

# **IAL-003 – Algoritmos e Programação de Computadores**

---

Prof. Me. Anderson Vanin



# INTRODUÇÃO

Um algoritmo pode ser representado em diferentes formas:

- Linguagem natural;
- Fluxograma;
- Pseudocódigo;

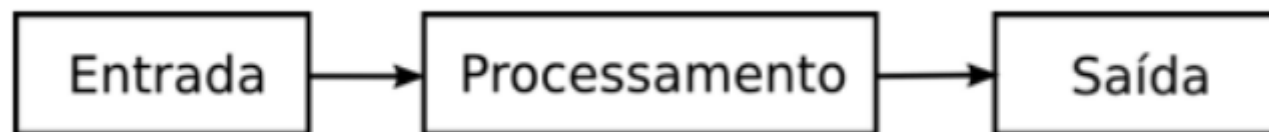
# ALGORITMOS

Um exemplo de algoritmos são receitas culinárias



# PARTES DE UM ALGORITMO

- Normalmente, um algoritmo é constituído por:
  - **Entrada de dados;**
  - **Processamento de dados;**
  - **Saída de dados.**
- Por exemplo: Um algoritmo que, a partir de um número fornecido, calcule o dobro e apresente o resultado.



# PARTES DE UM ALGORITMO

- O **processamento** é parte mais importante de um algoritmo
  - É ele quem define quais operações serão realizadas com os dados de entrada, e também como será apresentada a saída
- Em programas de computador, é comum...
  - ... os dados entrada serem valores digitados pelo usuário
  - ... e os dados de saída serem valores apresentados na tela (ou impressos na tela)

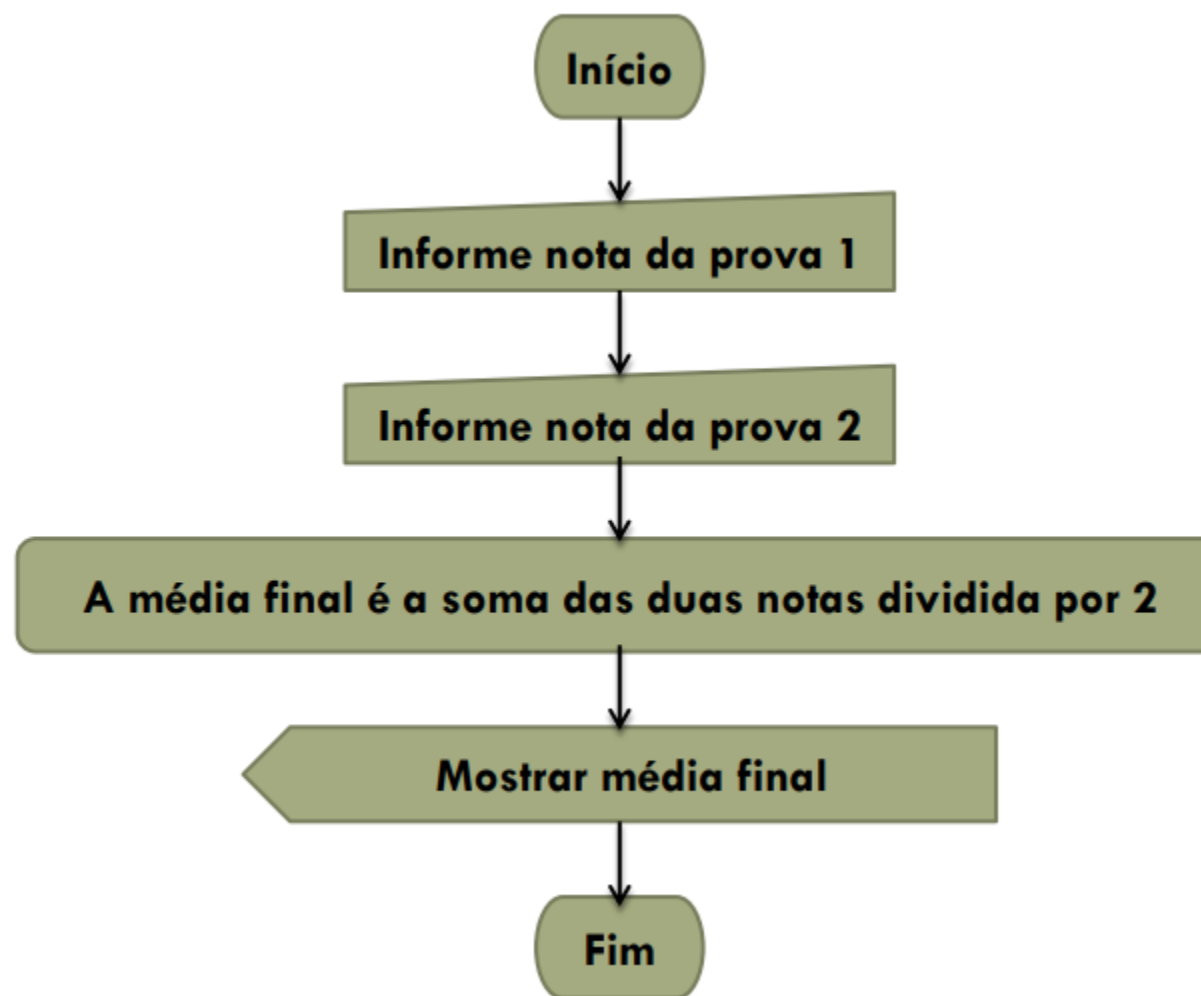
# FLUXOGRAMA

- Famosa frase:
  - **ENTENDEU?? OU QUER QUE EU DESENHE ??**
- Conceito: é um diagrama que representa o passo a passo de nosso algoritmo.
- Por se tratar de uma representação gráfica, possui um conjunto de símbolos padronizados.
- É representado pelo diagrama de blocos.

# DIAGRAMA DE BLOCOS






- É uma das formas de representação gráfica de algoritmos utilizando fluxograma.
- É uma forma padronizada e eficaz de representar os passos lógicos de um determinado processamento.
- Facilita o entendimento das ideias de uma pessoa ou equipe. Por esse motivo é muito popular.
- Os algoritmos são representados por um conjunto de figuras geométricas, com significado definido.

# DIAGRAMA DE BLOCOS





# SIGNIFICADO DAS FIGURAS

Símbolo	Função
	Indica o <b>início</b> e o <b>fim</b> de um processamento
	Indica <b>entrada de dados</b>
	Indica <b>processamento</b>
	Indica uma <b>decisão</b> a ser tomada (desvios)
	Indica <b>saída de dados</b>

# CONSTRUINDO O FLUXOGRAMA

- De forma semelhante a representação utilizando Linguagem Natural, também é necessário identificar os elementos do algoritmo na representação por fluxograma.
- 1. Quais os dados de entrada ?**
  - 2. Quais os processamentos ?**
  - 3. Quais as saídas ??**

# CONSTRUINDO O FLUXOGRAMA

Utilizando o exemplo da média de um aluno.

- 1. Quais os dados de entrada ?

**Nota 1 e Nota 2.**

- 2. Quais os processamentos ?

**Soma as duas notas e divide por 2.**

- 3. Quais as saídas ??

**A média das notas.**

# CONSTRUINDO O FLUXOGRAMA

Já conhecendo os elementos do algoritmo, a representação deste utilizando linguagem natural seria:

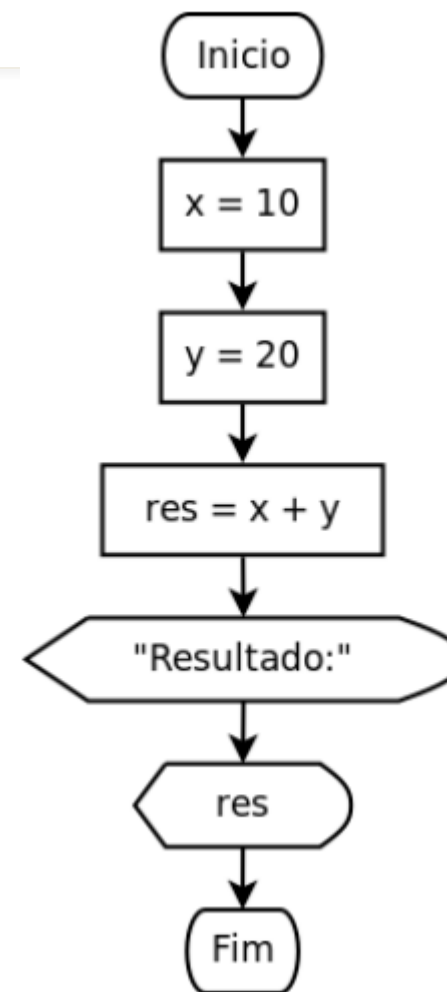
1. Informar Nota 1;
2. Informar Nota 2;
3. A média final é a soma da Nota 1 e Nota 2 dividido por 2;
4. Mostrar a média final.

# CONSTRUINDO O FLUXOGRAMA

- Conhecemos todos os passos do algoritmo.
- Podemos iniciar o processo de construção do fluxograma.
- É necessário traduzir o algoritmo da linguagem natural para forma de fluxograma.

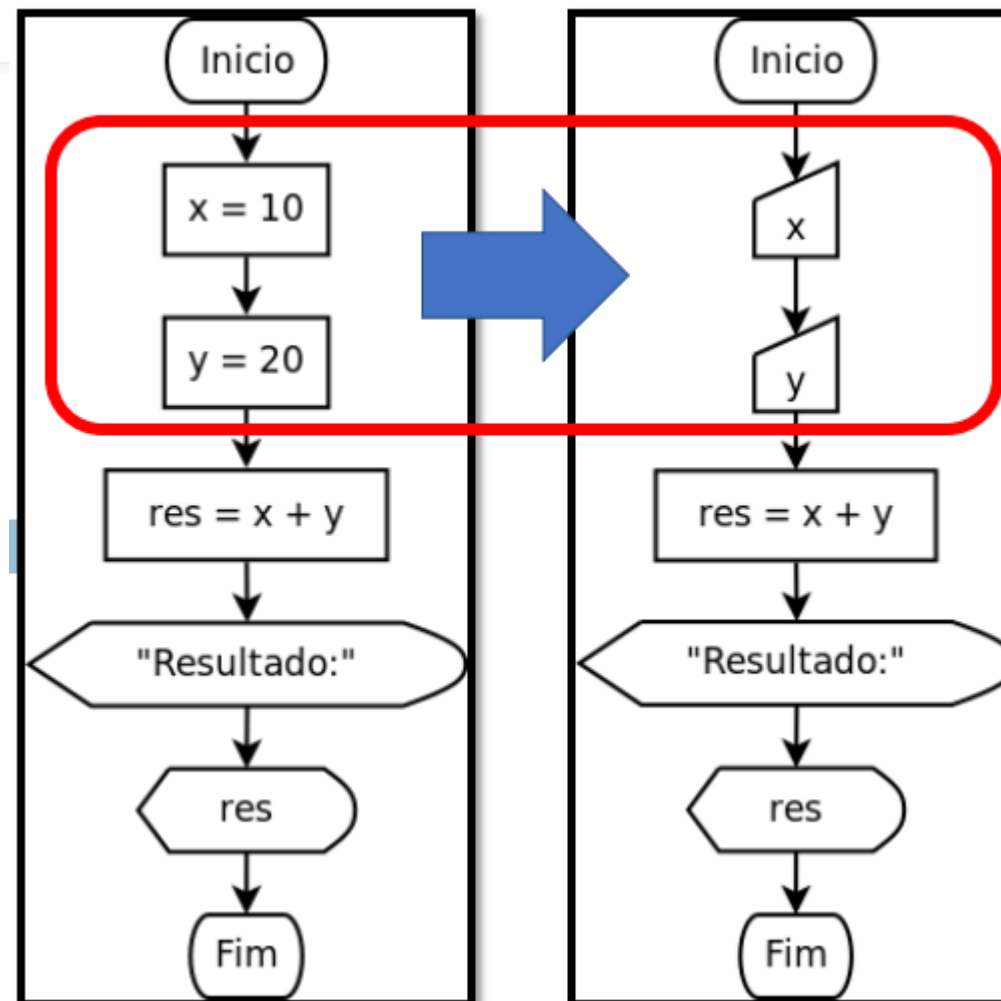
# OUTRO EXEMPLO - FLUXOGRAMA

- Todas estas instâncias são conectadas por linhas direcionadas (com setas)
- Estas linhas, juntamente com as marcações de início e fim, indicam por onde ocorrerá o fluxo
- Observe ao lado o fluxograma de um algoritmo simples de soma, que mostra o resultado ao final



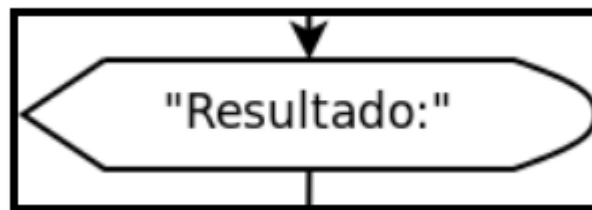
# OUTRO EXEMPLO - FLUXOGRAMA

- Neste fluxograma estamos atribuindo "estaticamente" os valores 10 e 20 a x e y
- Podemos utilizar a instância de leitura de dados para deixar o algoritmo um pouco mais dinâmico, sendo dependente da entrada do usuário

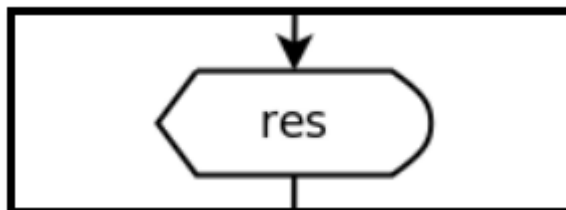


# OUTRO EXEMPLO - FLUXOGRAMA

- Repare também que a impressão de sequências de caracteres é dada por aspas duplas (")...



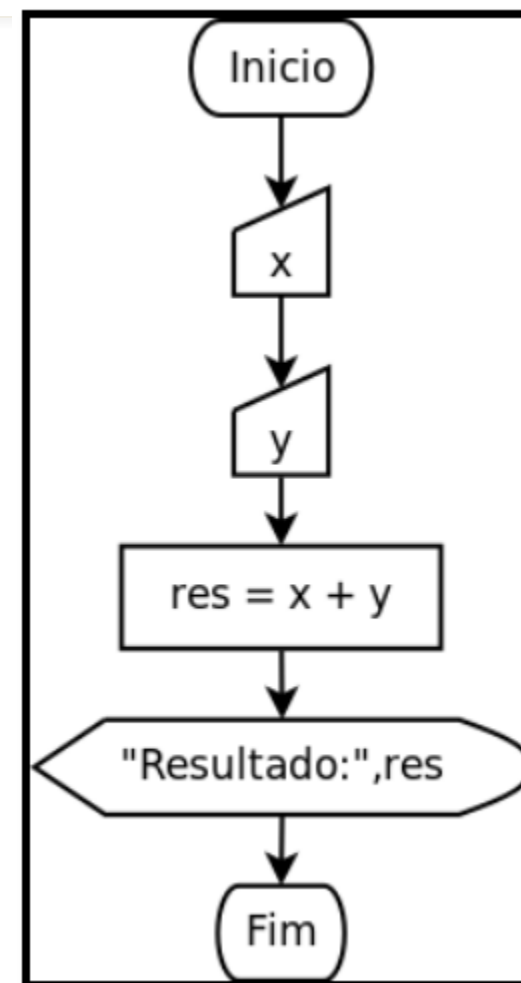
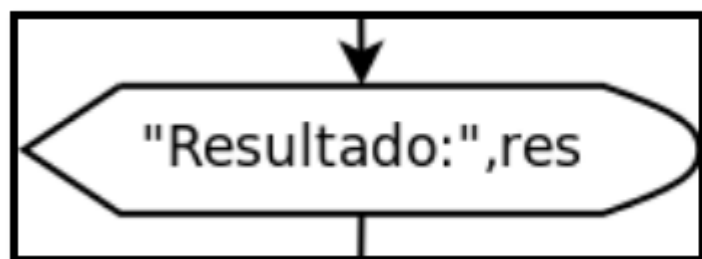
- ... e a impressão de variáveis, sem as aspas





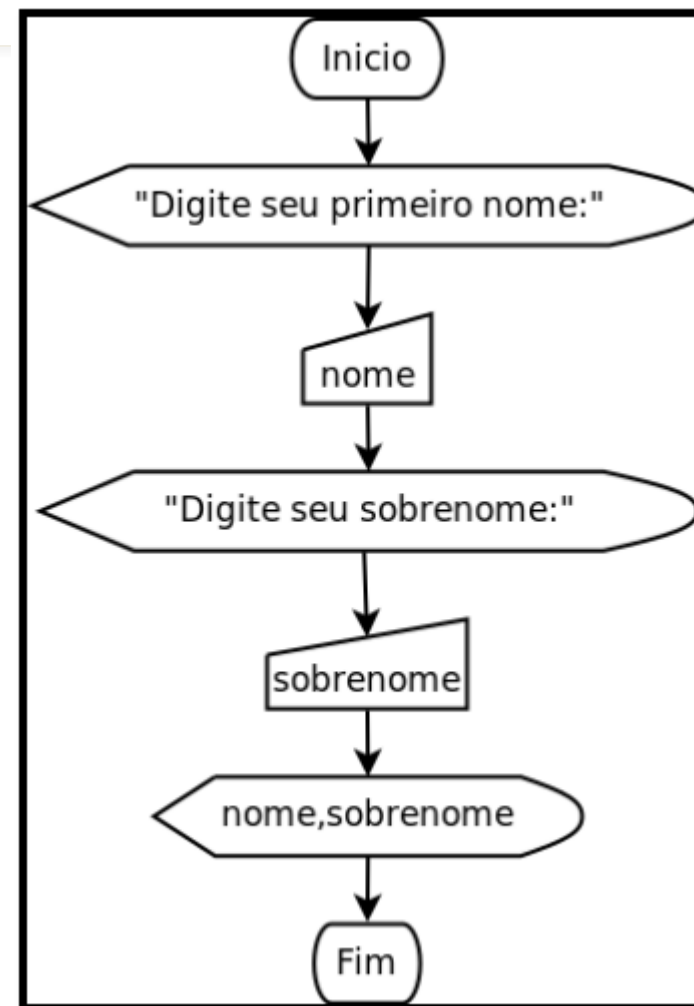
# OUTRO EXEMPLO - FLUXOGRAMA

- Para facilitar um pouco mais a notação de escrita, podemos separar as impressões por vírgula (,), da seguinte forma:



# OUTRO EXEMPLO - FLUXOGRAMA

- Programa para impressão de nome completo:



# EXERCÍCIO

1. Crie um fluxograma que calcule o dobro de um número digitado pelo usuário
2. Crie um fluxograma que receba 3 números digitados pelo usuário, e imprima na tela o produto deles
3. Crie um fluxograma que produza a saída:

Digite o seu peso (em kg): **91**  
Digite sua altura (em metros): **1.86**  
Seu imc é 26.3

- O texto **negrito significa a entrada** do usuário
- A fórmula do IMC é peso dividido pelo quadrado da altura
  - $\text{peso}/(\text{altura} \times \text{altura})$

# VARIÁVEIS E CONSTANTES

- As variáveis irão representar valores que serão utilizados durante o algoritmo.
- Uma variável precisa de um nome para ser acessada/modificada ao longo do tempo.

# VARIÁVEIS E CONSTANTES

Num programa de computador temos 2 estruturas básicas:

1. **Variáveis e Constantes:** espaços reservados na memória do computador para armazenar elementos de um certo conjunto ou tipo de dados.
  - **Variáveis:** durante a execução do programa, o conteúdo da variável pode mudar;
  - **Constantes:** o valor de uma constante não muda durante a execução do programa.
2. **Expressões:** durante a execução, combinam os valores armazenados nas variáveis e constantes para calcular novos valores.

# TIPOS DE DADOS

Tipos de dados armazenados nas variáveis e constantes

- **numérico:**

- inteiro:  $\dots, -4, -3, -2, -1, 0, 1, 2, 3, 4, \dots$

- real:  $\dots, -192.291, \dots, 192.291, \dots$

- **literal:**

- caractere: definido por um caractere “a”, “b”, etc.

- string: junção de caracteres, “bola”, “oi”, etc.

- **logico:** informação só pode ser verdadeiro ou falso.

# NOMES VÁLIDOS PARA VARIÁVEIS

Características dos nomes válidos para variáveis e constantes

- Podem ser usados números, letras minúsculas e maiúsculas e o caractere underscore (\_).
- Deve começar por uma letra, maiúscula ou minúscula, ou pelo caractere underscore (\_).
- Não podem ser usados símbolos como \$, #, !, ?, &, +, -.
- Não podem ser usados espaços em branco.
- Não podem ser usadas palavras reservadas da linguagem usada: algoritmo, leia, etc.

**Regra de ouro: Use nomes que façam algum sentido!**

# OPERADORES

Meios pelos quais se realizam operações sobre as variáveis e constantes, tais como atribuição de valores, incremento, decremento, multiplicação, comparação, etc.

- 1. Operadores de atribuição.**
- 2. Operadores aritméticos.**
- 3. Operadores relacionais.**
- 4. Operadores lógicos.**



# OPERADORES DE ATRIBUIÇÃO

Operadores de atribuição: usados para atribuir valores ou operações às variáveis ou constantes.

## Pseudocódigo

```
x ← 10
```

```
x ← x + 1
```

```
a ← "aula"
```

```
overlap ← falso
```

# OPERADORES ARITMÉTICOS

Operadores aritméticos: utilizados para operações com valores numéricos entre variáveis e constantes.

## Básicos

Operador	Símbolo	Exemplo
Soma	+	$a \leftarrow b + c$
Subtração	-	$a \leftarrow b - c$
Produto	*	$a \leftarrow b * c$
Divisão	/	$a \leftarrow b / c$

## Ordem de prioridade

Menor    +    -    \*    /    Maior

# OPERADORES ARITMÉTICOS – PRÉ DEFINIDOS

Operadores aritméticos: utilizados para operações com valores numéricos entre variáveis e constantes..

Operador	Símbolo	Exemplo
inteiro mais próximo do número real x	arredonda(x)	i ← arredonda(3.6)
parte inteira do número real x	parte_inteira(x)	i ← parte_inteira(0.8)
resto da divisão do número x pelo número y	resto(x,y)	r ← resto(8,3)
seno do ângulo x (expresso em radianos)	seno(x)	ang ← seno(3.1415)
cosseno do ângulo x (expresso em radianos)	cosseno(x)	ang ← cosseno(3.1415)
número x elevado ao número y (potência)	potencia(x,y)	p ← potencia(5,2)
raiz quadrada do número x	raiz_quadrada(x)	r2 ← raiz_quadrada(25)
raiz n do número x	raiz_enesima(x)	r3 ← raiz_enesima(3,27)

# OPERADORES RELACIONAIS

Operadores relacionais: usados na comparação entre valores ou expressões, retornando como resultado um valor lógico.

Operador	Símbolo	Exemplo
Igual a	=	$a = b$
Maior que	>	$a > b$
Menor que	<	$a < b$
Maior ou igual a	>=	$a >= b$
Menor ou igual a	<=	$a <= b$
Diferente de	<>	$a <> b$

# OPERADORES LÓGICOS

Operadores lógicos: relacionam entre si valores ou expressões lógicas, resultando em valores lógicos.

Operador	Símbolo	Especificação
Conjunção	e	Uma expressão E só é verdadeira se todos os valores das variáveis ou expressões forem verdadeiras.
Disjunção	ou	Uma expressão OU só é falsa se todos os valores das variáveis ou expressões forem falsos.
Negação	nao	O operador NÃO inverte o valor da expressão ou variável, se verdadeira inverte para falsa e vice-versa.