

# Lógica de Programação e Algoritmos em C

## Variáveis

Um programa de computador poderá manipular informações, como o nome de um funcionário, o salário que recebe todos os meses, a sua idade, etc.

Estas informações têm um tipo de dado, por exemplo: o nome do funcionário é uma sequência de caracteres, também chamada de “**string**” ou “**cadeia de caracteres**”. O salário é um número real, a idade é um número inteiro.

Para armazenar estes dados, usaremos áreas de memória do computador, que chamamos de “**variáveis**”.

Uma variável é um espaço de memória reservado para armazenar certo tipo de dado com um nome para fazer referência ao seu conteúdo.

## Regras de nomes das variáveis

Para dar nomes para as variáveis seguimos regras. As regras utilizadas são:

Regra	Significado
1	Os nomes das variáveis são construídos com letras e números
2	O primeiro caractere deve ser uma letra
3	Não poderá ter espaços em branco nos nomes das variáveis
4	Não poderá começar com um número
5	Letras maiúsculas e minúsculas são diferentes
6	Os nomes devem ter um significado relacionado com o conteúdo da variável
7	Nomes compostos devem começar com letras minúsculas e o início do próximo nome começa com letra maiúscula

## Tipos de dados das variáveis

Cada variável deve ter um tipo de dado especificado na declaração da variável.

tipo de dado	Significado	Escala
char	Caracter	-128 a 127
int	Inteiro	-2.147.483.648 a 2.147.483.647
short	Inteiro curto	-32.765 a 32.767
long	Inteiro longo	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807
float	Ponto flutuante de precisão simples	3.4E +/- 38 (7 dígitos)
Double	Ponto flutuante de precisão dupla	1.7E +/- 308 (15 dígitos)
unsigned int	Inteiro sem sinal	0 a 4.294.967.295
unsigned short	Inteiro curto sem sinal	0 a 65.535
unsigned long	Inteiro longo sem sinal	0 a 18.446.744.073.709.551.615

## Operadores aritméticos

Para os cálculos matemáticos utilizamos os operadores aritméticos.

Operador	Ação	Exemplo	Resultado
-	Subtração	5 - 1	4
+	Adição	5 + 1	6
*	Multiplicação	5 * 2	10
/	Divisão	10 / 2	5
%	Resto da divisão	11 % 4	3

## Entrada e saída de dados

Em nossa escola do ensino fundamental a nossa professora de matemática nos dava problemas de para resolver. Tínhamos que entender o que deveria ser calculado, saber os números envolvidos no cálculo e quais os resultados ela queria.

Se ela pedisse para calcular o volume de um paralelepípedo retângulo, tínhamos que saber a fórmula: “*Volume = comprimento x largura x altura*”, depois saber os valores para o comprimento, para a largura e altura, que deveriam ser fornecidos. O resultado teríamos que mostrar ou dizer que era o volume do sólido calculado.

Nosso computador poderia fazer este cálculo, pois ele é uma máquina capaz de aceitar informações, processá-las eletronicamente e fornecer os resultados desejados.

Se fossemos resolver o problema acima, através de um programa de computador, os valores do comprimento, largura e altura, provavelmente seriam fornecidos através do teclado. O programa faria o cálculo na sua memória, e o resultado, que é o volume, seria apresentado na tela.

O programa teria que “**LER**” do teclado, processar os dados, e “**ESCREVER**” os resultados na tela do computador.

### Exemplo de entrada e saída de dados em linguagem C

A função de biblioteca que escreve dados na tela: **"printf"**.

A função de biblioteca que vai ler dados do teclado fazendo a entrada de dados é a função: **"scanf"**.

Esta é a função definida no padrão ANSI da Linguagem C.

Veja o exemplo a seguir:

```
main()
{
    int valorA, valorB, soma;                /*declarando variáveis */

    printf("Digite um valor: ");              /* lendo o valor de A */
    scanf("%d", &valorA);
    printf("Digite outro valor: ");           /* lendo o valor de B */
    scanf("%d", &valorB);

    soma = valorA + valorB;                   /* calculando a soma */

    printf("O resultado da soma: %d \n", soma); /* apresentando o resultado */
}
```

Neste programa estamos declarando três variáveis inteiras.

Pedimos para o usuário digitar um valor na função: **"printf"**, e estamos lendo o que for digitado no teclado com a função: **"scanf"**:

Na função de leitura o tipo de dado que será lido é: **"%d"** e para guarda na variável, após a vírgula estamos passando o endereço da variável: **"valorA"**. O endereço é identificado pelo sinal **"&"** junto do nome da variável.

Se você esquecer de colocar o **"&"** junto da variável, o valor lido não será armazenado na variável.

Depois vamos ler um valor para a variável: **"valorB"**.

Terminada a entrada de dados, fazemos o processamento de dados, calculando a soma dos valores lidos.

E finalmente vamos mostrar os dados processados.

### Lendo dados do tipo: string

O nome de uma pessoa deve ser armazenado em uma variável que tenha espaço para uma sequência de caracteres, que geralmente é chamada de “**string**”. A variável tem como tipo básico o “**char**”. Mas, para armazenar várias letras e espaços, precisamos definir o tamanho máximo desta variável.

```
char nome[80];
```

Entre os “**colchetes**” estamos definindo que a variável: “**nome**” terá espaço para armazenar até 79 caracteres, porque o último é reservado para o caractere “**\0**” (**null**).

O caractere “**\0**” marca o final da string. Se digitarmos uma quantidade menor de caracteres ele encerra a string.

Na string a seguir: “**João Josué da Costa**”, cada caractere vai ocupar um byte na memória e o último caractere será o “**\0**”. Veja uma representação desta string no computador:

J	o	ã	o		J	o	s	u	é		d	a		C	o	s	t	a	\0
---	---	---	---	--	---	---	---	---	---	--	---	---	--	---	---	---	---	---	----

A função de biblioteca utilizada para ler um nome de uma pessoa é a função: “**gets()**”.