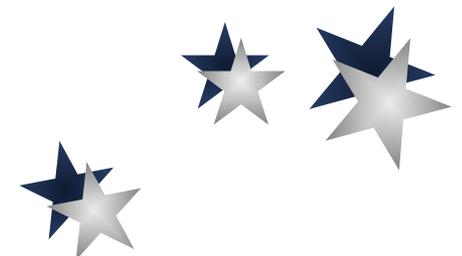


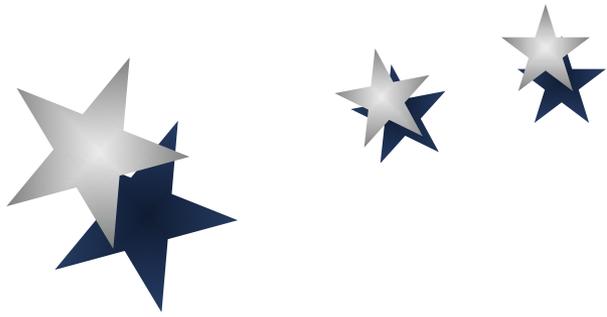
Aula 2:

Dados e Expressões

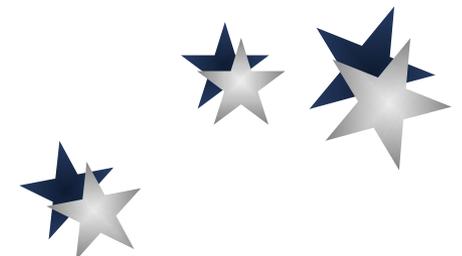
Profa. Dra. Elisa Yumi Nakagawa

1o semestre/2017





DADOS, INSTRUÇÕES E EXPRESSÕES



Uso da Informação

- Computador manipula **informações** contidas em sua **memória**.
- Classificadas em dois tipos:
 - **Instruções**: comandam o funcionamento da máquina e determinam a maneira como os dados devem ser tratados.
 - **Dados**: informação que devem ser manipulada pelo computador.

1- DADOS

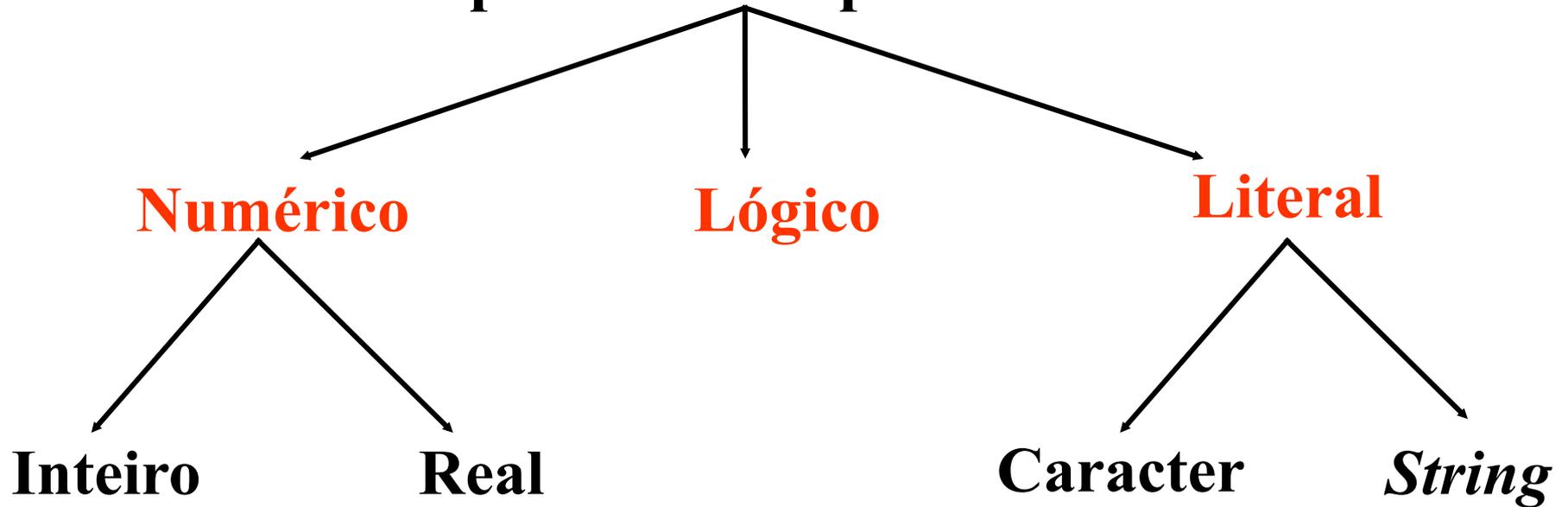
- Objetivo do computador: realizar tarefas envolvendo informações ou DADOS
- Exemplo de DADOS:
 - notas
 - nomes
 - medidas
- Existem vários tipos de dados
 - cada tipo é representado e processado de forma diferente

2- TIPO DE DADOS

- Definir um tipo de dados serve a dois propósitos:
 - Classificar os dados de acordo com as informações contidas neles.
 - indicar quanto espaço de memória deve ser alocado

2- TIPO DE DADOS

Tipos de dados primitivos



2- TIPO DE DADOS - INTEIRO

- Inteiro: Toda e qualquer informação numérica que pertença ao conjunto dos **números inteiros** relativos (negativa, nula ou positiva)
- Exemplos:
 - Ele tem **15** irmãos
 - A escada possui **8** degraus
 - Meu vizinho comprou **2** carros novos
- Linguagem C: tipo **int, long int, unsigned int, ...**

2- TIPO DE DADOS - REAL

- **Real**: Toda e qualquer informação numérica que pertença ao conjunto dos **números reais** (negativa, nula ou positiva)
- Exemplos:
 - Ela tem **1,73** metro de altura (usamos 1.73)
 - Meu saldo bancário é de **215,20** (215.20)
 - No momento estou pesando **82,5** Kg (82.5)

2- TIPO DE DADOS - REAL

- Números reais muito grandes ou muito pequenos são escritos em forma de ponto flutuante (ou notação científica)

- Exemplo:

+3.14E+07 (3.14 x 10⁷)

5.1E-18 (5.1 x 10⁻¹⁸)

- Linguagem C: tipo **float, double, long double**

2- TIPO DE DADOS - Literal

- **Literal**: Toda e qualquer informação composta por um conjunto de caracteres alfanuméricos: **numéricos** (0...9), **alfabéticos** (A...Z, a... Z) e **especiais** (por exemplo, #, ?, !, @).
- Exemplos:
 - Constava na prova: *‘Use somente caneta!’*.
 - O parque municipal estava repleto de placas: *‘Não pise na grama’*.
 - O nome do vencedor é *‘Felisberto Laranjeira.’*
- **Linguagem C**: tipo **char, unsigned char**

2- TIPO DE DADOS - LÓGICO

Lógico: Toda e qualquer informação que pode assumir apenas duas situações (**biestável**)

- Exemplos:

- A porta pode estar ***aberta*** ou ***fechada***.

- A lâmpada pode estar ***apagada*** ou ***acesa***.

- Linguagem C: **não tem tipo lógico!!!**

2- TIPO DE DADOS

Valores típicos para a linguagem C (padrão ANSI)

Tipo	Tamanho (bytes)	Intervalo
Char	1	-127 a 127
Unsigned char	1	0 a 255
Int	2	-32.768 a 32.767
Unsigned int	2	0 a 65.535
Long int	4	-2.147.483.647 a 2.147.483.647
Unsigned long int	4	0 a 4.294.967.295
Float	4	Seis dígitos de precisão
Double	8	Dez dígitos de precisão
Long double	10	Dez dígitos de precisão

3- Variáveis

- Durante a execução do programa os **dados** estão sendo manipulados
- Para que o computador não esqueça das informações **contidas em um dado** é necessário guarda-las em sua **memória**.
- As **variáveis** guardam informações sobre os dados (o seu conteúdo) que estão sendo manipulados.

3- Variáveis

- Armazenamento das variáveis na memória do computador.



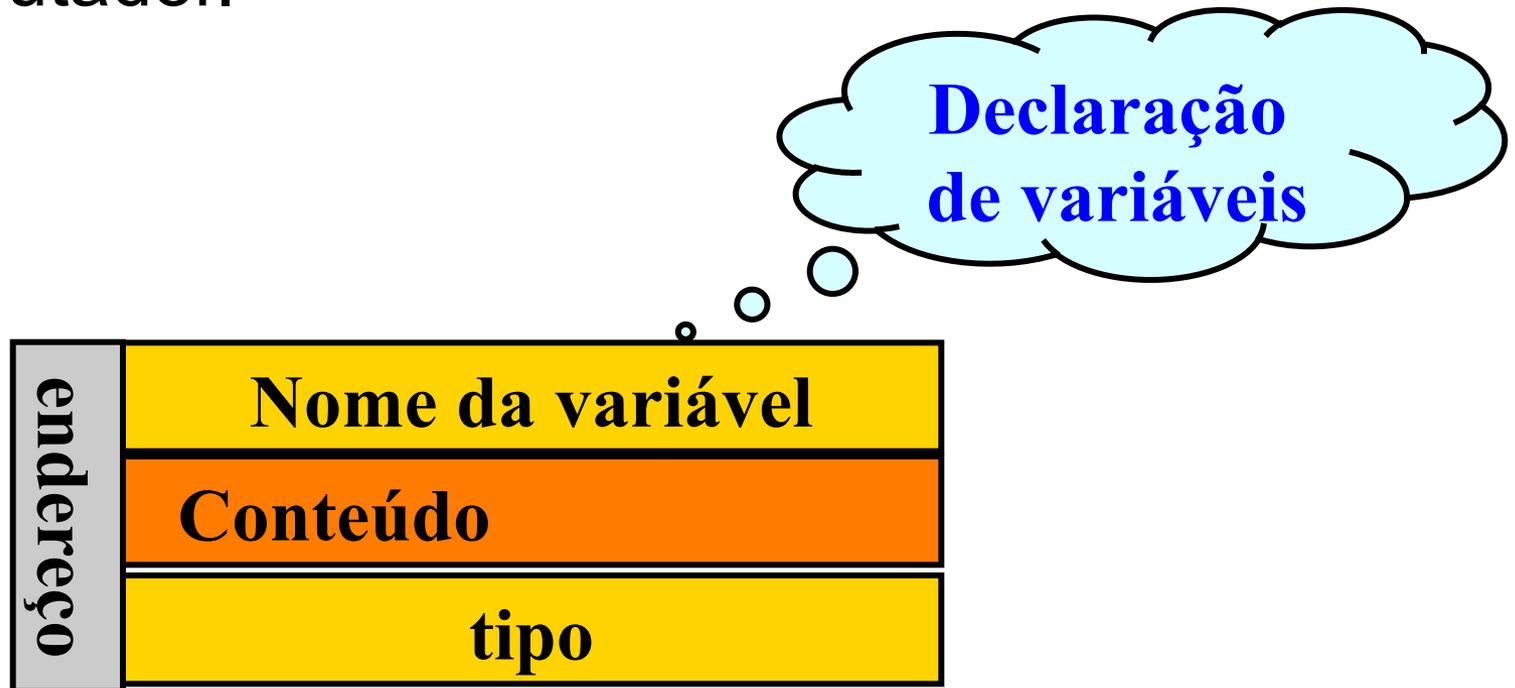
3- Variáveis

- Armazenamento das variáveis na memória do computador.



3- Variáveis

- Armazenamento das variáveis na memória do computador.



3- Declaração de Variáveis

- O **nome da variável** é único em todo o algoritmo.
- O **conteúdo** da variável deve ser do mesmo **tipo** usado na **declaração** da variável.
- O conteúdo da variável é **substituído** por outro que lhe será **atribuído**.
- O uso de uma variável em uma expressão representa o seu conteúdo **naquele momento**.
 - O **uso** não muda o seu conteúdo

3- Declaração de Variáveis

- Forma de declaração:

Algoritmo:

var

inteiro: idade;

real: peso;

caracter: sexo, nome[30];

lógico: tem-filhos;

C:

int idade;

float peso;

char sexo, nome[30];

char tem-filhos;

3- Declaração de Variáveis

- Após a declaração:

44F8	idade	43330	sexo	44322	tem-filhos
	???		???		???
	int		char		char
44F10	peso	44F15	nome		
	???		???		
	float		char[30]		

Ainda não existe valor atribuído para as variáveis

3- Declaração de Variáveis

- Qual o nome que uma variável pode receber?



3- Declaração de Variáveis

- Nome das variáveis:
 - Devem começar por um caracter **alfabético**;
 - Podem ser seguidos por caracteres **alfabético** e **alfanuméricos**;
 - Não é permitido o uso de **caracteres especiais** (menos o **sublinha**);

3- Declaração de Variáveis

- Nome das variáveis:

- Por convenção, os nomes das variáveis utilizam letras **minúsculas** (maiúsculas para nome de **constantes**);
 - Em C existe diferença entre maiúsculo e minúsculo:
 - nome
 - NOME
 - Nome
- Podem ter qualquer tamanho, entretanto, apenas os **31 primeiros caracteres** são utilizados pelo compilador;
- **Não** é permitido o uso de **palavras reservadas** da linguagem C
 - Exemplo: *for, while, do, if, else, nome de funções existentes...*



variáveis diferentes!

3- Declaração de Variáveis

- Nome das variáveis:
 - Utilizar nomes significativos para as variáveis.
 - Exemplo:
 - *Que informação é armazenada na variável **idade**?*
 - *E na variável **endereco**?*
 - *E na variável **x12aa**?*

3- Declaração de variáveis

EXEMPLOS DE NOMES DE VARIÁVEIS:

Inválidos:

5x, e(13), a:b, x-y, nota/2, awq*, p&aa

Válidos:

a, alpha, x, bj152, notas, h_12q

3- Variáveis

- As variáveis representam um dos **tipos de identificadores** que podem ser definidos em um programa (ou algoritmo)
- Um identificador pode identificar:
 - variáveis
 - constantes
 - tipos de dados definidos pelo usuário
 - funções

3- Exemplo

- Defina variáveis para armazenar os seguintes dados sobre uma pessoa:
 - Nome, RG, endereço, telefone, sexo, rendimento, número de dependentes.

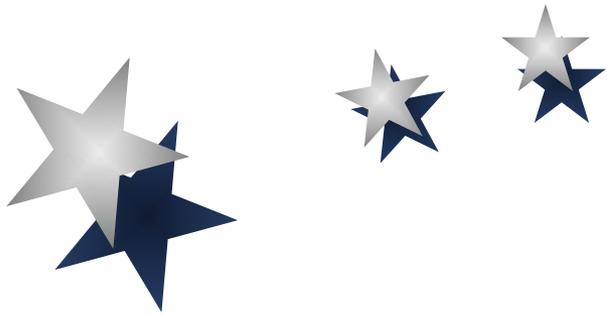
Var

character: nome[30], rg[12], fone[10],

character: sexo;

real: rend;

inteiro: num_dep;



INSTRUÇÕES

**Comandos que determinam ações que
devem ser realizadas**



1- Instruções Primitivas

- Comando de atribuição
- Comando de entrada
- Comando de saída

1- Comando de Atribuição

- Permite que se **atribua um valor** a uma certa variável.
- A **natureza** desse valor deve ser **compatível** com o tipo da variável na qual está sendo armazenado.
- Para se realizar uma atribuição em **C**, utiliza-se o sinal **=**

1- Comando de Atribuição

Linguagem Algorítmica	C
k = 1; TOTAL = $\sqrt{N + x^2}$; cor = “VERDE” ligado = ‘s’;	k = 1; TOTAL = sqrt (N) + pow(X,2); strcpy(cor, “VERDE”) ; ligado = ‘s’;

1- Comando de Atribuição

Linguagem Algorítmica	
<code>k = 1;</code> <code>TOTAL = + ;</code> <code>cor = "VERDE"</code> <code>ligado = 's';</code>	<code>k = 1;</code> <code>TOTAL = sqrt (N) + pow(X,2);</code> <code>strcpy(cor, "VERDE");</code> <code>ligado = 's';</code>

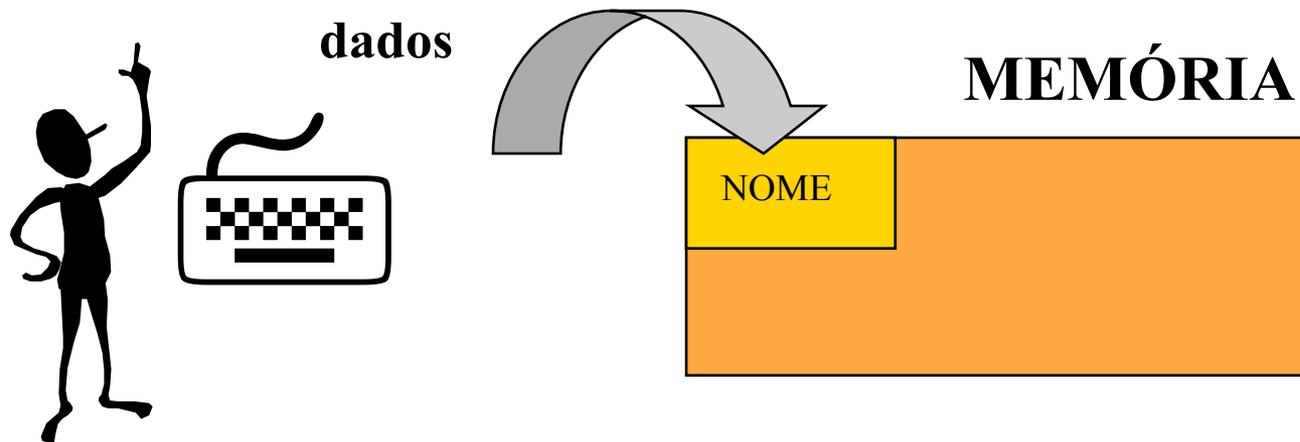
constante numérica

constante literal

expressão aritmética

2- Comando de Entrada

- O comando de entrada é utilizado para receber dados digitados pelo usuário (**DADOS DE ENTRADA**) e armazená-los em variáveis.
- Os dados de entrada são fornecidos ao sistema através de uma **unidade de entrada**, por exemplo o **teclado**.



2- Comando de Entrada

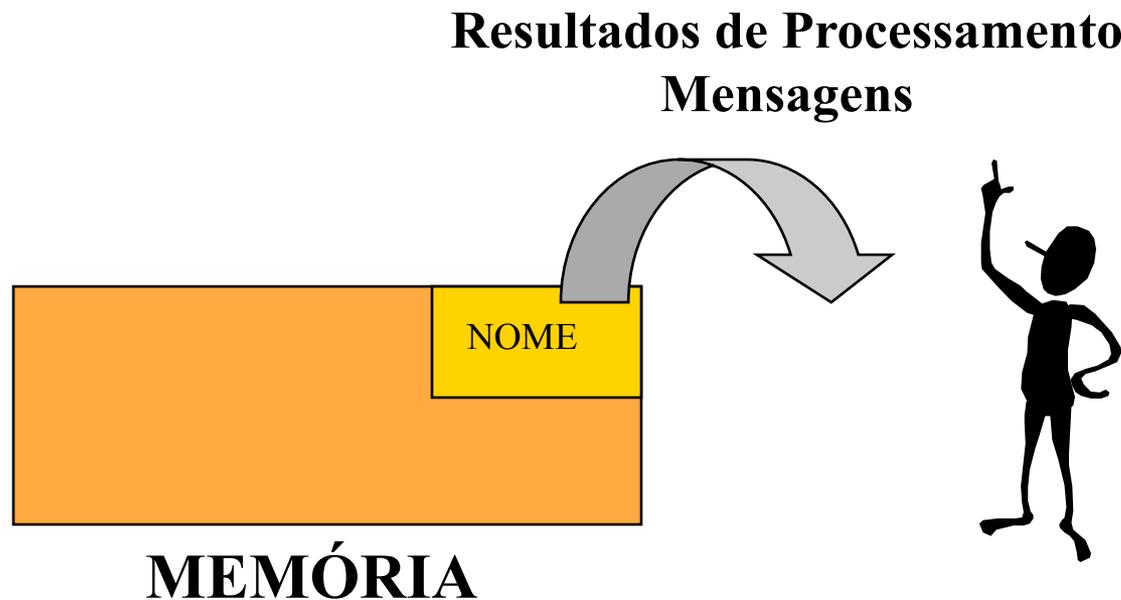
Algoritmo	C
leia(nome); leia(num); leia(salario); leia(sexo); leia(num1, num2);	gets(nome); scanf("%d", &num); scanf("%f", &salario); sexo = getch(); scanf("%d %d", &num1, &num2);

2- Comando de Entrada

- Na execução de um comando de entrada, o processamento é **interrompido**, até que sejam fornecidos, via unidade de entrada, **valores** para os dados de entrada.
- Os valores digitados pelo teclado devem ser **separados** pela digitação da tecla **<ENTER>**

3- Comandos de Saída

- O comando de saída é utilizado para que o sistema forneça, numa **unidade de saída**, os **resultados** do processamento e **mensagens**.

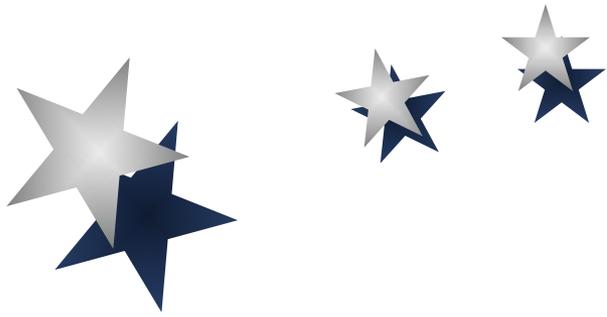


3- Comandos de Saída

- **Resultados de Processamento:** podem ser fornecidos através de conteúdos de **variáveis**, conteúdos de **constantes** e resultados de **expressões aritméticas e lógicas**.
- **Mensagens:** são utilizadas para que o programa dê **informações** ao usuário a respeito do **processamento** que está se realizando. Podem ser fornecidas através de conteúdo de **variáveis**, ou constantes do tipo **string** ou da **mensagem** propriamente dita.

3- Comandos de Saída

Algoritmo	C
escreva(nome); escreva(nome, num); escreva("valor inválido!");	printf("%s", nome); printf("%s %d", nome, num); printf("valor inválido!");



EXPRESSÃO ARITMÉTICA



Expressão Aritmética

- Denomina-se **expressão aritmética** aquela cujos operadores são aritméticos e cujos operandos são constantes ou variáveis do tipo numérico (inteiro ou real).
- Exemplo:

Expressão Aritmética

OPERADORES NUMÉRICOS

+	adição
-	subtração
*	multiplicação
/	divisão
/	quociente inteiro (para int)
%	resto da divisão

Expressão Aritmética

- Os operadores $+$, $-$, $*$, $/$ atuam com operandos do tipo **integer** ou **real**:
 - se pelo menos um dos operandos for do tipo **real**:
 - Resultado do tipo **real**
 - se todos os operandos são do tipo **integer**:
 - Resultado do tipo **integer**

Expressão Aritmética

- Operador `/`:
 - Se variável for integer, então é considerada somente a parte inteira da divisão
 - Se variável for float, então é considerada a parte decimal também.
- Operador `%`: resto da divisão
 - Só se aplica para operandos integer
- Exemplo:

30 / 7 = resulta 4

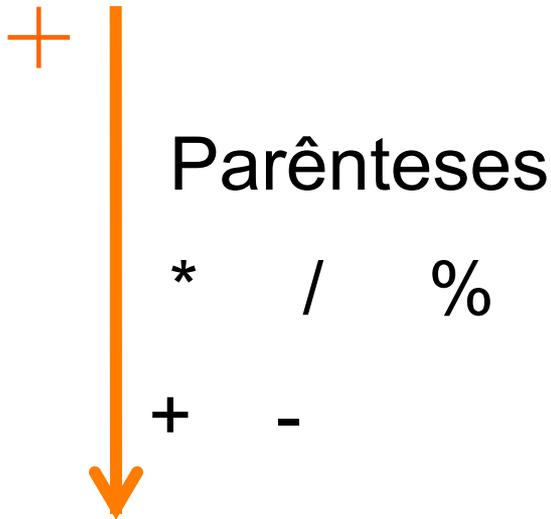
30 % 7 = resulta 2

5 / 7 = resulta 0

5 % 7 = resulta 5

Expressão Aritmética

- Na resolução das expressões aritméticas, as operações guardam uma hierarquia entre si, sendo resolvidos da esquerda para direita se são de igual prioridade:



Expressão Aritmética

- Exemplo:

$$5^2 - \frac{4}{2} + \sqrt{(1 + 3 * 5)} / 2$$

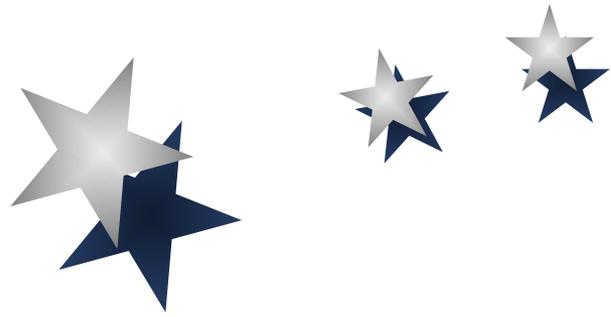
$$5^2 - \frac{4}{2} + \sqrt{(1 + 15)} / 2$$

$$5^2 - \frac{4}{2} + \sqrt{16} / 2$$

$$25 - \frac{4}{2} + 4 / 2$$

$$25 - 2 + 4 / 2$$

$$25 - 2 + 2$$



EXPRESSÃO LÓGICA



Expressão Lógica

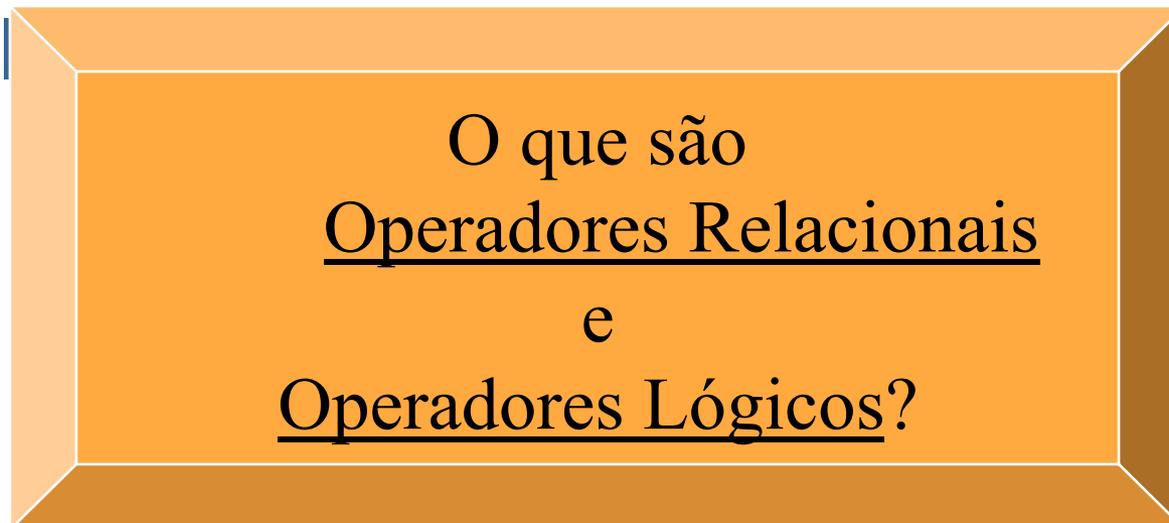
- Denomina-se **expressão lógica** aquela cujos operadores são lógicos ou relacionais e cujos operandos são relações ou variáveis ou constantes do tipo lógico.
- Exemplo (em C):

$(A+B == 0) \text{ and } (C != 1)$

Expressão Lógica

- Denomina-se **expressão lógica** aquela cujos operadores são lógicos ou relacionais e cujos operandos são relações ou variáveis ou constantes do tipo lógico.

- Exemplos



OPERADORES RELACIONAIS

- Utilizam-se os **operadores relacionais** para realizar comparações entre dois valores de mesmo tipo primitivo.
- Tais valores são representados por constantes, variáveis ou expressões aritméticas
- O resultado obtido de uma relação é sempre um **valor lógico**.

OPERADORES RELACIONAIS

OPERADORES	==	igual
	!=	diferente
	>	maior
	<	menor
	>=	maior ou igual
	<=	menor ou igual

OPERADORES RELACIONAIS

- Exemplos:

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

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

OPERADORES LÓGICOS

- Utilizam-se três **conectivos** básicos para a formação de novas proposições lógicas compostas a partir de outras proposições lógicas simples.

OPERADORES LÓGICOS

&&

e

||

ou

!

não

TABELAS VERDADE

- **Tabelas Verdade** é o conjunto de todas as possibilidades combinatórias entre os valores de diversas variáveis lógicas, as quais se encontram em apenas duas situações (V ou F), e um conjunto de operadores lógicos

TABELAS VERDADE

Operação de Negação

A	!A
true	false
false	true

Expressão Lógica

TABELAS VERDADE

Operação de Conjunção (e)

A	B	A && B
true	true	true
true	false	false
false	true	false
false	false	false

Exemplo:

A = Tenho dinheiro?

B = Tenho onde ir?

Expressão Lógica

TABELAS VERDADE

Operação de Disjunção Não-Exclusiva (ou)

A	B	A B
true	true	true
true	false	true
false	true	true
false	false	false

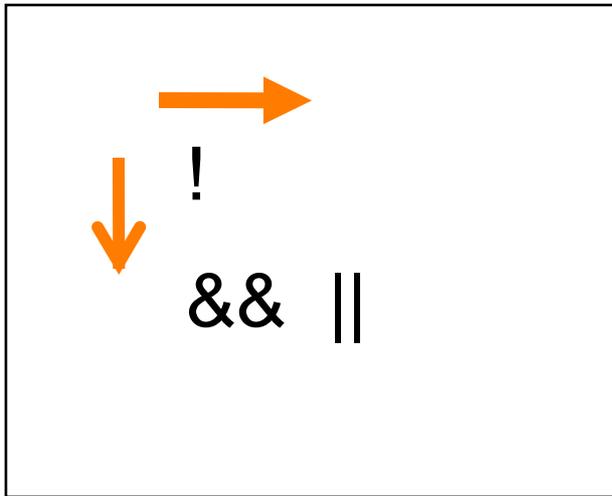
Exemplo:

A = Tenho dinheiro?

B = Tenho onde ir?

OPERADORES LÓGICOS

- Na resolução das expressões lógicas, os operadores lógicos guardam uma hierarquia entre si:



OPERADORES LÓGICOS

- Na resolução das expressões lógicas, os diversos operadores guardam uma hierarquia entre si:



parenteses mais internos

operadores aritméticos

operadores relacionais

operadores lógicos

Expressão Lógica

- Exemplos:

! (5 != 10 / 2 || V && 2 - 5 > 5 - 2 || V)

! (5 != 5 || V && -3 > 3 || V)

! (F || V && F || V)

! (F || F || V)

! (F || V)

! (V)

F

Expressão Lógica

- Exemplos:

$2^4 \neq 4 + 2 \parallel 2 + 3 * 5 / 3 \% 5 > 0$

$16 \neq 6 \parallel 2 + 15 / 3 \% 5 > 0$

$16 \neq 6 \parallel 2 + 5 \% 5 > 0$

$16 \neq 6 \parallel 2 + 0 > 0$

$16 \neq 6 \parallel 2 > 0$

$\vee \parallel \vee$

\vee

Primeiro Programa em C

```
#include <stdio.h>
#include <stdlib.h>

void main ()
{
    // Declarações de todas as variáveis utilizadas
    int n1, n2;

    // Comandos
    printf ("Entre com dois números inteiros: \n");
    scanf ("%d", &n1);
    scanf ("%d", &n2);
    printf ("A soma de %d e %d é %d.", n1, n2, n1+n2);

    return;
}
```

Exercícios

1. Escreva um algoritmo que calcula o preço total de um produto, tendo como entrada o preço unitário e a quantidade vendida.
2. Faça um algoritmo para cálculo do quadrado de um número, ou seja, o produto de um número por si mesmo.
3. Faça um algoritmo para o cálculo da função $2x + 3y^2$ num domínio real.