc111718254)

[-Escopo: 10](#_Toc111718255)

[ Passagem por Valor 10](#_Toc111718256)

[ Passagem por Referência 10](#_Toc111718257)

[Aula 13 – Funções 10](#_Toc111718258)

# Aula 1 – Introdução a Algoritmos

- Definição:

Manzano: *“****Algoritmos*** *são conjuntos de passos* ***finitos*** *e* ***organizados*** *que, quando executados, resolvem um determinado* ***problema****.”.*

- Conceito de Proporção Áurea: “*phi”*= 1,618....

Uma imagem contendo Diagrama

Descrição gerada automaticamente

# Aula 2 – Primeiro Algoritmo

Algoritmos Computacionais: São passos a serem seguidos por um **módulo processador** e seus respectivos usuários que, quando executados na ordem correta, conseguem realizar determinada tarefa**.**

**- “**Modulo processador”**:** Tudo aquilo que é programável

‘-> Ex: Sistemas de Gestão; Home Banking; Vídeos

**Para estruturar a lógica de programação temos alguns pseudocódigos/ferramentas, como:**

‘-> Fluxograma: representa a lógica através de um fluxo de informações seguido de um ponto ao outro.

‘-> Nassi Shneiderman

‘-> Portugol

Diagrama

Descrição gerada automaticamente

- O curso ensinará o pseudocódigos o portugol.

- NUNCA CONFUNDA!!!!: Fluxograma, Nassi e Portugol são ferramentas de representação de códigos, não os códigos.

- Usar o VISUALG

* **Comandos de Saída:**

- **Escreva (“**Olá, Mundo**”)**

- **Escreval (“**Me livrei da maldição**”)**

‘-> O ‘l’ em ‘escreval’ quer dizer linha, e adiciona uma linha abaixo do texto.

* Variáveis:

**VAR = Identificador : tipo**

- Identificadores**:**

Regras:

1. Deve começar com uma letra;
2. Os próximos podem ser letras ou números;
3. Não pode utilizar nenhum, símbolo exceto ‘**\_**’, isto é, o underline.
4. Não pode conter espaços em branco;
5. Não pode conter acentos;
6. Não pode ser uma palavra reservada.

* **Palavra Reservada**: No visualg, são palavras sublinhadas e/ou em tons de azul.

Exemplos de nomes identificadores corretos: Nota1; Inicio\_algoritmo;

Exemplos de nomes errados: Média; Salário Bruto; 9dade; Algoritmo; Inicio.

- Tipos Primitivos**:**

1. Inteiro

‘-> Ex: 1; 3; -5; 198; 0

1. Real:

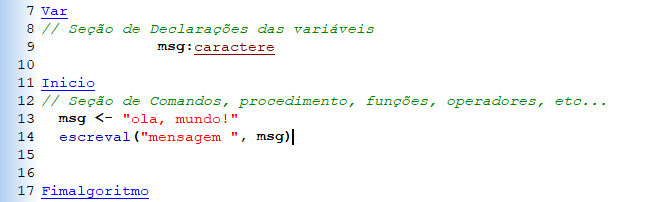
‘-> Ex: 0.5; 5.0[[1]](#footnote-1)[[2]](#footnote-2); -77.3; 3.1415

1. Caractere[[3]](#footnote-3):

‘-> “Gustavo”; “Algoritmo”; “123”

1. Logico:

‘-> Verdadeiro ou falso

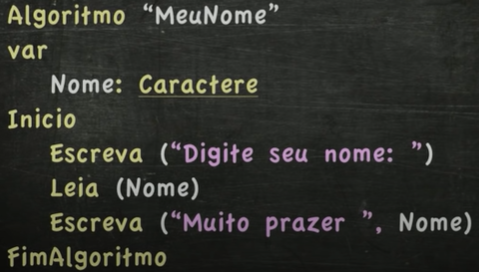


O símbolo ‘<-‘ é o de atribuição.

# Aula 3 - Comando de Entrada e Operadores

- Leia (var)

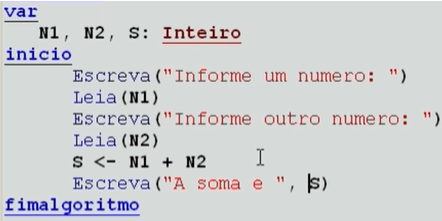
‘-> Comando de entrada que atribui uma mensagem enviada pelo usuário como identificador da variável.



Exercício:

1. Solicitar dois números para o usuário mostrar a soma entre eles.

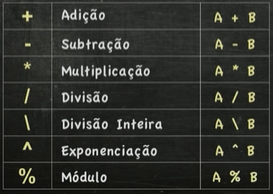
Código:



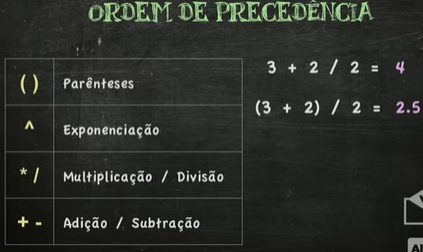
Outra opção é codificar o resultado da seguinte forma:

‘->Escreva ("A soma de", N1, " e", N2, ", é igual a: ", S)

Operadores Aritméticos**:**

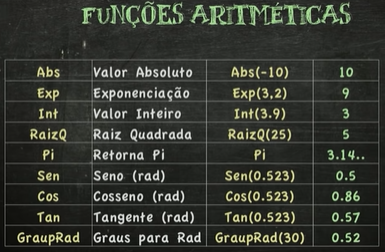


- Ordem de Precedência



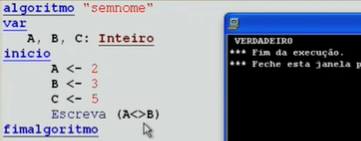
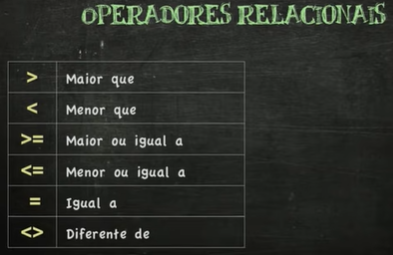
OBS: **A partir de uma operação de divisão**, automaticamente, a linguagem não trabalha mais com valores inteiros, mas **somente reais**.

Funções Aritméticas**:**

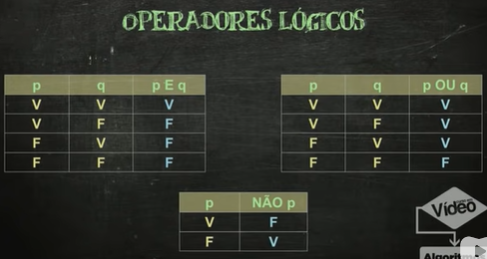


# Aula 4 - Operadores Lógicos e Relacionais

## Operadores Relacionais

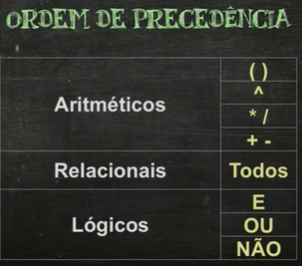
****

## Operadores Lógicos



- São os: e; ou; e não

## Ordem de Procedência Geral



# Aula 5 – Introdução ao Scratch

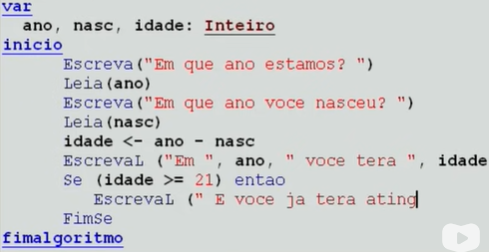
Eazzzy peazzzy.

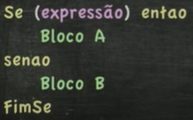
# Aula 6 – Exercícios de Algoritmo Resolvidos

# Aula 7 - Estruturas Condicionais 1

## - Função Se (expressão) Então ... Senão:

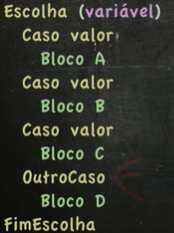
- Condicional Simples:



- Condicional Composta:

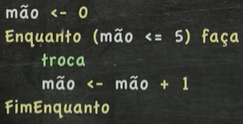
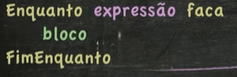
# Aula 8 - Estruturas Condicionais 2

## - Função Escolha (Var) ... Caso (valor) ...



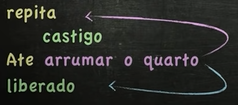
# Aula 9 – Estruturas de Repetição 1

## - Função Enquanto (var) ... Faça

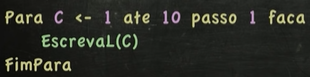
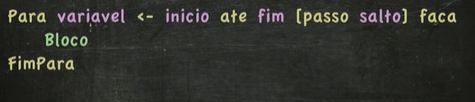
# Aula 10 – Estruturas de Repetição 2

**-** Função Repita ... Até (valor)**:**



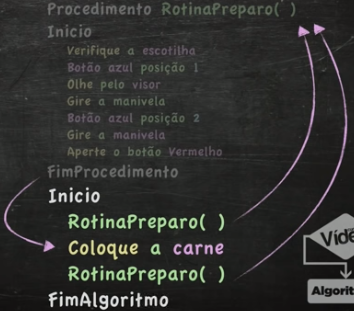
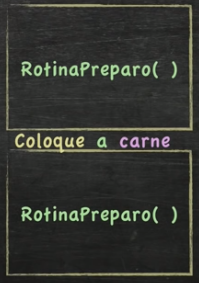
# Aula 11 – Estruturas de Repetição 3

## Função Para (expressão) até (passo) faça ...:

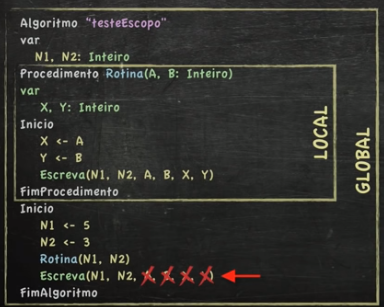


# Aula 12 – Procedimentos

Sempre que houver ações repetitivas, faça um procedimento e simplifique o código.



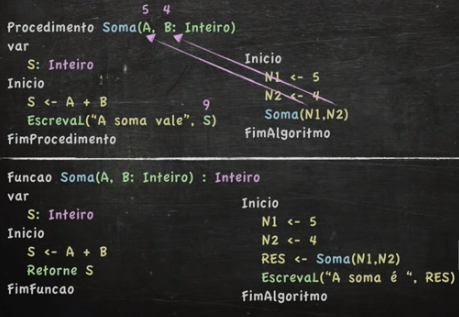
-Escopo: Local onde uma determinada variável irá funcionar.



## Passagem por Valor

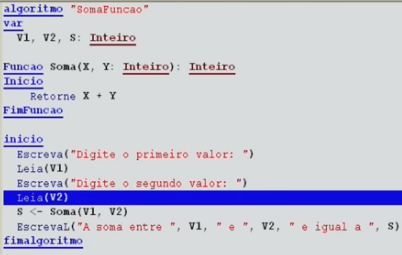
## Passagem por Referência

# Aula 13 – Funções



-Função Retorne:

Exemplos:



algoritmo "ParImpar"

var

N: Inteiro

R: Caractere

Funcao ParOuImpar(V: Inteiro): Caractere

inicio

Se (V%2 = 0) entao

Retorne "PAR"

senao

Retorne "IMPAR"

FimSe

FimFuncao

inicio

Escreva("Digite um numero: ")

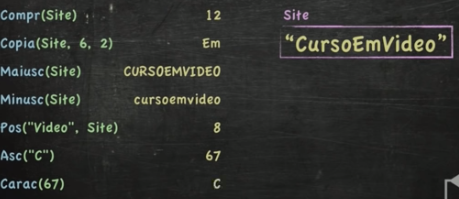
Leia(N)

R <- ParOuImpar(N)

Escreva ("O numero ", N, " e um valor ", R)

Fimalgoritmo

**- FUNÇÕES DO VISUALG:**

****

**Aula 14 –** Vetores

CADERNO

**Aula 15 –** Matrizes

- Variáveis Compostas Homogêneas Multidimensionais

Ex:

var

n: vetor [1..3, 1..2] de inteiro

1. O 5.0 e 5 é um número real, mas também é um número inteiro. Porém, importante salientar de que os reais estão contidos nos inteiros, então deve ser declarado como um inteiro primeiro. [↑](#footnote-ref-1)
2. Nas linguagens de programação ensinadas, as vírgulas são sempre representadas pelos pontos. [↑](#footnote-ref-2)
3. O tipo caractere admite acentos. [↑](#footnote-ref-3)