

Variáveis ↷



Lógica de Programação

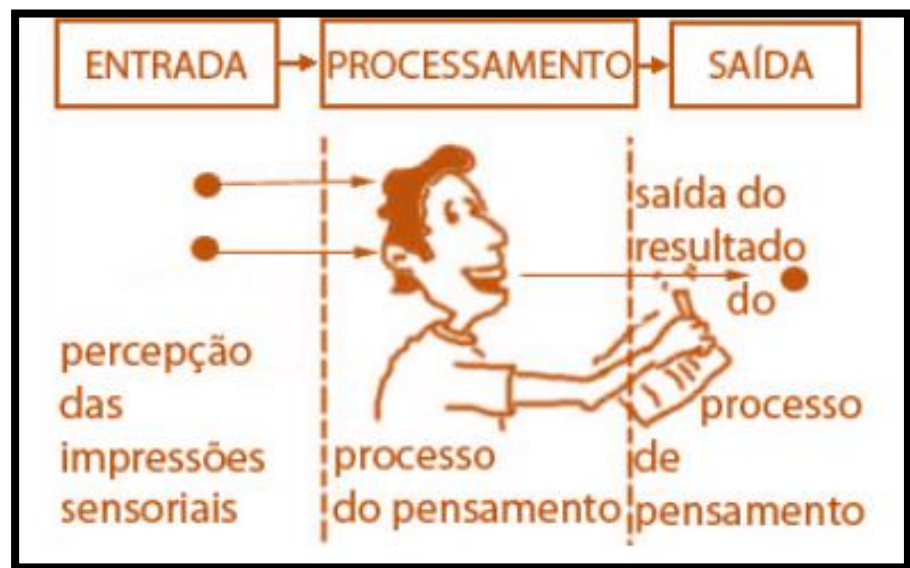
Tipos de dados e Operadores

Revisão

Lógica: sequência de raciocínio que permita a você utilizar de premissas e informações para ser possível obter uma **conclusão**.

Lógica de Programação: técnica de encadear pensamento para atingir um determinado objetivo.

→ **Algoritmo:** sequência lógica, um conjunto de passos a serem executados.



As 3 partes de um algoritmo



Revisão

ALGORITMO = Sequência lógica



Algoritmo <nome_do_algoritmo>

<declaração_de_variáveis>

<subalgoritmos>

Início

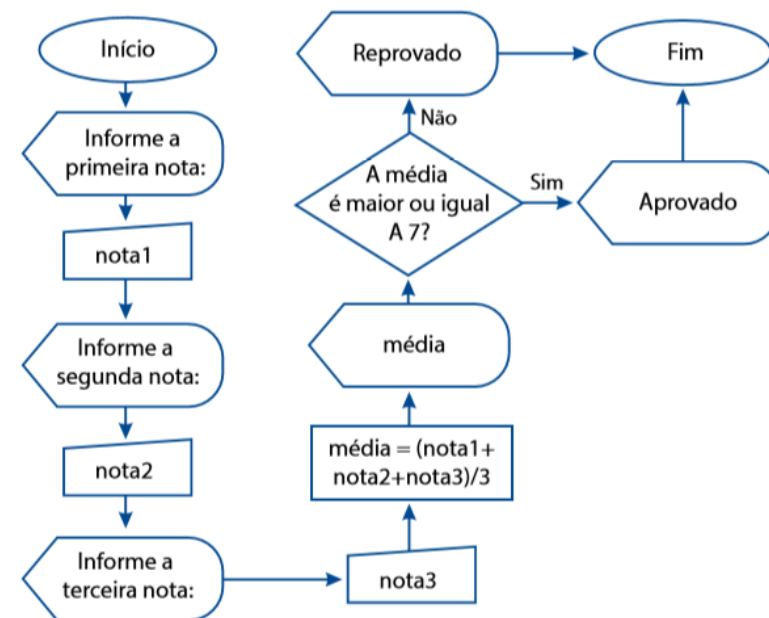
<corpo_do_algoritmo>

Fim.

Algoritmo/Pseudocódigo

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.

Fluxograma



Teste de Mesa

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

Variáveis

são endereços/posições de memória RAM

1-Identificador

2-Valor

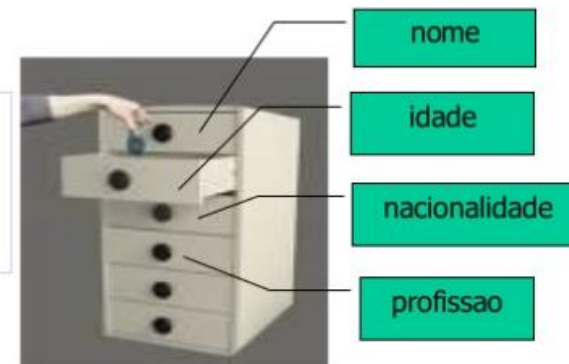
3-Tipo de Dado



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

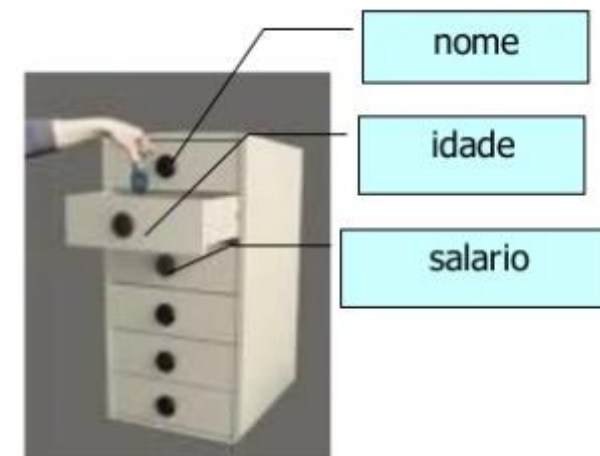
3-Tipo de Dado



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.

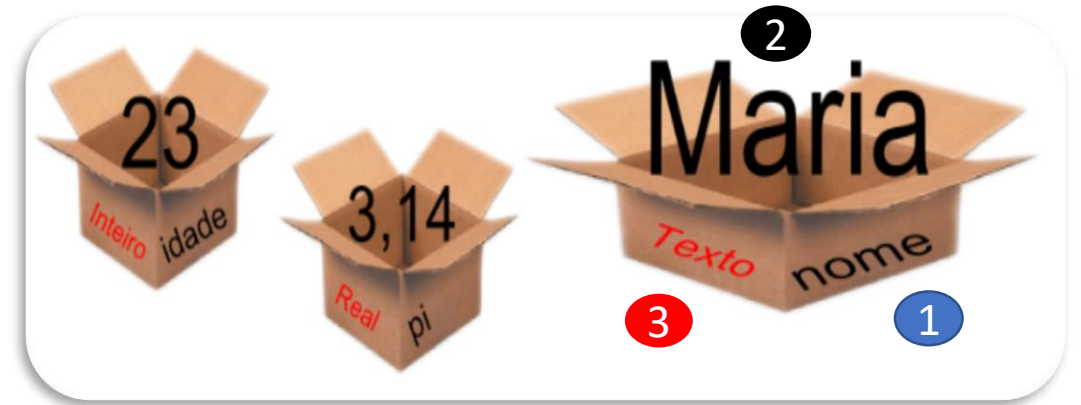
Variáveis

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

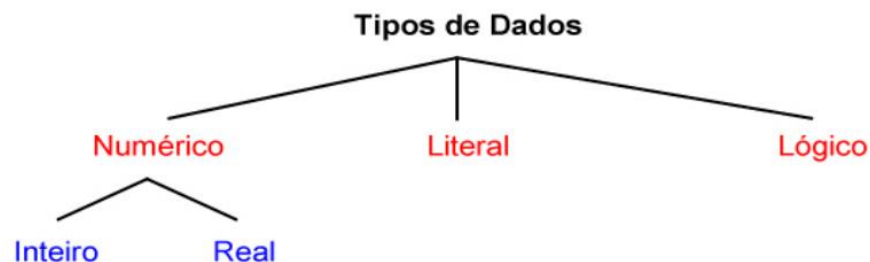
1-Identificador

2-Valor

3-Tipo de Dado



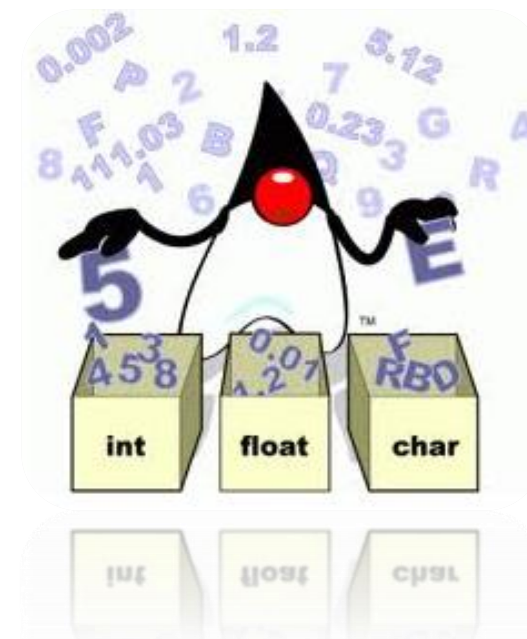
Tipo de Dado *(define que tipos de valores a variável poderá receber)*



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

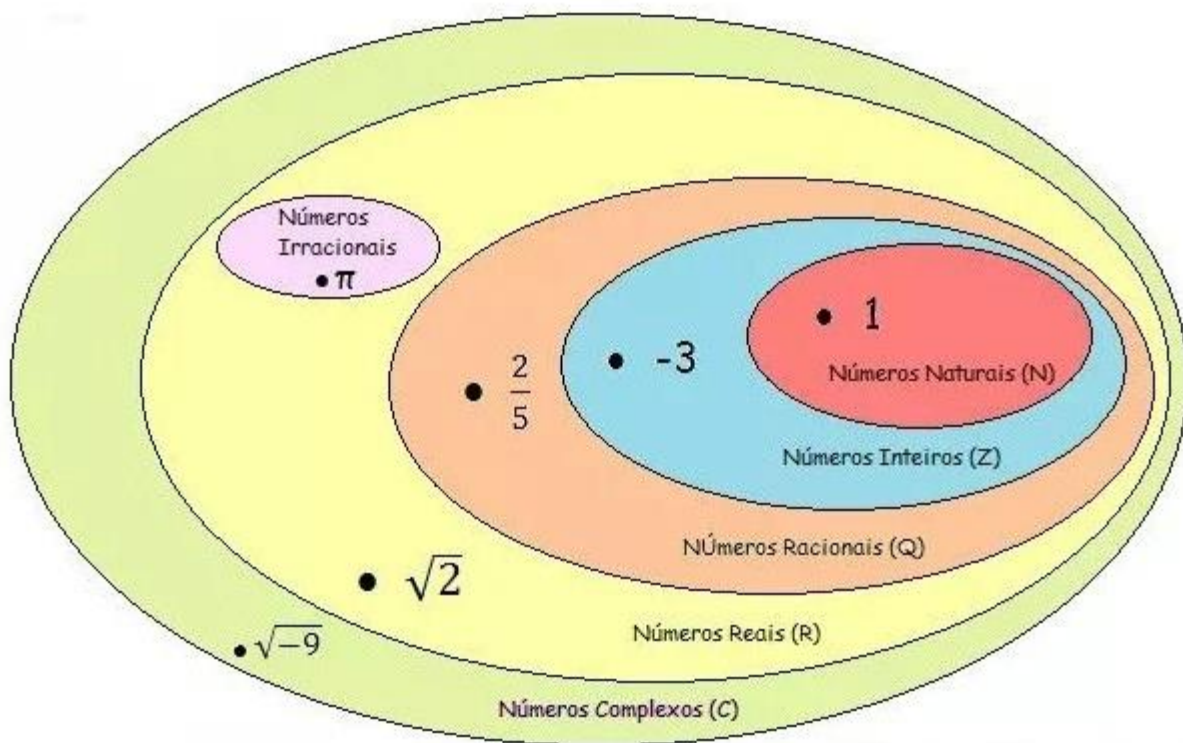


Variáveis

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

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

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

(Reais)

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

(Complexos)

*Raízes Quad. Negativas

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

$$\left(\begin{array}{l} \mathbb{Z}^* = \text{SEM ZERO} \\ \mathbb{Z}_+ = \text{ZERO e POSIT.} \\ \mathbb{Z}_- = \text{ZERO e NEGAT.} \end{array} \right.$$



Grupo Matemático

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

Usado para passar um valor para uma variável.

Simbologia no Pseudocódigo

<-

:=

←

v1 <- 5

v2 <- 7

v1 := 5

v1 := 7

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.

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

Operadores de Atribuição

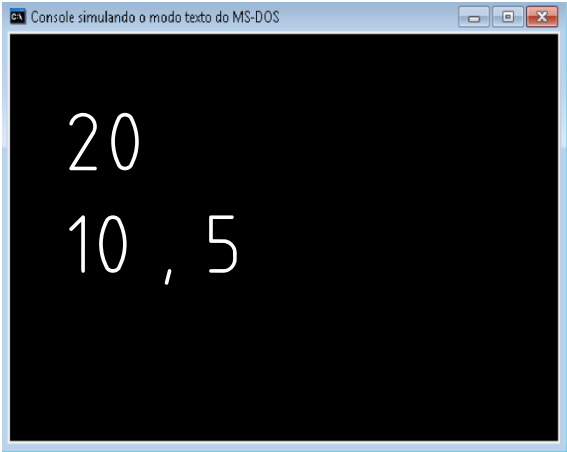
Exercício: Analise os algoritmos feitos em **Diagrama de Chapin** abaixo e diga o que será impresso na tela ao serem executados:

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 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

$35 \div 4 = 9$
 $35 \div 5 = 8,75$

`divisao1 = 36 div 4` = 9
`divisao2 = 35 div 4` = 8

`divisao3 <- 36 / 4` = 9
`divisao4 <- 35 / 4` = 8,75

(Tipo Real)

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

`multiplicacao <- 10 * 4`

`resto <- 35 MOD 4`

35

4

32

3

8

`resto <- 4 MOD 5` = 4
Dica: 4 < 5

`subtracao <- 10 - 3`

Regras de prioridade da matemática:

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

`5* (3+4)+4`
`5*7 + 4`
`35+4`
39

`5*4 / 2 MOD 6`
`20 / 2 MOD 6`
`10 MOD 6`
4

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 Aritméticos

Exercício: Escreva um algoritmo para ler um valor (do teclado) e escrever (na tela) o seu antecessor.

```
ALGORITMO Antecessor1
VAR num, antecessor: INTEIRO;
INICIO
    num <- 0;
    antecessor <- 0;
    ESCREVER "Digite um número : ";
    LER (num);
    antecessor <- num - 1;
    ESCREVER "O seu antecessor é: ", antecessor;
FIM.
```

2 variáveis e 6 linhas

```
ALGORITMO Antecessor2
VAR num: INTEIRO;
INICIO
    num <- 0;
    ESCREVER "Digite um número : ";
    LER (num);
    num <- num - 1;
    ESCREVER "O seu antecessor é: ", num;
FIM.
```

1 variável e 5 linhas

```
ALGORITMO Antecessor3
VAR num: INTEIRO;
INICIO
    num <- 0;
    ESCREVER "Digite um número : ";
    LER (num);
    ESCREVER "O seu antecessor é: ", num - 1;
FIM.
```

1 variável e 4 linhas

Diagrama de Chapin: Antecessor1

num ← 0;
antecessor ← 0
LER num
antecessor ← num - 1
ESCREVER antecessor

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

2 * 4

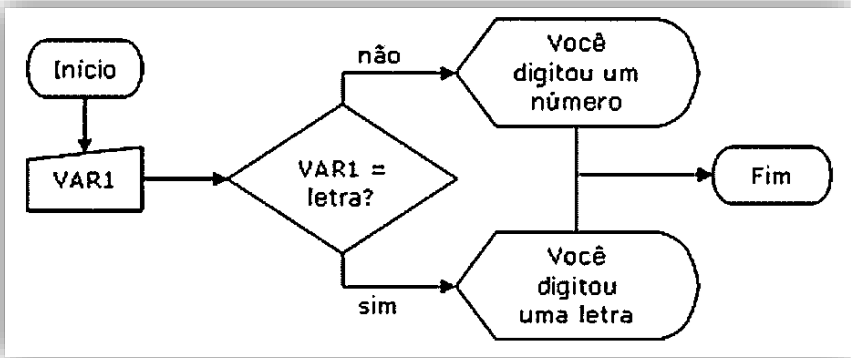
8

==

V

24 / 3

8



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

2 + 8 % 7

2 + 1

3

>=

>=

>=

3 * 6 - 15

18 - 15

3

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)





































 E 

 OU 



 NÃO 

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

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 varável.

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

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

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)
```

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

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.

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			

Estudando Lógica de Programação

Algoritmos **não** 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



Atividade proposta:

Título: Cálculo IR com declaração de variáveis em diferentes linguagens

SA1 - Atividade 4 - Presencial - Linguagens de programação com variáveis tipadas e não tipadas.



Entre as seis linguagens de programação pesquisadas na atividade anterior, em grupo, você e seus colegas deverão selecionar uma linguagem com variáveis tipadas e outra sem variáveis tipadas. Para cada uma dessas linguagens, deverão realizar a implementação de um exemplo com o cálculo solicitado e a declaração de variáveis que representem informações referentes a uma pessoa, utilizando os tipos de dados para otimizar o consumo de memória.

Variáveis que devem ser declaradas:

- Nome;
- Data de Nascimento (Considerar barras);
- Idade;
- Telefone;

- CPF (Considerar pontos e hífen);
- RG (Considerar pontos e hífen);
- Endereço;
- E-mail;

- Telefone;
- Sexo;
- Salário;
- Empregado Ativo (true ou false).

Essas informações devem ser fictícias. Observe que, no caso da linguagem tipada, para cada tipo de informação deve ser indicado o tipo de dado específico. Por exemplo, a informação para a variável idade é do tipo inteiro.

Após a inserção dos dados simule **o cálculo do imposto de renda considerando taxa de 15%.**

Exemplo: `Imposto de Renda <- Salário * 0.15`

Após a conclusão da atividade com as declarações das variáveis solicitadas e a simulação do cálculo do Imposto de Renda, faça alguns testes. **Tente implementar um “Hello World”** da linguagem, faça algumas operações e tente exibi-las.

Atividade proposta:

Título: Cálculo IR com declaração de variáveis em diferentes linguagens

Desenvolvimento de algoritmos com cálculo proposto (IR) e declaração de variáveis solicitadas no enunciado.

RESUMO (Quais são os Requisitos?) :

- Selecionar uma linguagem com variáveis tipadas e outra sem variáveis tipadas. Realizar em ambas linguagens:
 - Declarar as variáveis listadas com os tipos de dados específicos
 - Simule o cálculo do imposto de renda considerando taxa de 15%.
 - Faça alguns testes. Tente implementar um “Hello World” da linguagem, faça algumas operações e tente exibi-las.

Resultados Esperados na Atividade: Que a partir dos conceitos essenciais e apoio do livro didático, o aluno seja capaz de elaborar ao menos **dois algoritmos** com implementação do cálculo proposto e as declarações das variáveis, conforme enunciado e descrição da Situação de Aprendizagem 1.



Dicas para a Atividade

- IDE Online - **Python**: <https://www.programiz.com/python-programming/online-compiler/>
- IDE Online - **Java**: <https://repl.it/languages/java10>
- IDE Online - **JavaScript**: https://www.w3schools.com/js/tryit.asp?filename=tryjs_default

6 compiladores online para estudantes e profissionais de programação
<https://elias.praciano.com/2016/12/6-compiladores-online-para-estudantes-de-programacao/>

Seguem alguns sites oficiais com a documentação das linguagens:

- Python: <https://www.python.org>
- Java: <https://www.oracle.com/br/java/technologies/javase/javase-jdk8-doc-downloads.html>
- Microsoft C++, C e Assembler: <https://docs.microsoft.com/pt-br/cpp/?view=msvc-160>
- PHP: https://www.php.net/manual/pt_BR/index.php
- JavaScript: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>
- Ruby: <https://www.ruby-lang.org/pt/documentation/>
- Lua: <https://www.lua.org/>
- Swift: <https://swift.org/>



Dicas para a Atividade

Calculo imposto de renda a lógica é bem semelhante

ALGORITMO Antecessor1

VAR *num* , *antecessor*: INTEIRO;

INICIO

ESCREVER "Digite um número : ";

LER (*num*);

antecessor **<-** *num* - 1;

ESCREVER "O seu antecessor é: ", *antecessor*;

FIM.

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>

