



# Lógica de Programação

Profª Fabrícia Noronha

# O que é Lógica?

*“Estudo das leis do raciocínio.”* (Dicionário Luft)

*“Conjunto de regras e princípios que orientam, implícita ou explicitamente, o desenvolvimento de uma argumentação ou de um raciocínio, a resolução de problemas, ...”* (Dicionário Aurélio)

# Lógica

Faz parte do dia a dia!

Sequência de passos lógicos é =

ALGORITMO

# Algoritmo

## **Caminho para a solução de um problema**

Em geral, os caminhos  
que levam a uma solução  
são muitos.





# Algoritmo Natural

Conjunto de passos ordenados utilizados no dia-a-dia para realizar alguma ação, como por exemplo: uma receita de bolo, um saque em um caixa eletrônico, trocar o pneu de um carro e etc.

# Exemplo:

## Receita de Bolo

**Entrada:** ingredientes(farinha de trigo, ovos, açúcar, ...)

**Processamento:** Modo de preparo

1. Untar a forma
2. Misturar os ingredientes
3. Mexer a massa
4. Colocar a massa na forma untada
5. Levar a forma ao forno
6. Esperar até o bolo assar
7. Tirar o bolo do forno

**Saída:** bolo pronto.



# Algoritmo Computacional

Conjunto de passos, sequência lógica de ações, organizados **sintaticamente** para a solução de um problema através dos computadores.

Entrada

Processamento

Saída

# Formas de Representação do Algoritmo

Fluxograma	Português Estruturado (Portugol)
<pre>graph TD; inicio([inicio]) --&gt; imprimir1[/Imprimir Informe dois valores/]; imprimir1 --&gt; ler[/Ler valor1, valor2/]; ler --&gt; soma[Soma= valor1+valor2]; soma --&gt; imprimir2[/Imprimir Soma/]; imprimir2 --&gt; fim([fim]);</pre>	<p><b>Algoritmo</b> soma_valores</p> <p><b>Variáveis</b> valor1, valor2, soma: inteiro</p> <p><b>inicio</b></p> <p>escrever('informe dois valores')</p> <p>ler(valor1, valor2)</p> <p>soma ← valor1+valor2</p> <p>escrever('Soma=', soma)</p> <p><b>fim</b></p>



# PASSOS

**Passo 1:** Solicitar ao usuário que informe um valor

**Passo 2:** Armazenar o valor informado

**Passo 3:** Solicitar ao usuário que informe outro valor

**Passo 4:** Armazenar o valor informado

**Passo 5:** Efetuar a soma dos valores informados e armazenar

**Passo 6:** Mostrar o resultado da soma

# 10 Mandamentos da Lógica de Programação

(Adaptado do material da prof Tanisi)

- 1) Assistir as aulas.
- 2) Copiar a resolução dos algoritmos.
- 3) Entregar a resolução dos algoritmos solicitados.
- 4) Desenvolver muitos, mas muitos algoritmos.
- 5) Não deixar para estudar um dia antes da prova.
- 6) Não desanimar se não conseguir resolver os primeiros algoritmos.
- 7) Prestar atenção na correção dos algoritmos. Entender a solução quando o professor explica já é meio caminho andado.
- 8) Fazer teste de mesa.
- 9) Codificar os algoritmos resolvidos em aula utilizando uma linguagem de programação.
- 10) Só se aprende algoritmos desenvolvendo algoritmos.

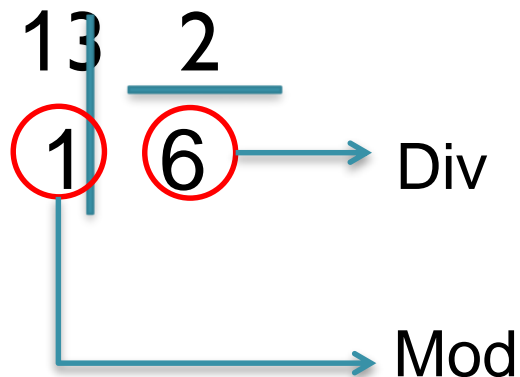
# Operadores Aritméticos

Operação	Operador
Subtração	-
Adição	+
Multiplicação	*
Divisão	/
Potenciação	$\wedge$ , * *

# Operadores Especiais

DIV – operador de divisão inteira

MOD – operador de resto da divisão inteira



$$13 \text{ Div } 2 = 6$$

$$13 \text{ Mod } 2 = 1$$

# Algumas Funções Pré-definidas

Função	Descrição
ABS(x)	Valor absoluto de x
SQR (x)	Eleva x ao quadrado
SQRT (x)	Raiz quadrada de x
FRAC (x)	Parte fracionária de x
INT (x)	Parte inteira de x
ROUND (x)	Valor de x arredondado
LOG (x)	Logaritmo de x

# Linearização de Expressões ou Fórmulas

Para a execução de cálculos matemáticos em algoritmos todas as expressões aritméticas devem ser linearizadas, ou seja, colocadas em linhas, devendo também ser feito o mapeamento dos operadores da aritmética tradicional para os do Português Estruturado.

Aritmética Tradicional	Forma Computacional
$x = \frac{43 \cdot [55 : (30 + 2)]}{2}$	$x \leftarrow (43/2 * (55/(30+2)))$
$area = \frac{b \cdot h}{2}$	$area \leftarrow (b * h) / 2$

# Estrutura do Algoritmo

❑ **Cabeçalho** (palavra “Algoritmo” seguida de um nome que dê uma idéia do objetivo do algoritmo)

EX: Algoritmo Calcula\_media

❑ **Área de Declaração** (variáveis e constantes que serão utilizadas)

❑ **Corpo do Algoritmo** ( delimitado pelas palavras ‘início’ e ‘fim’)

EX: início

    <comando...>

    fim

# Variáveis

## Características:

- ❑ **nome** (identificador - diferencia uma variável de outra)

Devem seguir as seguintes regras:

- ❑ Iniciar por letra(a...z, A...Z) ou underscore
- ❑ Depois podem ser seguidos por letras ou números
- ❑ Não pode ser palavra reservada nem caracter especial
- ❑ Não existe distinção para maiúsculas e minúsculas(NOME = nome)
- ❑ Nomes de variáveis devem ter no máximo 127 caracteres

Exemplo de identificadores não válidos: Calcula media, 5total, perc%

- ❑ **conteúdo** (dato guardado na variável)

- ❑ **tipo de dato** (tipo de informação que poderá ser armazenada na variável)



# Tipo de Dado

## ❑ Numérico

**Inteiro** (positivos ou negativos, sem ponto decimal) Ex: idade:inteiro

**Real** (positivos ou negativos, com ponto decimal) Ex: salário:real

## ❑ Caracter (sempre entre aspas simples)

- ❑ letras, números, símbolos especiais
- ❑ números quando declarados caracter, tornan-se representativos

Ex: cpf :caracter

nome: caracter

## ❑ Lógico ou Boolean

- ❑ verdadeiro ou Falso

Ex: fechado:lógico

# Declaração de Variáveis

- ❑ Começa com a palavra '**Variáveis**' e logo após todas as variáveis utilizadas no algoritmo devem ser declaradas
- ❑ Deve ter um nome significativo
- ❑ É necessário definir o tipo (inteiro, real, caracter, ...)

**Sintaxe:** Variaveis

identificador: tipo de dado

**Ex:** variaveis

valor2: inteiro

- ❑ Variáveis do mesmo tipo podem ser declaradas uma ao lado da outra, separadas por vírgulas.

**Ex:** media, raio, salario: real

# Declaração de Constantes

- Uma constante é um valor que não pode ser alterado ao longo da execução do algoritmo, ou seja, permanece com seus valores inalterados.

**Sintaxe:** constantes

identificadores = valor

**Exemplo:** constantes

pi = 3.14

juros = 0.3

# Comando de Atribuição

- ❑ Usado para atribuir valor a uma variável

**Sintaxe:**

**variavel ← valor ou expressão**

- ❑ O valor atribuído a uma variável deve ser sempre do mesmo tipo definido para a variável

**EX:** media ← (prova1 + prova2)/2

nome ← 'Maria'

a ← 25

# Comandos – Entrada e Saída de Dados

- **Escrever:** imprime informações na tela

**Ex:** escrever('Informe um valor')

escrever('Valor:', valor)

OBS: Usa-se “,” (vírgula) para concatenar o valor de uma variável com um texto explicativo.

- **Ler:** lê informações do teclado e armazena em uma variável.

**Ex:** ler(nome)

# Comentários

- São textos escritos dentro do algoritmo para explicar aspectos relativos ao mesmo.

**Sintaxe:** `(* ... *)`  
`{ ... }`

Estes comentários são de extrema importância especialmente em códigos mais complexos e devem acrescentar uma informação, não apenas frasear as instruções.

**Comentando:**

`total ← valorunitario * quantidade; (*calcula o total à pagar*)`

**Fraseando:**

`total ← valorunitario* quantidade; {atribui à total o valor unitário vezes a quantidade}`

# Teste de Mesa

- ❑ É a execução do algoritmo passo-a-passo como se ele fosse executado pelo computador mostrando a evolução do conteúdo das variáveis.

# Exemplo

Algoritmo que leia as duas notas de um aluno e calcule a média.

**Algoritmo** calcula\_media

**Variáveis**

nota1, nota2, media: real

**Início**

escrever('Informe nota1') {aparece na tela **Informe nota1**}

ler(nota1) {armazena a nota digitada na variável nota1}

escrever('Informe nota2') {aparece na tela **Informe nota2**}

ler(nota2) {armazena a nota digitada na variável nota2}

media  $\leftarrow$  (nota1 + nota2)/2 {calcula a média}

escrever('Média:', media) {aparece na tela **Média:** e o valor atribuído a  
variável média}

**fim**



# Teste de Mesa

nota1	nota2	media
8	6	7

# Identação

Recuo estabelecido nas estruturas de controle

**Estético???**



Fundamental para a legibilidade do programa.  
Contribui para o desenvolvimento e  
entendimento

# Exemplo

**Algoritmo** calcula\_media

**Variáveis**

nota1, nota2, media: real

**Inicio**

escrever('Informe nota1')

ler(nota1)

escrever('Informe nota2')

ler(nota2)

media  $\leftarrow$  (nota1 + nota2)/2

escrever('Média:', media)

**fim**