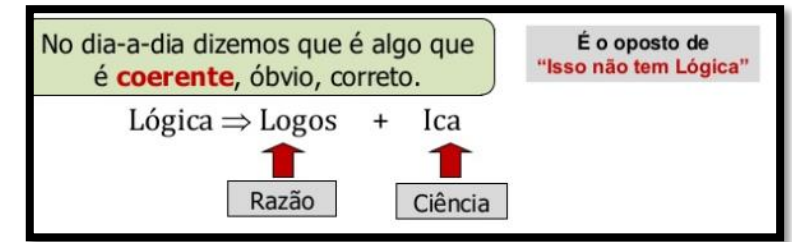




Lógica de Programação

Conceitos Iniciais: Lógica e Algoritmos

Lógica



→ Consiste em utilizar de princípios e conhecimentos a fim de buscar atingir um raciocínio correto.

→ É necessário para formular uma **sequência de raciocínio** que permita a você utilizar de premissas e informações para ser possível obter uma **conclusão**.

Raciocínio Lógico

- Evidências/Premissas/Informações + Análise = Conclusão
- Evidências/Premissas/Informações **erradas** + Análise = Conclusão **falha**
- Evidências/Premissas/Informações + Análise = Conclusão **errada** **falha**
- Evidências/Premissas/Informações + Análise = Conclusão **falha**



Exemplos dia a dia:

- Cálculo matemático?
- Como fazer uma receita?
- Um processo comum do dia a dia
- Um processo técnico, profissional?
- Como chegar em um lugar?
- Como jogar um jogo?

Todos os peixes vivem na água. (premissa1)
Golfinhos são peixes. (premissa 2)
 \therefore Golfinhos vivem na água. (conclusão)

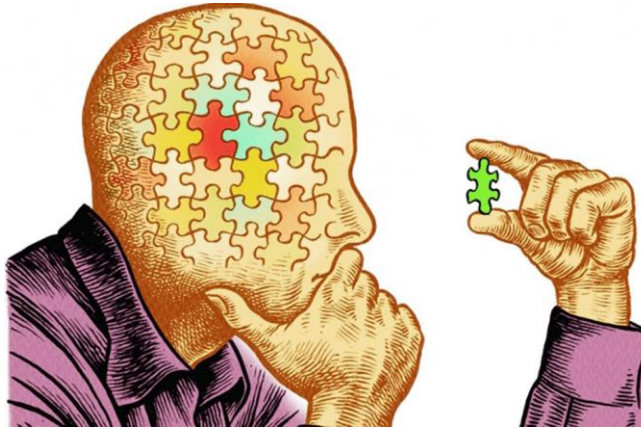


As coisas mais simples podem ser descritas por uma sequência lógica
Ex: "Chupar uma bala"

Lógica

Todo homem é mortal.
Sócrates é homem.
Logo, Sócrates é mortal.

Se $A = B$
Se $B = C$
Logo, $A = C$



$$2X + Y = 10$$

$$X + Y = 7$$

$X = 2$ e $Y = 3$ (V ou F?)

$X = 3$ e $Y = 4$ (V ou F?)

Qual é o valor da sequencia lógica?

1, 3, 5, 7, ?

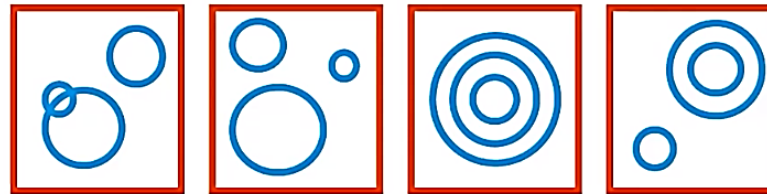
1, 1, 2, 3, 5, 8, ?

S, T, Q, ?, S, S, D

AC BD CE DF ?

SEQUÊNCIAS ENVOLVENDO FIGURAS

EXEMPLO 02



1

2

3

4

Animais, gatos, canetas

- a) 1
- b) 2
- c) 3
- d) 4

Seja Trainee : Testes de raciocínio lógico | Ebook e Exercícios resolvidos

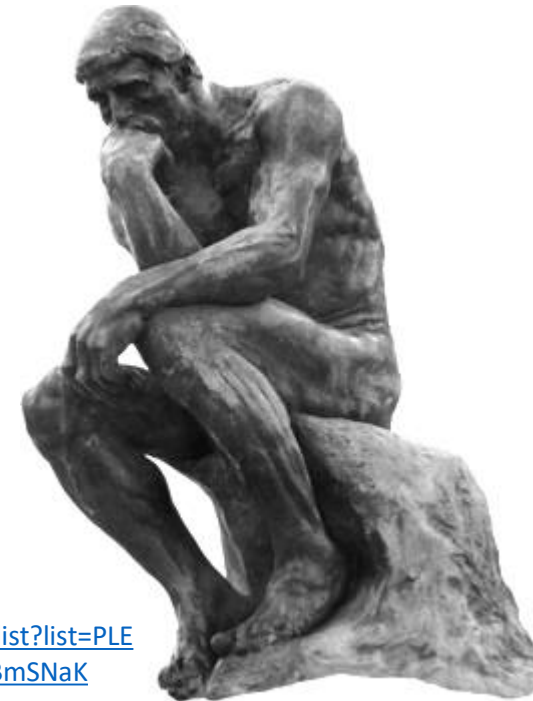
<https://sejatraineer.com.br/testes-de-raciocinio-logico-dos-processos-traineer/>

Noções de Lógica (playlist)

<https://www.youtube.com/playlist?list=PLEfwqyY2ox85ShbZ1O0fXmkUxavBmSNaK>

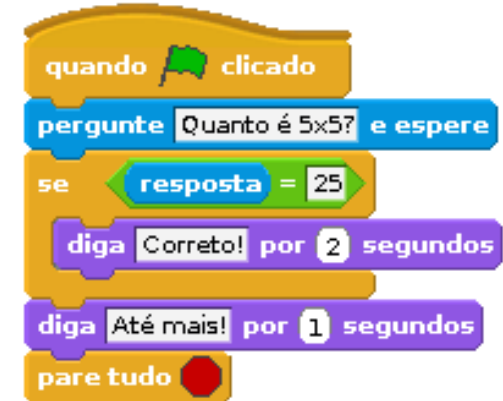
Link de um Drive com material sobre lógica para se preparar em testes e concursos

https://drive.google.com/drive/folders/1EuWOLylLZtkMvR08neVanF--4Ycp_4QB?usp=sharing



A Lógica de Programação e Algoritmos

- Uma técnica de encadear pensamento para atingir um determinado **objetivo**.
- sequência lógica = que é um conjunto de passos a serem executados → **ALGORITMO**



<https://scratch.mit.edu/about>

Um ALGORITMO funcional deve

1. Ter um número **finito** de passos;
2. Que os passos sejam **bem definidos**;
3. Que existam **zero ou mais entradas**, tomadas em conjunto e bem estruturadas;
4. Existam **uma ou mais saídas**;
5. Que exista uma **condição de fim** sempre bem definida assim que o algoritmo completar sua tarefa para qualquer entrada num tempo determinado.
- 6- As frases (instruções) de um algoritmos geralmente são compostas por **verbos** no **IMPERATIVO** ou **INFINITIVO**



As 3 partes de um algoritmo

CONSTRUÇÃO DE UM BOM PSEUDOCÓDIGO OU ALGORITMO

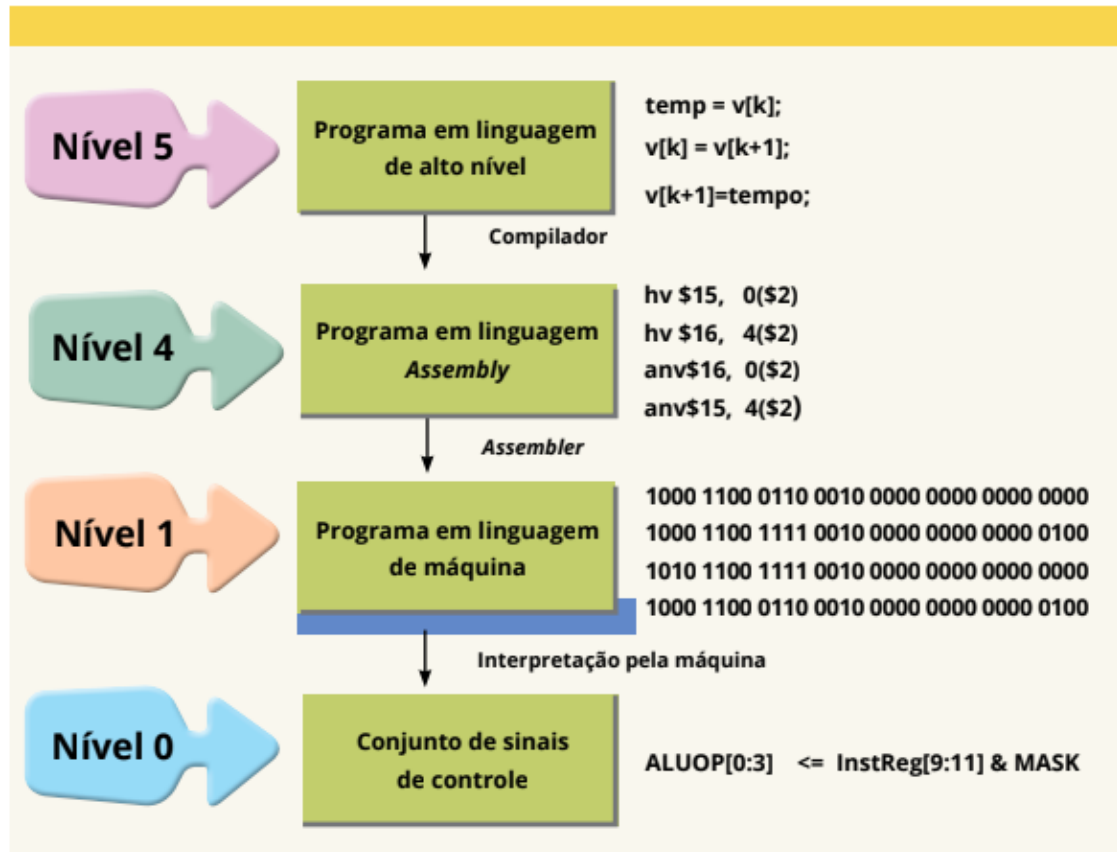
Para se escrever um bom pseudocódigo e tentar descrever uma sequência lógica de um programa, de forma que seja objetiva na finalidade de interpretar uma ação computacional por meio de uma linguagem, (no caso o português ou português estruturado) levaríamos em conta os seguintes detalhes abaixo:

1. Usar apenas verbos na fase de escrita;
2. Imaginar que a escrita do pseudocódigo está direcionada a pessoas que não conhecem de informática.
3. Usar frases curtas e diretas;
4. Ser claro e objetivo;
5. Tomar cuidado com palavras que possam ser encaradas com duplo sentido.



Linguagem de Programação

Figura - Codificação de programas e "níveis" na arquitetura de computadores



Nível 5 - Linguagem de Alto Nível: A = 5;

Nível 4 - Linguagem Assembly: MOVE A, #5;

Nível 0 - Linguagem de Máquina: 0011001100000101



LINGUAGEM INTERPRETADA

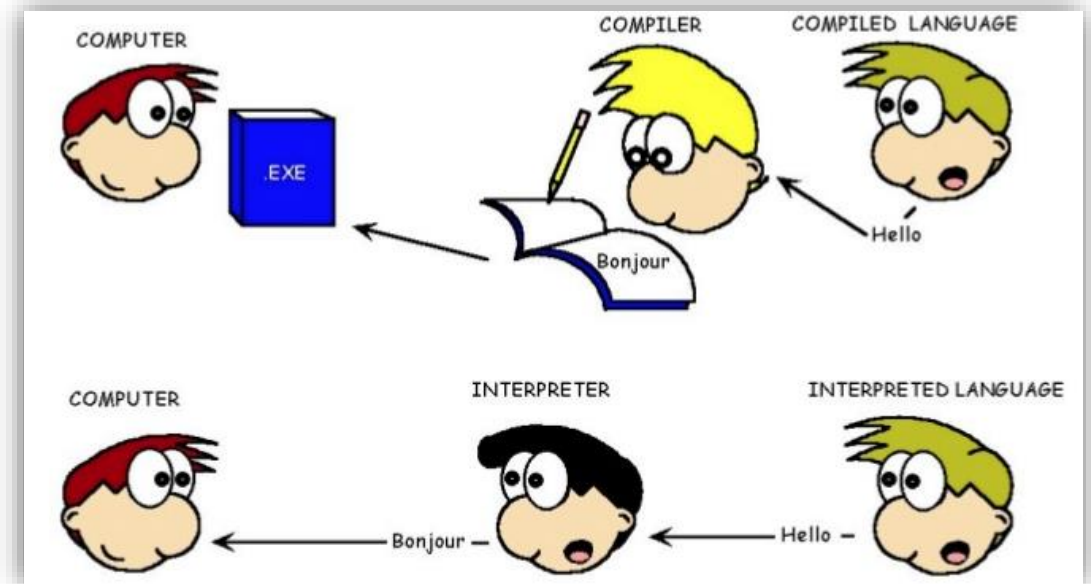
- Executam o código do programa à medida que este vai sendo traduzido → mais flexível
- Ex: JavaScript, e Python.

Java e C#

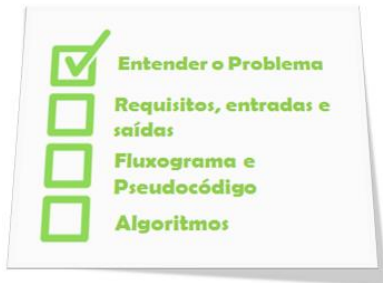


LINGUAGEM COMPILADA

- Traduzem todo o código do programa para apenas depois executar → **compilador** → tradução para a linguagem de máquina.
- Ex: C e C++



Linguagem de Programação



INICIO

Código fonte

```
#include <iostream.h>
#include <conio.h>

const int TAM = 7, V=1, F=0;
void bolha(int conjunto[TAM], int tamar
int houveTroca = V, posLimInf, i, t
for(posLimInf= tamanho-1; (houveTroca
houveTroca = F;
for(i = 0; (i < posLimInf); i++)
if (conjunto[i] > conjunto[i+1])
trab = conjunto[i];
conjunto[i] = conjunto[i+1];
conjunto[i+1] = trab;
houveTroca = V;
}
```



Compilador

traduz o código de uma linguagem de alto nível para o código em linguagem assembly



Código Assembly

```
.model small
.stack
.code
mov ah,2h
mov dl,2ah

int 21h
mov ah,4ch
int 21h
end
```



Montador

Traduz um código assembly para o código de máquina



0000	FF	D8	FF	E1	1D	FE	45	78	69	66	00	00	49	49	2A	00
0010	08	00	00	00	09	00	0F	01	82	00	06	00	00	00	7A	00
0020	00	00	10	01	82	00	14	00	00	00	80	00	00	00	12	01
0030	03	00	01	00	00	00	01	00	00	00	1A	01	05	00	01	00
0040	00	00	A0	00	00	00	18	01	05	00	01	00	00	00	A8	00
0050	00	00	28	01	83	00	01	00	00	00	02	00	00	00	32	01
0060	02	00	14	00	00	00	80	00	00	00	13	02	03	00	01	00
0070	00	00	01	00	00	00	69	87	04	00	01	00	00	00	C4	00
0080	00	00	3A	06	00	00	43	61	6E	6F	6E	00	43	61	6E	6F
0090	4E	20	50	6F	77	65	72	53	6A	6F	74	20	41	36	30	00
00A0	00	00	00	00	00	00	00	00	00	00	00	00	84	00	00	00
00B0	01	00	00	00	84	00	00	00	01	00	00	00	32	30	30	34
00C0	3A	30	36	3A	32	35	20	31	32	3A	33	30	3A	32	35	00
00D0	1F	00	5A	82	05	00	01	00	00	00	86	03	00	00	90	82
00E0	05	00	01	00	00	00	8E	03	00	00	00	90	07	00	04	00

Arquivo Objeto -
Linguagem de máquina
(arquivo binário)



Link Editor

Interliga os diversos módulos de um programa para gerar um pacote que será posteriormente carregado para a memória

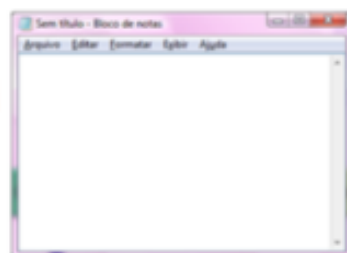


Pacote



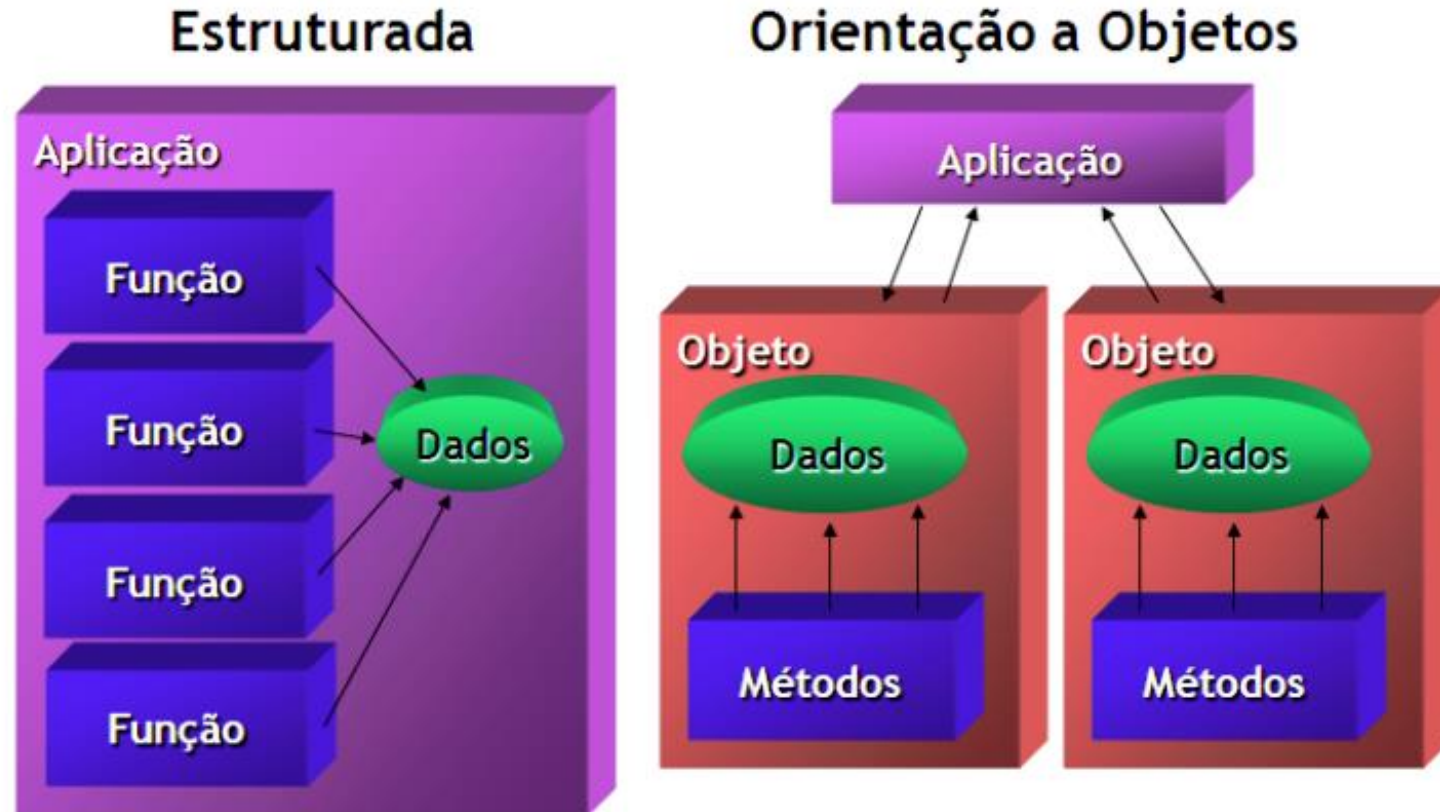
Carregador

transfere o conteúdo do pacote para a memória e encaminha o início da execução do programa



FIM

Linguagem de Programação



Qual linguagem de programação escolher?

<https://www.youtube.com/watch?v=DKBC5pEpCLY&list=PL99AE7FAAE4560E2A&index=4>

A Lógica de Programação e Algoritmos

Problema

Fazer um bolo de chocolate.

Saída

Qual o **resultado** esperado?

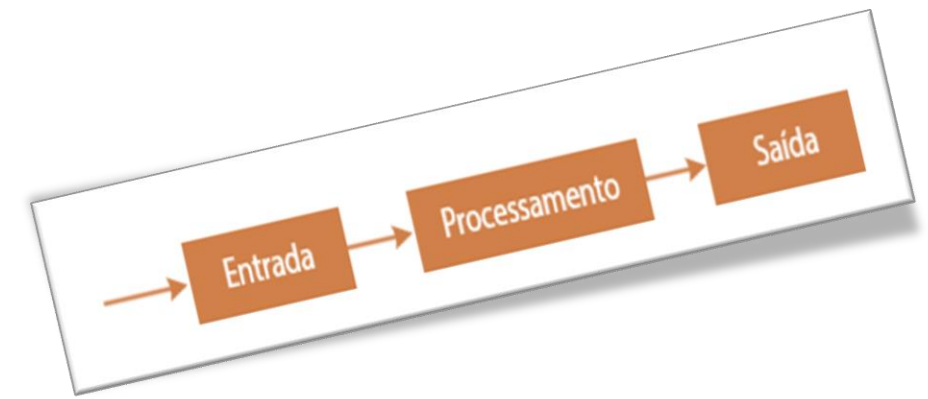
Passos

O que é preciso para se obter o resultado desejado?

☐ **Ingredientes**

Passos

Qual a sequência necessária (de uso dos **ingredientes**) para se obter o resultado desejado?



Dados de Entrada

Ingredientes:

- ☐ 5 ovos.
- ☐ 250 gramas de margarina cremosa.
- ☐ 2 xícaras (chá) de açúcar.
- ☐ 1 xícara (chá) de farinha de trigo.
- ☐ 1 xícara (chá) de chocolate.
- ☐ 200 gramas de côco ralado.
- ☐ 1 copo de leite.
- ☐ 1 colher (sopa) de fermento.

Processamento

Modo de Preparo

- ☐ Bata a margarina, as gemas e o açúcar até ficar cremoso.
- ☐ Junte o leite, o côco e a farinha e continue batendo.
- ☐ Acrescente o fermento e, por último, as claras em neve.
- ☐ Unte uma forma com manteiga e leve ao forno para assar.

A Lógica de Programação e Algoritmos

Escreva um algoritmo para ler dois números (a e b) e trocar os seus valores. Exibir os valores de a e b após a troca



- **SAÍDA:** Quais os resultados que o algoritmo deve fornecer ?
- **ENTRADA:** Quais os insumos necessários para se obter a saída?
- **PROCESSAMENTO:** Como transformar os insumos na saída?

- **SAÍDA:** valores de a e b (trocados)
- **ENTRADA:** dois números (A, B)
- **PROCESSAMENTO:**
 - $A = B$
 - $B = A$

```
5 inteiro a, b
6 escreva ("Digite o valor de a: ")
7 leia(a)
8 escreva ("Digite o valor de b: ")
9 leia(b)
10 a = b
11 b = a
12 escreva("\n a = ", a)
13 escreva("\n b = ", b)
```

Declaração de Variáveis

Entrada de Dados

Processamento

Saída de Dados

OBS: Esse programa não atende ao enunciado de forma correta, possui um **erro!**
(Trabalharemos na correção em "Teste de Mesa")

Representação de Algoritmos

ALGORITMO = Sequência lógica

Exemplo 1

DESCRIÇÃO NARRATIVA

diretamente em linguagem natural



Algoritmo Exemplo: FRITAR UM OVO

1. Pegue a frigideira, ovo, óleo e sal
2. Coloque o óleo na frigideira
3. Acenda o fogo
4. Coloque a frigideira no fogo
5. Espere o óleo esquentar
6. Quebre o ovo
7. Despeje o ovo no óleo quente
8. Coloque o sal
9. Retire quando estiver pronto
10. Desligue o fogo

Representação de Algoritmos

ALGORITMO = Sequência lógica

DESCRIÇÃO NARRATIVA diretamente em linguagem natural

Exemplo 2

Obter as quatro notas do aluno;

Somar as quatro notas obtidas;

Dividir o resultado da adição por quatro;

Se a média obtida for maior ou igual a 7 (sete):

O aluno foi aprovado.

Caso contrário:

O aluno foi reprovado.



$$\text{Média} = \frac{\text{nota1} + \text{nota2} + \text{nota3} + \text{nota4}}{4}$$

Exemplo 3 – Trocar uma lâmpada

- Verificar se a lâmpada está quebrada;
- Comprar outra lâmpada;
- Pegar a escada;
- Verificar se a energia está desligada;
- Subir na escada;
- Afrouxar a lâmpada e retirar do bocal;
- Descer da escada e pegar outra lâmpada;

- Subir na escada;
- Rosquear a outra lâmpada no bocal;
- Descer da escada;
- Ir ao interruptor e verificar se a lâmpada acende;
- Retirar a escada do lugar e guardar;
- Descartar a lâmpada ruim.



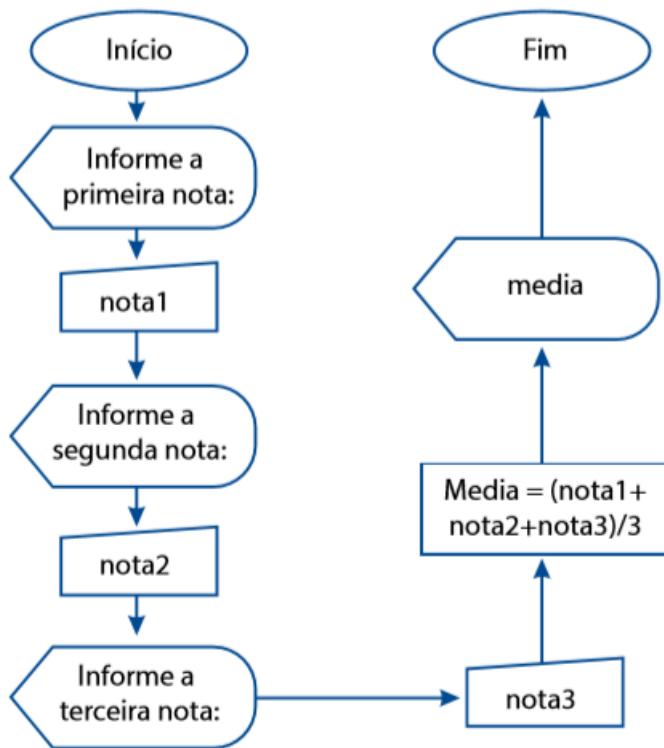
É pouco utilizada na prática, pois esse tipo de representação quando não é bem estruturada dá margem a **ambiguidades e imprecisões**.

Exemplo: “pegar a escada”, nesse caso precisaria descrever como vou pegar a escada, e ainda mais, descobrir primeiro onde ela está. Só assim seriam diminuídas as chances de se cometer algum tipo de equívoco.

Representação de Algoritmos

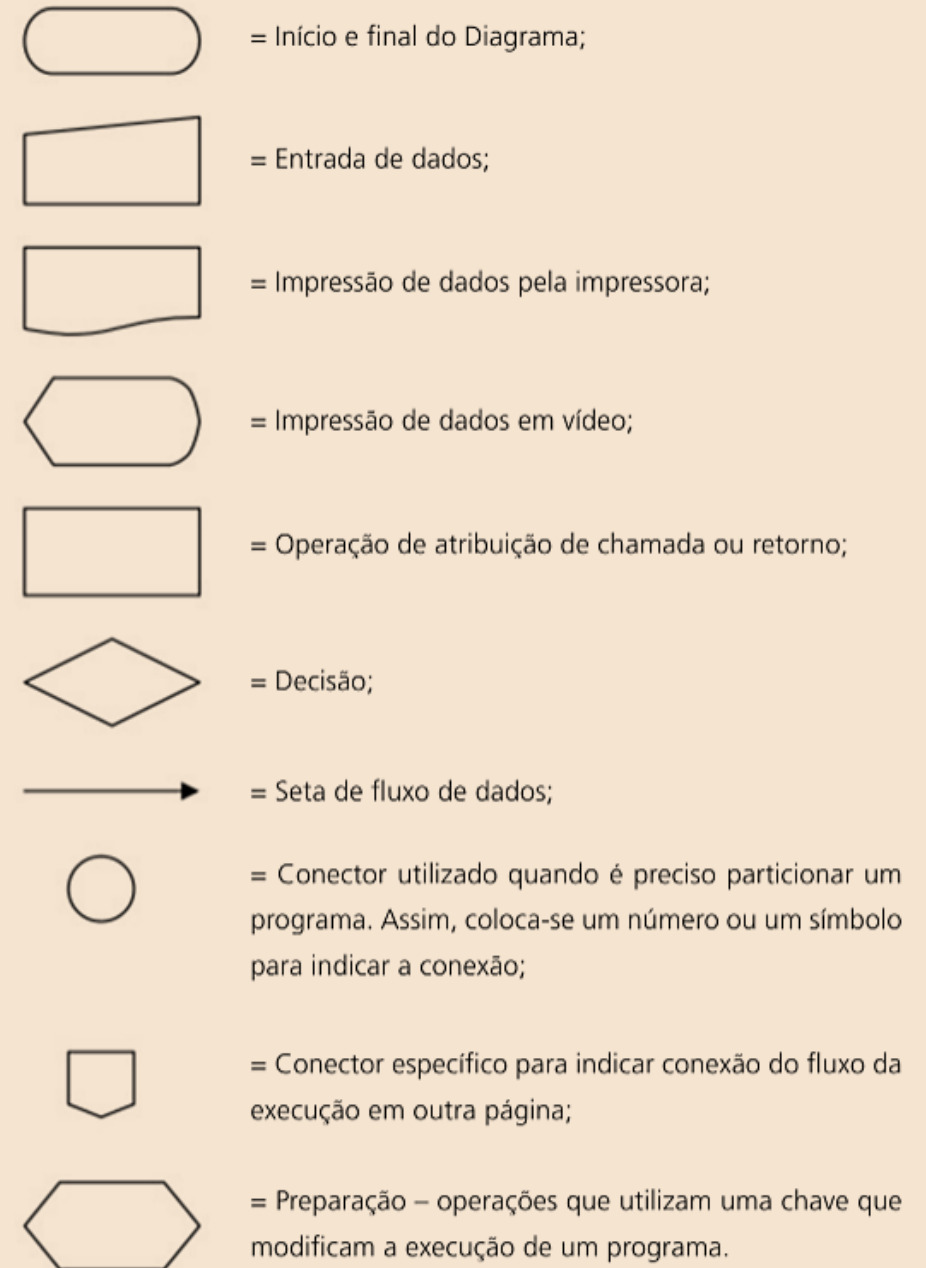
ALGORITMO = Sequência lógica

FLUXOGRAMA OU DIAGRAMA DE BLOCOS



Quais serão as **saída de dados**?
Quais são os **dados de entrada**?
Qual será o **processamento** a ser utilizado?

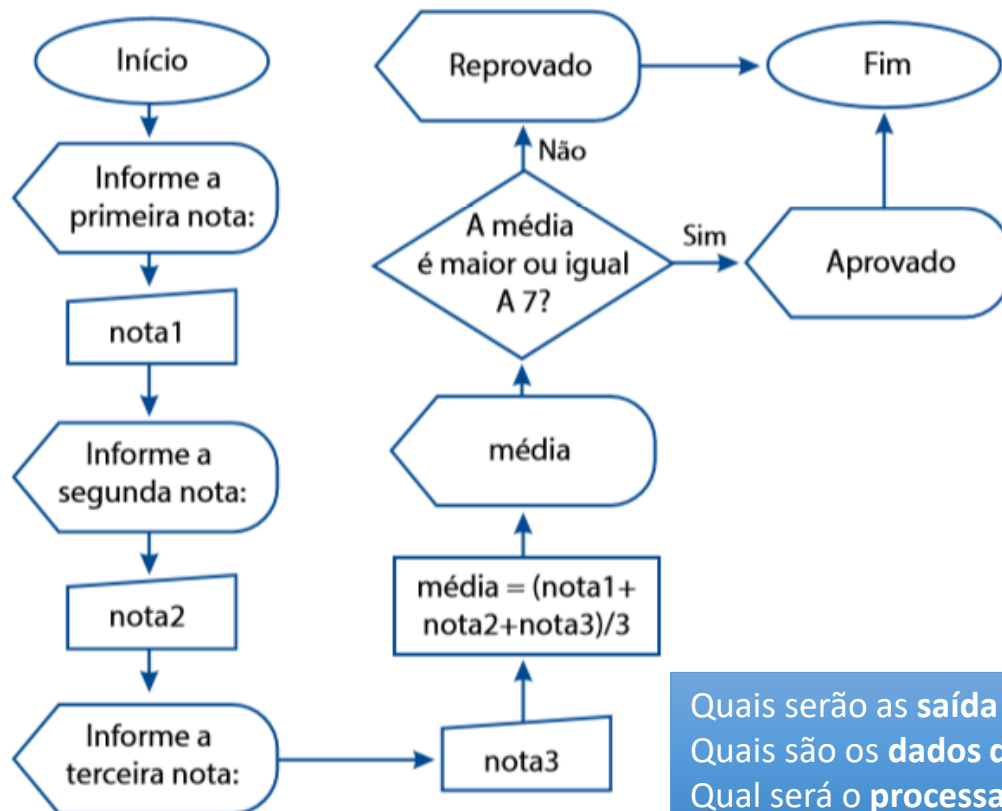
$$\text{Média} = \frac{\text{Nota1} + \text{Nota2} + \text{Nota3}}{3}$$



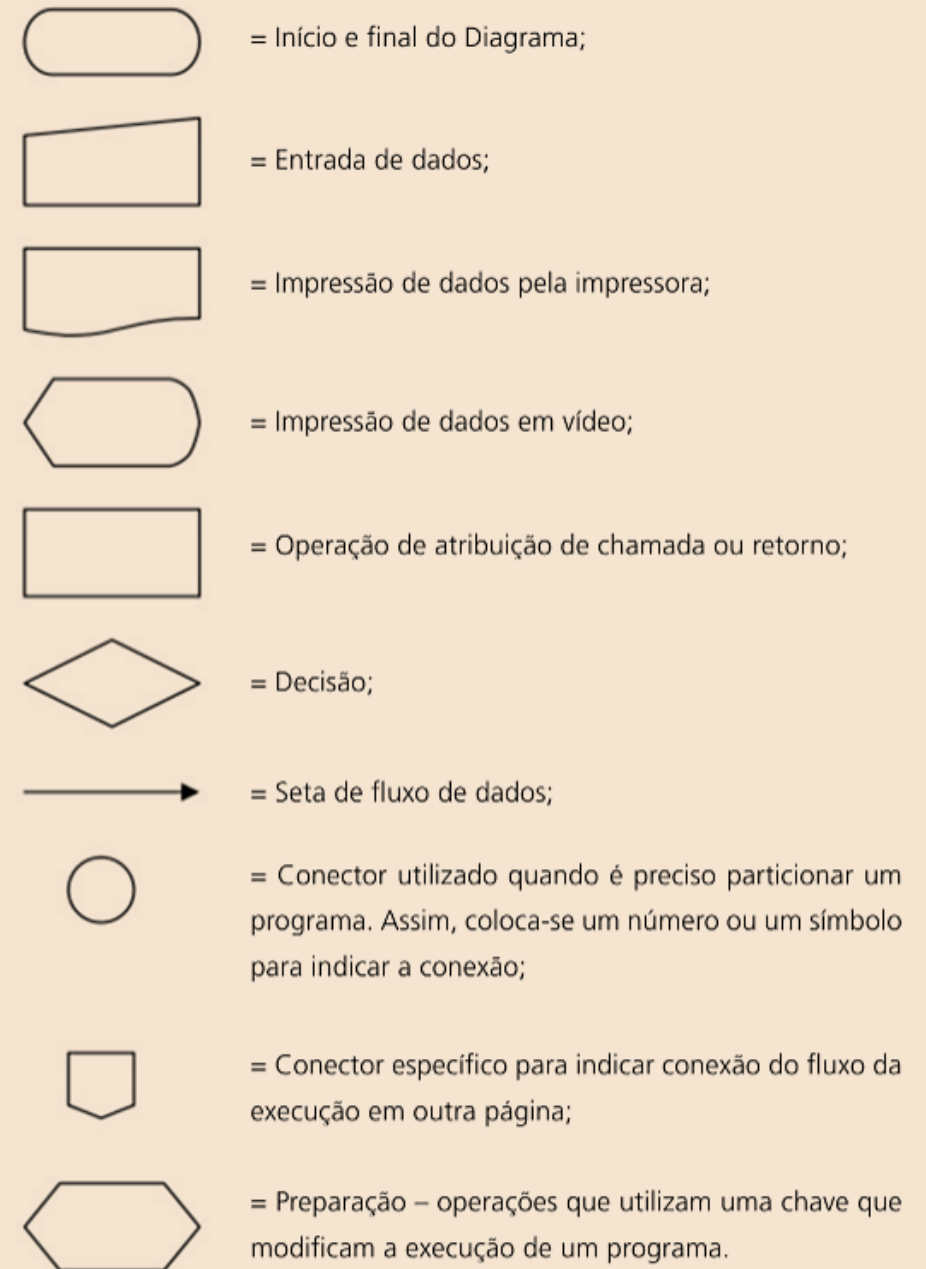
Representação de Algoritmos

ALGORITMO = Sequência lógica

FLUXOGRAMA OU DIAGRAMA DE BLOCOS



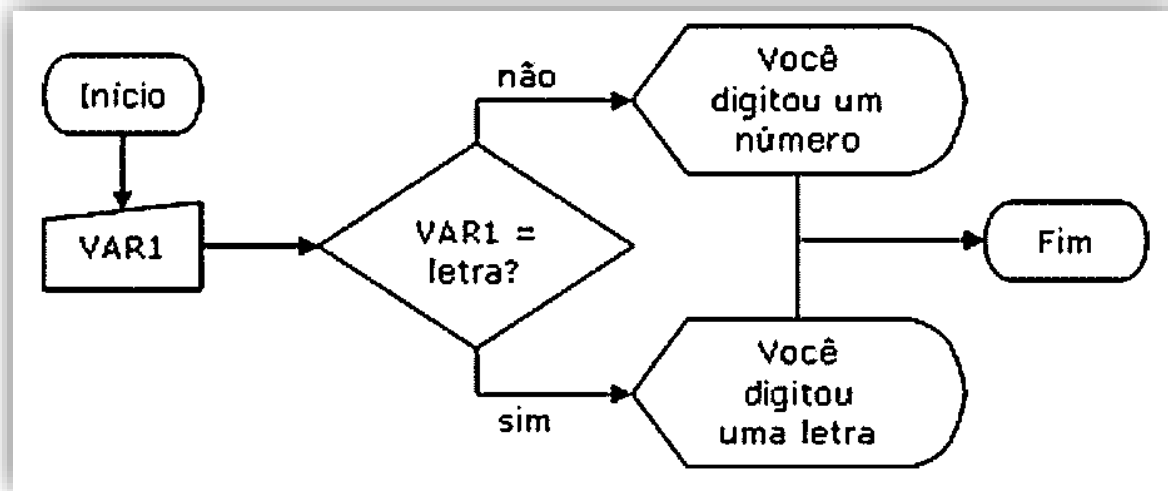
Quais serão as **saída de dados**?
Quais são os **dados de entrada**?
Qual será o **processamento** a ser utilizado?



Representação de Algoritmos

ALGORITMO = Sequência lógica

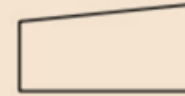
FLUXOGRAMA OU DIAGRAMA DE BLOCOS



Quais serão as **saída de dados**?
Quais são os **dados de entrada**?
Qual será o **processamento** a ser utilizado?



= Início e final do Diagrama;



= Entrada de dados;



= Impressão de dados pela impressora;



= Impressão de dados em vídeo;



= Operação de atribuição de chamada ou retorno;



= Decisão;



= Seta de fluxo de dados;



= Conector utilizado quando é preciso particionar um programa. Assim, coloca-se um número ou um símbolo para indicar a conexão;



= Conector específico para indicar conexão do fluxo da execução em outra página;



= Preparação – operações que utilizam uma chave que modificam a execução de um programa.

Representação de Algoritmos

ALGORITMO = Sequência lógica

PSEUDOCÓDIGOS

Algoritmo <nome_do_algoritmo>

<declaração_de_variáveis>

<subalgoritmos>

Início

<corpo_do_algoritmo>

Fim.

Comandos básicos:

ESCREVA ou ESCREVER: Comando para exibir ao usuário um texto ou valor.

LEIA ou LER: Comando que receberá um valor digitado pelo usuário atribuindo a uma variável criada.

1- Faça um Programa que mostre a mensagem "Alo mundo" na tela.

```
ALGORITMO Programa_Alo
INICIO
    ESCREVER "Alo mundo";
FIM.
```

2- Faça um Programa que leia o nome do usuário mostre a mensagem "Olá <nome_do_usuario>"

```
ALGORITMO OlaUsuario
VAR nome : literal
INICIO
    ESCREVER "Qual é o seu nome? ",
    LER nome;
    ESCREVER "Olá ", nome;
FIM.
```


Representação de Algoritmos

ALGORITMO = Sequência lógica

PSEUDOCÓDIGOS

Algoritmo <nome_do_algoritmo>

<declaração_de_variáveis>

<subalgoritmos>

Início

<corpo_do_algoritmo>

Fim.

3- Faça um Programa que peça dois números e imprima a soma.

ALGORITMO Programa_Soma

VAR num1, num2, soma : INTEIRO;

INICIO

num1<- 0;

num2<- 0;

soma <- 0;

ESCREVER "Digite o primeiro número : ";

LER (num1);

ESCREVER "Digite o segundo número : ";

LER (num2);

soma <- num1 + num2;

ESCREVER "A soma é: ", soma;

FIM.

Representação de Algoritmos

ALGORITMO = Sequência lógica

PSEUDOCÓDIGOS

Algoritmo <nome_do_algoritmo>

<declaração_de_variáveis>

<subalgoritmos>

Início

<corpo_do_algoritmo>

Fim.

Algoritmo Média

Var N1, N2, Média

Início

Leia N1, N2

Média := (N1+N2)/2

Se Média >= 7

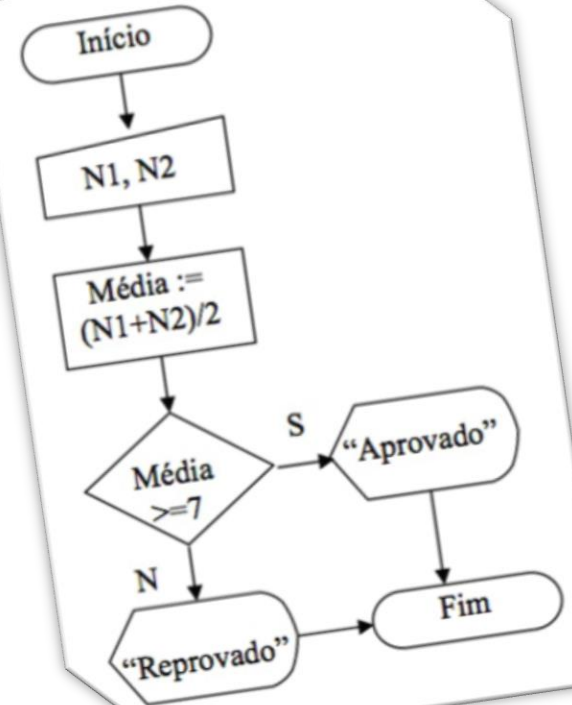
Então

Escreva "Aprovado"

Senão

Escreva "Reprovado"

Fim.



Representação de Algoritmos

ALGORITMO = Sequência lógica

PSEUDOCÓDIGOS

Algoritmo <nome_do_algoritmo>

<declaração_de_variáveis>

<subalgoritmos>

Início

<corpo_do_algoritmo>

Fim.

1. algoritmo CalcularMedia

2. var

3. nota1, nota2, nota3, media : real

4.

5. início

6. escrever ("Digite a primeira nota:")

7. ler (nota1)

8. escrever ("Digite a segunda nota:")

9. ler (nota2)

10. escrever ("Digite a terceira nota:")

11. ler (nota3)

12. $media \leftarrow (nota1 + nota2 + nota3) / 3$

13. escrever (media)

14. se $media \geq 7$ então

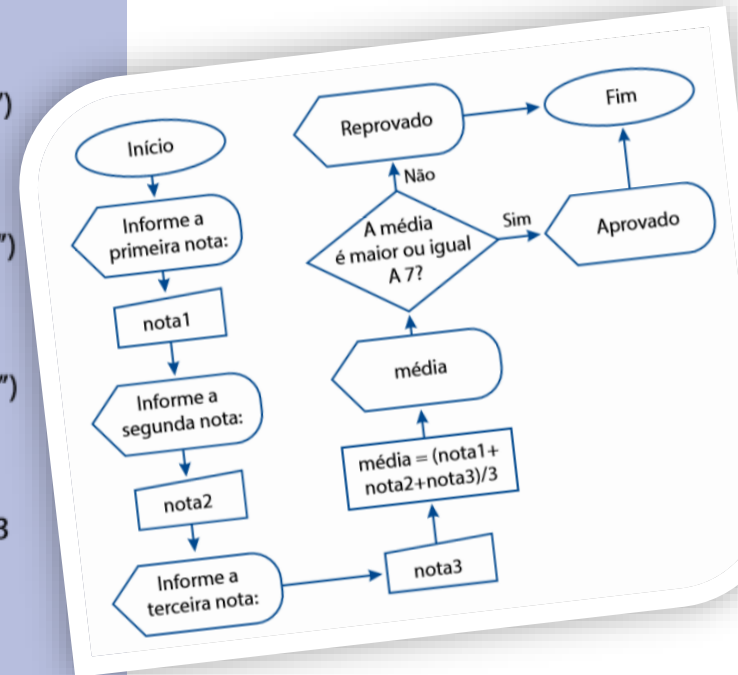
15. escrever ("Aprovado")

16. senão

17. escrever ("Reprovado")

18. fimse

19. fim.



Variáveis ↷



Lógica de Programação

Tipos de dados e Operadores

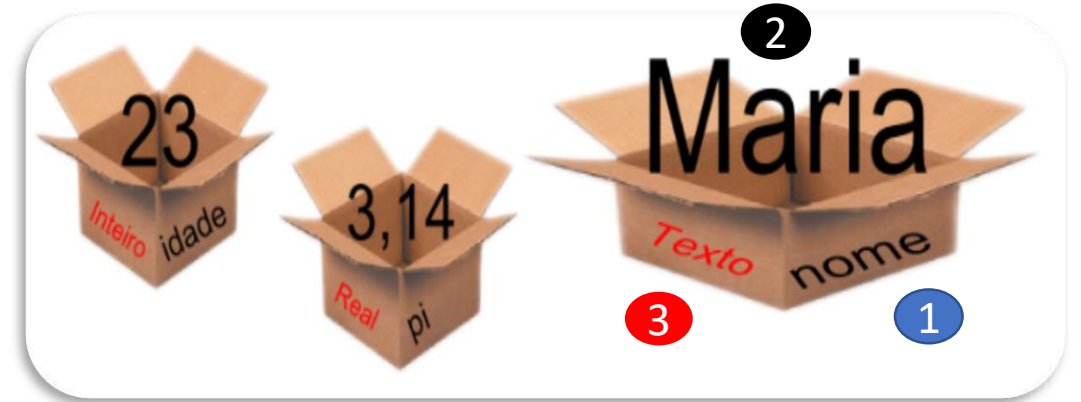
Variáveis

são endereços/posições de memória RAM

1-Identificador

2-Valor

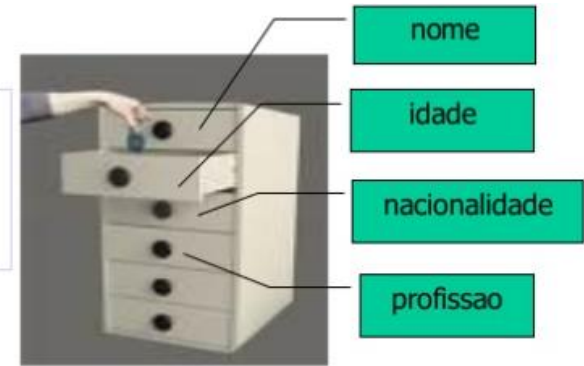
3-Tipo de Dado



Identificador (nomes dados pelo programador à variável)

- a) **Palavras:** palavras com letras maiúsculas ou minúsculas, sem espaço e sem acento.
- b) **Números:** podem ser utilizados desde que apareçam após uma ou mais letras.
- c) **Underline (_) e cifrão (\$):** são considerados caracteres válidos e são aceitos em qualquer local do identificador. Estes caracteres não são muito utilizados, porém, algumas poucas linguagens utilizam o underline para identificar variáveis que tenham espaço no nome.

Armário ↔
Memória do
computador



Exemplos de nomes de variáveis:

Salario - correto ;

1ANO - errado (não começou uma letra);

ANO1 - correto;

a casa - errado (contém espaço em branco);

Convenção Mundial (padrão): “*minusculoMaiusculo*”

nome

nomeCompleto

valorDescontoCedido

SAL/HORA - errado (contém caractere especial “/”)

SAL_HORA_DESCONTO - correto;

Variáveis

são endereços/posições de memória RAM

1-Identificador

2-Valor

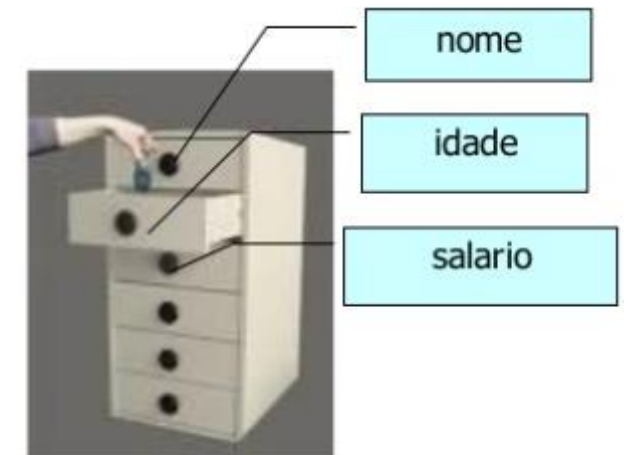
3-Tipo de Dado



Valor

(conteúdo da variável)

Uma variável assume um único valor por vez.



Variáveis

são endereços/posições de memória RAM

1-Identificador

2-Valor

3-Tipo de Dado



Tipo de Dado

(define que tipos de valores a variável poderá receber)

Tipos de declarações de variáveis:

- **Variável Global:** todo o código poderá acessá-la.
- **Variável Locais:** possui um acesso mais restrito, declarada dentro de um bloco de código específico.



Reduzindo o escopo das variáveis torna o código mais seguro a bugs. A solução básica é utilizando **variáveis locais** e **funções com passagem de valor**.

→ Existem Linguagens de Programação Tipadas (Java, C++ e C#) e as **não tipadas** (Javascript e PHP).

→ **Constantes:** são variáveis com valores fixos, imutáveis, que devem ser definidos no início do algoritmo.

Variáveis

são endereços/posições de memória RAM

1-Identificador

2-Valor

3-Tipo de Dado



Tipo de Dado

(define que tipos de valores a variável poderá receber)

Tipada (Fortemente Tipada)

Definição de variáveis em algoritmos

```
VAR  NOME :      literal
      IDADE :     inteiro
      SALARIO:    real
      TEM_FILHOS: lógico
```

Não tipada (Fracamente tipada / dinamicamente tipada):

x = 15

y = "Rogerão Araújo"

z = 1.84

Variáveis

são endereços/posições de memória RAM

1-Identificador

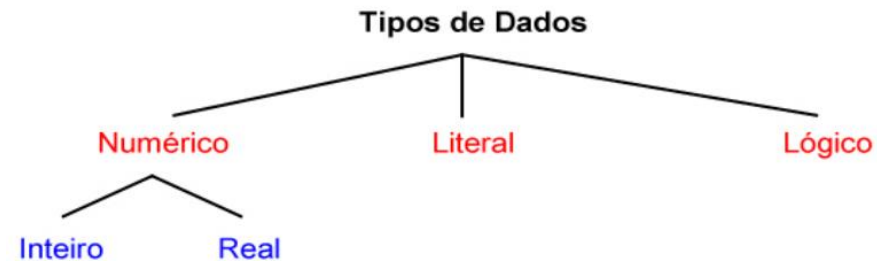
2-Valor

3-Tipo de Dado



Tipo de Dado

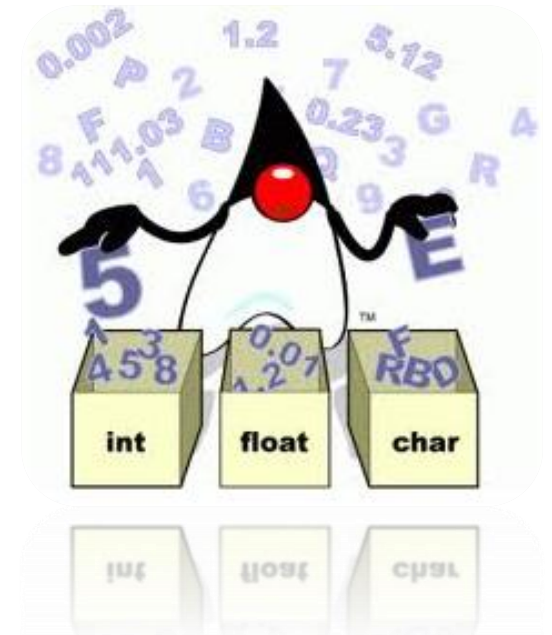
(define que tipos de valores a variável poderá receber)



Os 4 tipos de dados base:

- **Inteiro:** tipo numérico que define números inteiros negativos e positivos;
- **Real:** tipo numérico que define números decimais com vírgula negativos e positivos;
- **Lógico:** tipo de dado que aceita apenas dois valores, verdadeiro (1) ou falso (0);
- **Literal:** tipo de dado que define cadeias de caracteres, aceitando letras, números e símbolos.

Cada tipo ocupa um espaço de memória → Otimização



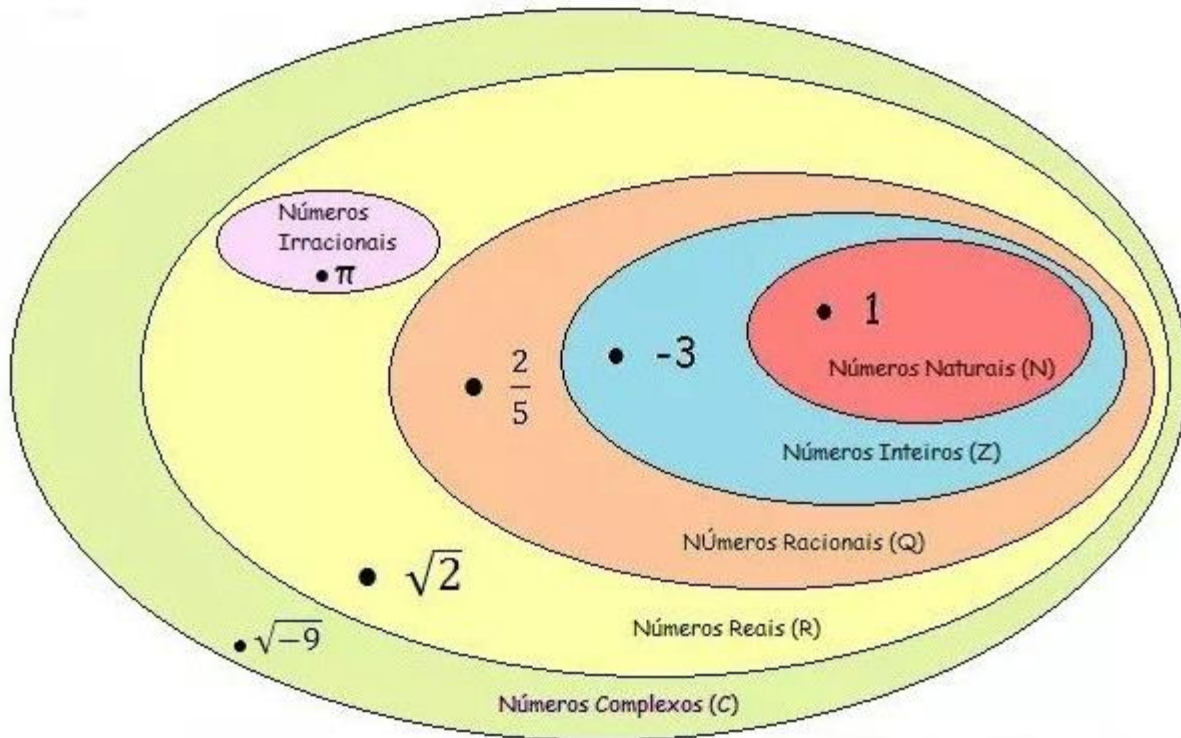
Variáveis

são endereços/posições de memória RAM

Tipo de Dado

Numéricos: **Inteiro**
Real

Conjuntos Numéricos



Conjuntos

$$\mathbb{N} = 0, 1, 2, 3, 4, \dots$$

(Naturais)

*Zero é o primeiro número natural

$$\mathbb{Z} = \dots -2, -1, 0, 1, 2, 3, \dots$$

(Inteiros)

*Acrescenta os negativos

$$\mathbb{Q} = \dots -1, 0, 1, 2, \dots \text{ e frações}$$

(Racionais)

*Dízimas periódicas são frações

$$\mathbb{I} = \text{Só as não frações}$$

(Irracionais)

*Raízes NÃO inteiras

$\pi = 3,1415\dots$ $\sqrt{2}$, $\sqrt{3}$, $\sqrt{5}$... *Dízimas NÃO periódicas

$$\mathbb{R} = \text{TODOs os anteriores !!!} \uparrow$$

(Reais)

$$\mathbb{C} = \text{Núm. Imaginários (i)}$$

(Complexos)

*Raízes Quad. Negativas

$\sqrt{-4}$ $z = a + bi$ $3i$



Quando usar cada tipo?

- Quantidade de filhos
- Idade
- Altura
- Salário
- Nota
- Média
- largura
- Gramas
- Km
- Contador

Se eu preciso calcular alguma média ou realizar algum cálculo com virgula, então usar REAL

Variáveis

são endereços/posições de memória RAM

Tipo de Dado

Definição de variáveis em algoritmos

```
VAR  NOME :      literal
      IDADE :     inteiro
      SALARIO:    real
      TEM_FILHOS: lógico
```

Forma Geral: <tipo_de_dado> <Identificador>

Exemplos:

- inteiro idade
- real nota1, nota2, nota3
- caracter conceito
- cadeia sobrenome

Exercícios:

1- Identifique o tipo de dados dos seguintes valores:

- (a) "9 de agosto de 1968"
- (b) 1.3
- (c) Falso
- (d) -31
- (e) "?"

2- Escreva o tipo de dado ideal para se representar cada uma das seguintes informações:

- (a) O nome de uma rua
- (b) Número de quartos de uma casa.
- (c) Se uma pessoa é diabética ou não
- (d) O saldo de uma conta bancária
- (e) O resultado de uma operação de raiz quadrada

Variáveis

são endereços/posições de memória RAM

Tipo de Dado

Exercícios:

1- Identifique o tipo de dados dos seguintes valores:

- (a) "9 de agosto de 1968" **Literal**
- (b) 1.3 **Real**
- (c) Falso **Lógico**
- (d) -31 **Inteiro**
- (e) "?" **Literal**

Inteiro

Real

Lógico

Literal



2- Escreva o tipo de dado ideal para se representar cada uma das seguintes informações:

- (a) O nome de uma rua **Literal**
- (b) Número de quartos de uma casa. **Inteiro**
- (c) Se uma pessoa é diabética ou não **Lógico**
- (d) O saldo de uma conta bancária **Real**
- (e) O resultado de uma operação de raiz quadrada **Real**

Operadores

Usadas para representar expressões de cálculo, comparação, condição e expressão.
Quatro tipos principais:

Operadores de Atribuição

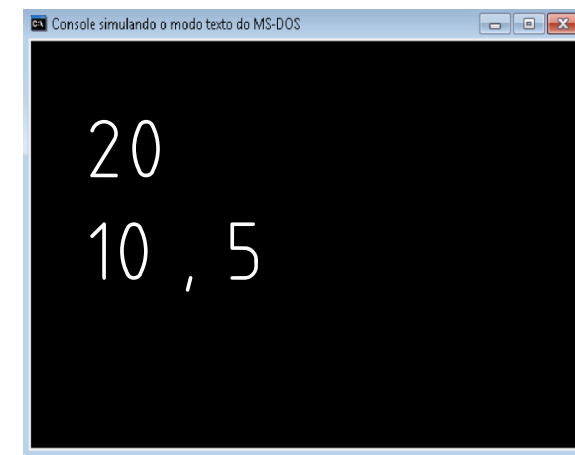
Exercício: Analise os algoritmos feitos em **Diagrama de Chapin** abaixo e diga o que será impresso na tela ao serem executados:

a)

$A \leftarrow 10$
$B \leftarrow 20$
Escrever B
$B \leftarrow 5$
Escrever A, B

TESTE DE MESA		
LINHA	COMANDO	
5	$A \leftarrow 10$	
6	$B \leftarrow 20$	
7	Escrever B	
8	$B \leftarrow 5$	
9	Escrever A, B	

LINHA	A	B
5	[10]	?
6	10	[20]
7	10	{20}
8	10	[5]
9	{10}	{5}



Operadores

Usadas para representar expressões de cálculo, comparação, condição e expressão.
Quatro tipos principais:

Operadores de Atribuição

Exercício: Analise os algoritmos feitos em **Diagrama de Chapin** abaixo e diga o que será impresso na tela ao serem executados:

b)

$A \leftarrow 30$
$B \leftarrow 20$
$C \leftarrow A + B$
Escrever C
$B \leftarrow 10$
Escrever B, C
$C \leftarrow A + B$
Escrever A, B, C

c)

$A \leftarrow 10$
$B \leftarrow 20$
$C \leftarrow A$
$B \leftarrow C$
$A \leftarrow B$
Escrever A, B, C

d)

$A \leftarrow 10$
$B \leftarrow A + 1$
$A \leftarrow B + 1$
$B \leftarrow A + 1$
Escrever A
$A \leftarrow B + 1$
Escrever A, B

e)

$A \leftarrow 10$
$B \leftarrow 5$
$C \leftarrow A + B$
$B \leftarrow 20$
$A \leftarrow 10$
Escrever A, B, C

f)

$X \leftarrow 1$
$Y \leftarrow 2$
$Z \leftarrow Y - X$
Escrever Z
$X \leftarrow 5$
$Y \leftarrow X + Z$
Escrever X, Y, Z



Operadores

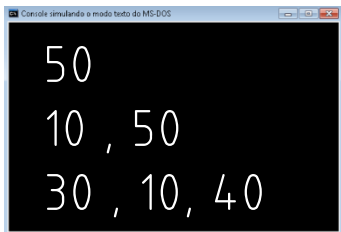
Usadas para representar expressões de cálculo, comparação, condição e expressão.
Quatro tipos principais:

Operadores de Atribuição

Exercício: Analise os algoritmos feitos em **Diagrama de Chapin** abaixo e diga o que será impresso na tela ao serem executados:

b)

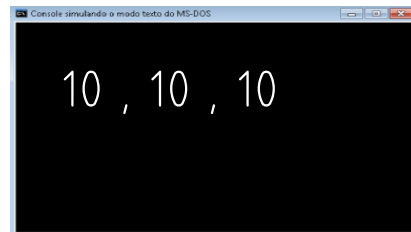
$A \leftarrow 30$
$B \leftarrow 20$
$C \leftarrow A + B$
Escrever C
$B \leftarrow 10$
Escrever B, C
$C \leftarrow A + B$
Escrever A, B, C



```
50
10 , 50
30 , 10, 40
```

c)

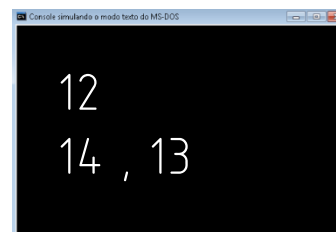
$A \leftarrow 10$
$B \leftarrow 20$
$C \leftarrow A$
$B \leftarrow C$
$A \leftarrow B$
Escrever A, B, C



```
10 , 10 , 10
```

d)

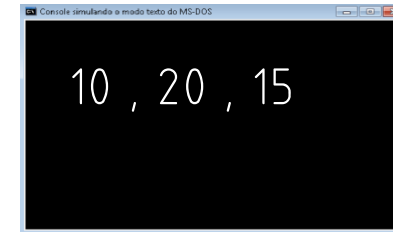
$A \leftarrow 10$
$B \leftarrow A + 1$
$A \leftarrow B + 1$
$B \leftarrow A + 1$
Escrever A
$A \leftarrow B + 1$
Escrever A, B



```
12
14 , 13
```

e)

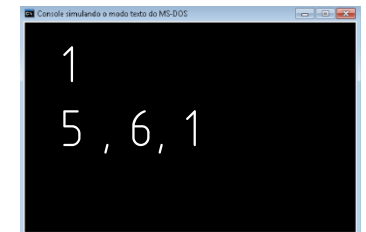
$A \leftarrow 10$
$B \leftarrow 5$
$C \leftarrow A + B$
$B \leftarrow 20$
$A \leftarrow 10$
Escrever A, B, C



```
10 , 20 , 15
```

f)

$X \leftarrow 1$
$Y \leftarrow 2$
$Z \leftarrow Y - X$
Escrever Z
$X \leftarrow 5$
$Y \leftarrow X + Z$
Escrever X, Y, Z



```
1
5 , 6, 1
```

Teste de Mesa

Ponto de Interrogação (?): significa que nesta linha ainda não foi definido um valor para a variável.

Valor entre colchetes ([]): significa que a variável foi lida, ou seja, o valor foi obtido a partir de uma entrada de dados.

Valor entre chaves ({ }): significa que o valor da variável foi escrito, ou seja, o valor foi apresentado por meio de uma saída de dados.

Escreva um algoritmo para ler dois números (a e b) e trocar os seus valores. Exibir os valores de a e b após a troca

- **SAÍDA:** Quais os resultados que o algoritmo deve fornecer ?
- **ENTRADA:** Quais os insumos necessários para se obter a saída?
- **PROCESSAMENTO:** Como transformar os insumos na saída?

- **SAÍDA:** valores de a e b (trocados)
- **ENTRADA:** dois números (A, B)
- **PROCESSAMENTO:**
 - A = B
 - B = A

Algoritmo

```
5 inteiro a, b
6 escreva ("Digite o valor de a: ")
7 leia(a)
8 escreva ("Digite o valor de b: ")
9 leia(b)
10 a = b
11 b = a
12 escreva("\n a = ", a)
13 escreva("\n b = ", b)
```

Neste **Teste de Mesa** foram inseridos no programa (o programa leu) os valores 12 para a variável a e 3 para a variável b.

Teste de Mesa

LINHA	a	b
7	(12)	?
9		(3)
10	3	
11		3
12	{3}	
13		{3}

Nota-se que o RESULTADO exibido pelo programa é **INVÁLIDO**, ou seja, não atende ao enunciado. O programa deveria ter exibido **3** e **12**.

O teste de mesa cumpriu seu objetivo que foi "detectar" o ERRO.

Teste de Mesa

Ponto de Interrogação (?): significa que nesta linha ainda não foi definido um valor para a variável.

Valor entre colchetes ([]): significa que a variável foi lida, ou seja, o valor foi obtido a partir de uma entrada de dados.

Valor entre chaves ({ }): significa que o valor da variável foi escrito, ou seja, o valor foi apresentado por meio de uma saída de dados.

Escreva um algoritmo para ler dois números (a e b) e trocar os seus valores. Exibir os valores de a e b após a troca

- **SAÍDA:** Quais os resultados que o algoritmo deve fornecer ?
- **ENTRADA:** Quais os insumos necessários para se obter a saída?
- **PROCESSAMENTO:** Como transformar os insumos na saída?

- **SAÍDA:** valores de a e b (trocados)
- **ENTRADA:** dois números (A, B)
- **PROCESSAMENTO:**
 - $A = B$
 - $B = A$

Nova solução

Algoritmo

```

5  inteiro a, b, aux
6  escreva ("Digite o valor de a: ")
7  leia(a)
8  escreva ("Digite o valor de b: ")
9  leia(b)
10 aux = a
11 a = b
12 b = aux
13 escreva("\n a = ", a)
14 escreva("\n b = ", b)
    
```

Neste **Teste de Mesa** foram inseridos no programa (o programa leu) os valores 12 para a variável **a** e 3 para a variável **b**.

Teste de Mesa

LINHA	a	b	aux
7	(12)	?	?
9		(3)	
10			12
11	3		
12		12	
13	{3}		
14		{12}	

Para efetuar a troca necessita-se de uma **TERCEIRA** variável (auxiliar)

Teste de Mesa

Ponto de Interrogação (?): significa que nesta linha ainda não foi definido um valor para a variável.

Valor entre colchetes ([]): significa que a variável foi lida, ou seja, o valor foi obtido a partir de uma entrada de dados.

Valor entre chaves ({}): significa que o valor da variável foi escrito, ou seja, o valor foi apresentado por meio de uma saída de dados.

Neste Teste de mesa foram inseridos no programa (o programa leu) os valores 8 para **Nota1**, 5 para **Nota2** e 8 para a **Nota3**. O valor 7 é escrito na tela na linha 8

LINHA	NOTA1	NOTA2	NOTA3	MEDIA
1	?	?	?	?
2	[8]	?	?	?
3	8	?	?	?
4	8	[5]	?	?
5	8	5	?	?
6	8	5	[8]	?
7	8	5	8	7
8	8	5	8	{7}

Quadro 3 - Exemplo de teste de mesa 2

Algoritmo

1. algoritmo CalcularMedia
2. var
3. nota1, nota2, nota3, media : real
- 4.
5. início
6. escrever ("Digite a primeira nota:")
7. ler (nota1)
8. escrever ("Digite a segunda nota:")
9. ler (nota2)
10. escrever ("Digite a terceira nota:")
11. ler (nota3)
12. media <- (nota1+nota2+nota3)/3
13. escrever (media)
14. se media >= 7 então
15. escrever ("Aprovado")
16. senão
17. escrever ("Reprovado")
18. fimse
19. fim.

Teste de Mesa

Agora sua vez!

Ponto de Interrogação (?): significa que nesta linha ainda não foi definido um valor para a variável.

Valor entre colchetes ([]): significa que a variável foi lida, ou seja, o valor foi obtido a partir de uma entrada de dados.

Valor entre chaves ({}): significa que o valor da variável foi escrito, ou seja, o valor foi apresentado por meio de uma saída de dados.

Neste Teste de mesa serão inseridos no programa os valores 6 para **Nota1**, 8 para **Nota2** e 3 para a **Nota3**.

Linha	NOTA1	NOTA2	NOTA3	MEDIA
1	?	?	?	?
2	?	?	?	?
3	?	?	?	?
4	?	?	?	?
5	?	?	?	?
6	?	?	?	?
7	?	?	?	?
8	?	?	?	?

Teste de Mesa

Algoritmo

1. algoritmo CalcularMedia
2. var
3. nota1, nota2, nota3, media : real
- 4.
5. início
6. escrever ("Digite a primeira nota:")
7. ler (nota1)
8. escrever ("Digite a segunda nota:")
9. ler (nota2)
10. escrever ("Digite a terceira nota:")
11. ler (nota3)
12. media <- (nota1+nota2+nota3)/3
13. escrever (media)
14. se media >= 7 então
15. escrever ("Aprovado")
16. senão
17. escrever ("Reprovado")
18. fimse
19. fim.

Teste de Mesa

Agora sua vez!
RESPOSTA

Ponto de Interrogação (?): significa que nesta linha ainda não foi definido um valor para a variável.

Valor entre colchetes ([]): significa que a variável foi lida, ou seja, o valor foi obtido a partir de uma entrada de dados.

Valor entre chaves ({ }): significa que o valor da variável foi escrito, ou seja, o valor foi apresentado por meio de uma saída de dados.

Neste Teste de mesa serão inseridos no programa os valores 6 para **Nota1**, 8 para **Nota2** e 3 para a **Nota3**.

Linha	NOTA1	NOTA2	NOTA3	MEDIA
1	?	?	?	?
2	[6]	?	?	?
3	6	?	?	?
4	6	[8]	?	?
5	6	8	?	?
6	6	8	[3]	?
7	6	8	3	?
8	6	8	3	{5,67}

Teste de Mesa

Algoritmo

1. algoritmo CalcularMedia
2. var
3. nota1, nota2, nota3, media : real
- 4.
5. início
6. escrever ("Digite a primeira nota:")
7. ler (nota1)
8. escrever ("Digite a segunda nota:")
9. ler (nota2)
10. escrever ("Digite a terceira nota:")
11. ler (nota3)
12. media <- (nota1+nota2+nota3)/3
13. escrever (media)
14. se media >= 7 então
15. escrever ("Aprovado")
16. senão
17. escrever ("Reprovado")
18. fimse
19. fim.

Teste de Mesa

Agora sua vez!

Algoritmo

001 ALGORITMO CaluloEstranho

002 VAR num1,num2, resultado : inteiro;

003 INICIO

004 resultado = 0;

005 num1 = 0;

006 num2 = 0;

007 ESCREVA ("Digite o num1: ");

008 Leia (num1);

009 resultado = 10 + num1;

010 Escreva ("Digite o num2: ");

011 Leia (num2);

012 resultado = resultado + num2;

013 ESCREVA ("O resultado será : " resultado);

014 resultado = resultado * 2;

015 ESCREVA ("O dobro do resultado será : " resultado);

016 FIM

Para realizar o teste vamos simular que o usuário digitou 5 para o num1 e depois digitou 100 para num2

Linha	num1	num2	resultado	Escreve
002	?	?	?	?
003	?	?	?	?
004	?	?	?	?
005	?	?	?	?
006	?	?	?	?
007	?	?	?	?
008	?	?	?	?
009	?	?	?	?
010	?	?	?	?
011	?	?	?	?
012	?	?	?	?
013	?	?	?	?
014	?	?	?	?
015	?	?	?	?
016	Fim do programa			

Teste de Mesa

Agora sua vez!
RESPOSTA

Algoritmo

001 ALGORITMO CaluloEstranho

002 VAR num1,num2, resultado : inteiro;

003 INICIO

004 resultado = 0;

005 num1 = 0;

006 num2 = 0;

007 ESCREVA ("Digite o num1: ");

008 Leia (num1);

009 resultado = 10 + num1;

010 Escreva ("Digite o num2: ");

011 Leia (num2);

012 resultado = resultado + num2;

013 ESCREVA ("O resultado será : " resultado);

014 resultado = resultado * 2;

015 ESCREVA ("O dobro do resultado será : " resultado);

016 FIM

Para realizar o teste vamos simular que o usuário digitou **5 para o num1** e depois digitou **100 para num2**

Linha	num1	num2	resultado	Escreve
002	?	?	?	?
003	?	?	?	?
004	?	?	0	?
005	0	?	0	?
006	0	0	0	?
007	0	0	0	Digite o num1:
008	5	0	0	?
009	5	0	15	?
010	5	0	15	Digite o num2:
011	5	100	15	?
012	5	100	115	?
013	5	100	115	O resultado será: 115
014	5	100	230	?
015	5	100	120	O dobro do resultado será : 230
016	Fim do programa			

Operadores

Usadas para representar expressões de cálculo, comparação, condição e expressão.
Quatro tipos principais:

Operadores Aritméticos

OPERADOR	SÍMBOLO	EXEMPLO
Adição	+	2 + 2
Subtração	-	2 - 2
Multiplicação	*	2 * 2
Divisão de inteiro	DIV	2 DIV 2
Divisão de real	/	2 / 2
Módulo	MOD	4 % 3

soma1 <- 5 + 10
soma2 <- 10 + 15 + 20

multiplicacao <- 10 * 4

subtracao <- 10 - 3

$$35 \div 4 = 9$$

$$35 \div 5 = 8,75$$

$$\text{divisao1} = 36 \text{ div } 4 = 9$$

$$\text{divisao2} = 35 \text{ div } 4 = 8$$

$$\begin{aligned} \text{divisao3} &<- 36 / 4 = 9 \\ \text{divisao4} &<- 35 / 4 = 8,75 \end{aligned}$$

(Tipo Real)

$$\begin{array}{r} 35 \overline{) 4} \\ \underline{-32} \\ 3 \end{array}$$

$$\text{resto} <- 35 \text{ MOD } 4$$

$$\text{resto} <- 4 \text{ MOD } 5 = 4$$

Dica: $4 < 5$

Regras de prioridade da matemática:

- 1º – Parênteses;
- 2º – Módulo, divisão e multiplicação;
- 3º – Adição e subtração.

$$\begin{aligned} &5 * (3 + 4) + 4 \\ &5 * 7 + 4 \\ &35 + 4 \\ &39 \end{aligned}$$

$$\begin{aligned} &5 * 4 / 2 \text{ MOD } 6 \\ &20 / 2 \text{ MOD } 6 \\ &10 \text{ MOD } 6 \\ &4 \end{aligned}$$

Operadores

Usadas para representar expressões de cálculo, comparação, condição e expressão.

Operadores Aritméticos

Os pares de instruções abaixo produzem o mesmo resultado?

$A \leftarrow (4/2)+(2/4)$ e $A \leftarrow 4/2+2/4$

$B \leftarrow 4/(2+2)/4$ e $B \leftarrow 4/2+2/4$

$C \leftarrow (4+2)*2-4$ e $C \leftarrow 4+2*2-4$

Exercício: Reescreva as instruções abaixo com o mínimo de parênteses possível, mas sem alterar o resultado:

$A \leftarrow 6*(3+2)$	$F \leftarrow (6/3)+(8/2)$
$B \leftarrow 2+(6*(3+2))$	$G \leftarrow ((3+(8/2))*4)+(3*2)$
$C \leftarrow 2+(3*6)/(2+4)$	$H \leftarrow (6*(3*3)+6)-10$
$D \leftarrow 2*(8/(3+1))$	$I \leftarrow (((10*8)+3)*9)$
$E \leftarrow 3+(16-2)/(2*(9-2))$	$J \leftarrow ((-12)*(-4))+(3*(-4))$

Operadores

Usadas para representar expressões de cálculo, comparação, condição e expressão.
Quatro tipos principais:

Operadores Aritméticos

Os pares de instruções abaixo produzem o mesmo resultado?

$A \leftarrow (4/2)+(2/4)$	e	$A \leftarrow 4/2+2/4$	✓
$B \leftarrow 4/(2+2)/4$	e	$B \leftarrow 4/2+2/4$	✗
$C \leftarrow (4+2)*2-4$	e	$C \leftarrow 4+2*2-4$	✗

Exercício: Reescreva as instruções abaixo com o mínimo de parênteses possível, mas sem alterar o resultado:

$A \leftarrow 6*(3+2)$	✓	$F \leftarrow (6/3)+(8/2)$	$6/3 + 8/2$
$B \leftarrow 2+(6*(3+2))$	$2 + 6 * (3+2)$	$G \leftarrow ((3+(8/2))*4)+(3*2)$	$(3 + 8/2)*4 + 3*2$
$C \leftarrow 2+(3*6)/(2+4)$	$2 + 3*6/(2+4)$	$H \leftarrow (6*(3*3)+6)-10$	$6*3*3 + 6 -10$
$D \leftarrow 2*(8/(3+1))$	$2*8/(3+1)$	$I \leftarrow (((10*8)+3)*9)$	$(10*8 + 3)*9$
$E \leftarrow 3+(16-2)/(2*(9-2))$	✓	$J \leftarrow ((-12)*(-4))+(3*(-4))$	$(-12)*(-4) + 3*(-4)$

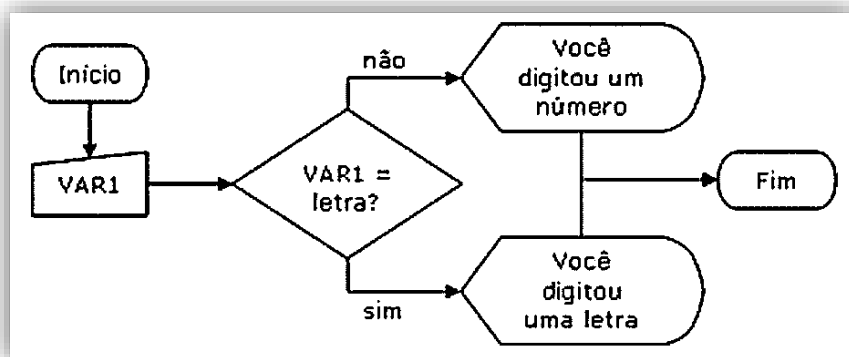
Operadores

Usadas para representar expressões de cálculo, comparação, condição e expressão.
Quatro tipos principais:

Operadores Relacionais

OPERADOR	SÍMBOLO	EXEMPLO
Igualdade	==	5=5
Diferença	<>	5<>4
Menor que	<	4<5
Menor ou igual que	<=	5<=5
Maior que	>	5>4
Maior ou igual que	>=	5>=5

$$\begin{array}{ccc}
 2 * 4 & == & 24 / 3 \\
 8 & == & 8 \\
 & \text{V} &
 \end{array}$$



se media >= 7 então
 escrever ("Aprovado")
 senão
 escrever ("Reprovado")

$$\begin{array}{ccc}
 2 + 8 \% 7 & >= & 3 * 6 - 15 \\
 2 + 1 & >= & 18 - 15 \\
 3 & >= & 3 \\
 & \text{V} &
 \end{array}$$

Operadores

Usadas para representar expressões de cálculo, comparação, condição e expressão.
Quatro tipos principais:

Operadores Lógicos

E , OU, NÃO, XOU

a E b		
a	b	Resultado
Verdadeiro	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Falso
Falso	Verdadeiro	Falso
Falso	Falso	Falso

a OU b		
a	b	Resultado
Verdadeiro	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Verdadeiro
Falso	Verdadeiro	Verdadeiro
Falso	Falso	Falso

a	Resultado NÃO
Verdadeiro	Falso
Falso	Verdadeiro

Regras de prioridade:

1º - NÃO;
2º - E;
3º - OU.

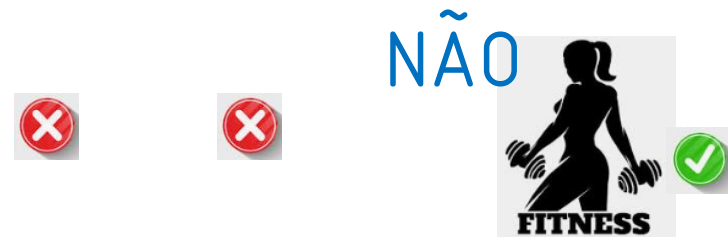
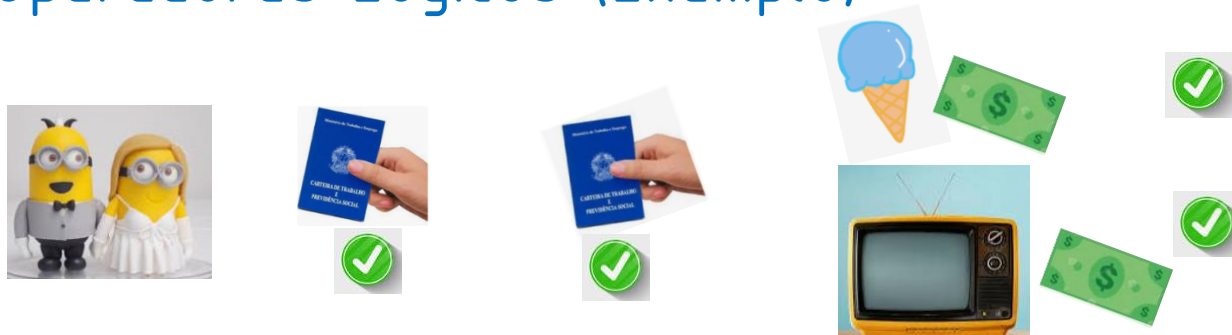


a XOU b		
a	b	Resultado
Verdadeiro	Verdadeiro	Falso
Verdadeiro	Falso	Verdadeiro
Falso	Verdadeiro	Verdadeiro
Falso	Falso	Falso

Operadores

Usadas para representar expressões de cálculo, comparação, condição e expressão.
Quatro tipos principais:

Operadores Lógicos (Exemplo)



Operadores

Usadas para representar expressões de cálculo, comparação, condição e expressão.
Quatro tipos principais:

Operadores Lógicos

E , OU, NÃO, XOU

Exemplos:

$3 + 2 \leq 4$ OU $7 = 3 * 2$

$5 \leq 4$ OU $7 \leq 6$

false OU false

false

$3 * 3 < 8$ E $7 * 3 / 3 = 7$

$9 < 8$ E $7 * 1 = 7$

$9 < 8$ E $7 = 7$

false E true

false

$5 + 5 < 10 * (17 - 15)$ E NÃO $4 > 3$ E $5 \leq 4$ OU $7 \text{ MOD } 3 < 8 \text{ DIV } 3$
 $5 + 5 < 10 * 2$ E NÃO $4 > 3$ E $5 \leq 4$ OU $7 \text{ MOD } 3 < 8 \text{ DIV } 3$
 $10 < 20$ E NÃO $4 > 3$ E $5 \leq 4$ OU $1 < 2$
verdadeiro E NÃO verdadeiro E verdadeiro OU verdadeiro
verdadeiro E falso E verdadeiro OU verdadeiro
falso E verdadeiro OU verdadeiro
falso OU verdadeiro
VERDADEIRO

Teste de Mesa

Algoritmo

```
001 ALGORITMO CaluloEstranho

002 VAR num1,num2, resultado : inteiro;
003 INICIO
004 resultado = 0;
005 num1 = 0;
006 num2 = 0;
007 ESCREVA ("Digite o num1: ");
008 Leia (num1);
009 resultado = 10 + num1;
010 Escreva ("Digite o num2: ");
011 Leia (num2);
012 resultado = resultado + num2;
013 ESCREVA ("O resultado será : " resultado);
014 resultado = resultado * 2;
015 ESCREVA ("O dobro do resultado será : " resultado);
016 FIM
```

Teste de Mesa
Para realizar o teste vamos simular que o usuário digitou 20 para o num1 e depois digitou 30 para num2

Linha	num1	num2	resultado	Escreve
002	?	?	?	?
003	?	?	?	?
004	?	?	0	?
005	0	?	0	?
006	0	0	0	?
007	0	0	0	Digite o num1:
008	20	0	0	?
009	20	0	30	?
010	20	0	30	Digite o num2:
011	20	30	30	?
012	20	30	60	?
013	20	30	60	O resultado será: 60
014	20	30	120	?
015	20	30	120	O dobro do resultado será : 120
016	Fim do programa			

Agora sua vez!
Para verificar se entendeu, você poderia refazer esse teste de mesa com duas outras entradas e enviar aqui como resposta a tabela. Por exemplo, 5 para num1 e 100 para num2.

Estudando Lógica de Programação

Algoritmos **nã**o se aprendem
Copiando algoritmos / Estudando algoritmos

Algoritmos **só** se aprendem
Construindo algoritmos / Testando algoritmos



Lista de exercícios – Estrutura Sequencial
<https://wiki.python.org.br/EstruturaSequencial>



Entender o Problema



Requisitos, entradas e saídas



Fluxograma e Pseudocódigo



Algoritmos



Todo mundo deveria aprender a programar (What most schools don't teach - dublado):

<https://www.youtube.com/watch?v=4iKu9qtCSXg>

O QUE POSSO FAZER COM PROGRAMAÇÃO?

https://www.youtube.com/watch?v=EvHeuJs_hcQ

Afinal, o que é uma linguagem de programação?

<https://www.youtube.com/watch?v=PmefpISZ7Ew&list=PL99AE7FAAE4560E2A&index=1>

Afinal, o que é um algoritmo?

<https://www.youtube.com/watch?v=yTKiRkCi0Bg&index=3&list=PL99AE7FAAE4560E2A>

Lógica de Programação com VisualG- **Prof. Thiago de Oliveira** (Vários conteúdos para estudo e aprofundamento)

<https://sites.google.com/a/ifmg.edu.br/troliveira/disciplinas/logica-de-programacao>

Nesse vídeo temos dois exercícios básicos feitos e simulados no VisualG:

<https://www.youtube.com/watch?v=mz7xFwwSZts>



Aula 2 - Introdução à programação de computadores - parte1

<https://pt.slideshare.net/henriquecarmona/introduo-programao-de-computadores-parte1>

Aula 2 - Introdução à programação de computadores - parte 2

<https://pt.slideshare.net/henriquecarmona/aula-2-introduo-programao-de-computadores-parte2>

Aula 3 - Algoritmos computacionais - parte 1

<https://pt.slideshare.net/henriquecarmona/aula-3-algoritmos-computacionais-parte1>

Aula 3 - Algoritmos computacionais - parte 2 (exercícios resolvidos)

<https://pt.slideshare.net/henriquecarmona/aula-3-algoritmos-computacionais-parte-2>

RedMonk – site que divulga o ranking anual das linguagens de programação mais utilizadas no mundo: <https://redmonk.com/sogrady/2020/02/28/language-rankings-1-20/>

Tecnoblog – artigo recente sobre as linguagens mais populares: <https://tecnoblog.net/378494/javascript-python-linguagens-programacao-mais-populares/>



Lógica de Programação com VisualG- **Prof. Thiago de Oliveira** (Vários conteúdos para estudo e aprofundamento)

<https://sites.google.com/a/ifmg.edu.br/troliveira/disciplinas/logica-de-programacao>

