# ALGORITMO

*Lógica de Programação*

Não é a linguagem de programação que define o programador, mas sim sua lógica.

David Ribeiro Guilherme
(https://www.pensador.com/frases_de_programador/)

Proibido a venda e a distribuição deste produto

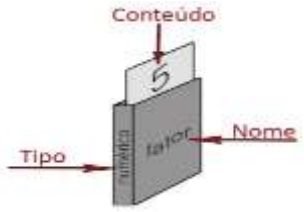
## Variáveis

Toda vez que criamos um programa / algoritmo estamos planejando quais instruções o computador vai executar.

Podemos entender a programação como um planejamento.

Muitas vezes, ao iniciar um programa é necessário declarar as variáveis que serão utilizadas neste algoritmo.

Quando declaramos uma variável estamos reservando um local na Memória Principal do Computador, isto é, um endereço para armazenar o conteúdo de uma variável.




## Palavras reservadas

Não devem ser usadas como nome de variáveis.

| | | | |
|---|---|---|---|
| abstract | as | base | bool |
| break | by | byte | case |
| catch | char | checked | class |
| const | continue | decimal | default |
| delegate | do | double | descending |
| explicit | event | extern | else |
| enum | false | finally | fixed |
| float | for | foreach | from |
| goto | group | if | implicit |
| in | int | interface | internal |
| into | is | lock | long |
| new | null | namespace | object |
| operator | out | override | orderby |
| params | private | protected | public |
| readonly | ref | return | switch |
| struct | sbyte | sealed | short |
| sizeof | stackalloc | static | string |
| select | this | throw | true |

Proibido a venda e a distribuição deste produto

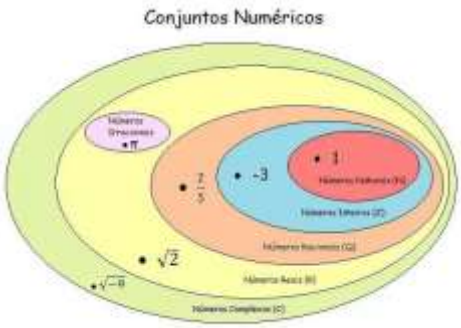
| | | | |
|---|---|---|---|
| try | typeof | uint | ulong |
| unchecked | unsafe | ushort | using |
| var | virtual | volatile | void |
| while | where | yield | |

Caso seja realmente necessário chamar uma variável com o nome de uma palavra reservada para, por exemplo, garantir clareza ao código, você pode utilizar o símbolo @ (arroba) na frente do nome da variável.

Deste modo o compilador vai entender que, apesar de você estar utilizando uma palavra reservada, você na verdade quer nomear uma variável. Exemplos de declarações: int @int, double @double .

## Tipos de Variáveis

Observe a imagem abaixo:




Os tipos de dados tem uma organização parecida com a dos conjuntos numéricos.

- bool
- byte
- char
- decimal
- double
- float
- int
- long
- Object
- sbyte
- short
- string
- uint
- ulong
- ushort

## Declaração de variável

Evite usar underline no meio do nome;

Não crie variáveis que se diferenciem apenas por sua forma.

Por exemplo: **seuNome** e outra chamada **SeuNome**.

O nome da variável deve representar sua função. Por exemplo: **idade** representa a idade de uma pessoa ou alguma coisa, **cont** representa um contador.

Inicie o nome da variável com uma letra minúscula;

Não utilize todas as letras maiúsculas.

Quando o nome tiver mais que uma palavra, a primeira letra de cada palavra após a primeira deve ser maiúscula (notação **Camel Case**);

Não utilize acentos ou caracteres especiais para declarar os nomes das variáveis.

Variáveis não podem ter nomes iguais, C# é case sensitive.

- $\text{Nome} \neq \textit{nome}$

**Sintaxe:**

```
tipo nome ;
int numero;
int numero = 10;
int numero1 = 7, numero2 = 5, numero3;
```

## Atribuindo valores às variáveis

Depois de declarado a variável é necessário atribuir-lhe um valor.

Em C# você não podemos usar uma variável antes de dar um valor para ela, o que poderia gerar um erro de compilação.

```
int idade;
idade = 23;
```

Também poderia ser feito da seguinte maneira.

```
int idade = 23;
```

## Entrada de dados

**Leitura do teclado:**

Faz a leitura do teclado, ou seja, você poderá interagir com o programa.

Comando '**Console.ReadLine( )**

**Exemplo**

```
Console.WriteLine("Digite um nome!");
string nome = Console.ReadLine();
Console.WriteLine("O nome digitado foi: " + nome);
```

Quando o valor digitado for atribuído a uma variável com tipagem diferente de uma string será necessário fazer a conversão do tipo.

**Exemplo**

Proibido a venda e a dis... produto

```
Console.WriteLine("Digite um valor  -- String");
string valorString = Console.ReadLine();
Console.WriteLine("O valor digitado foi: " + valorString);

Console.WriteLine("Digite um valor  -- Int");
int valorInt = int.Parse(Console.ReadLine());
Console.WriteLine("O valor digitado foi: " + valorInt);

Console.WriteLine("Digite um valor  -- Double");
double valorDouble = double.Parse(Console.ReadLine());
Console.WriteLine("O valor digitado foi: " + valorDouble.ToString("F2"));

Console.WriteLine("Digite um valor  -- Char");
char valorChar = char.Parse(Console.ReadLine());
Console.WriteLine("O valor digitado foi: " + valorChar);
```

## Operador de atribuição " = "

**Exemplo:**

```
int a = 10, b;
b = a;
Console.WriteLine(a + " - " + b);

int x, y, z;
x = y = z = 110;
Console.WriteLine(x + " - " + y + " - " + z);
```

## Operadores Aritméticos

| | |
|---|---|
| + | soma |
| - | subtração |
| * | multiplicação |
| / | divisão |
| % | módulo |

**OBSERVAÇÃO**:

Proibido a venda e a distribuição deste produto

Se a divisão for feita entre dois números inteiros, o resultado será um número inteiro, ou seja, pode se perder o valor das casas decimais.

O operador % (módulo) só funciona com números inteiros.

## Lista de exercício 01

01 – Elaborar um algoritmo que imprima a frase abaixo:
"Aprendendo Algoritmo"

Aula gravada: https://youtu.be/SDyOU1HQxWo

02 – Elabore um algoritmo que imprima a frase da maneira descrita abaixo, uma frase abaixo da outra:

Aprendendo Algoritmo
e Fazendo muito Exercício
Primeiro fazendo exercício em 'C#'

Aula gravada: https://youtu.be/xdBoUQ-USD8

03 – Crie um algoritmo que leia dois nomes e imprima os nomes na sequência em que foram escritos e posteriormente os imprima na ordem inversa.

Aula gravada (exercício similar): https://youtu.be/xM_DJ8KCdzc

04 – Crie um algoritmo que receba duas variáveis do tipo inteiro.
Atribuir um valor para cada variável e posteriormente exibir este valor.
Para finalizar, exibir a primeira variável acrescida de uma unidade e exibir a segunda variável decrescida de uma unidade.

05 – Cria um algoritmo que receba 5 números do tipo double e exiba a soma com a seguinte frase:

"Os números digitados foram ..., ..., ..., ... e sua soma é ... .

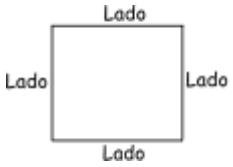
06 – Ler dois números e exibir as seguintes mensagens:
O números digitados foram ... e .... .
A soma dos números ... e ... é ... .
A subtração dos números ... e ... é ... .
A multiplicação dos números ... e ... é ... .
A divisão dos números ... e ... é ... .
A média dos números ... e ... é ... .

07 – Ler um número inteiro e imprimir seu antecessor e seu sucessor.
SAÍDA:
O número digitado foi ..., seu antecessor é ... e seu sucessor é ...

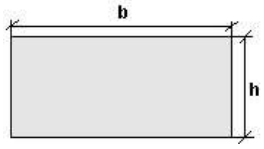
08 – Elabore um algoritmo que calcule a área e o perímetro de um quadrado.

Proibido a venda e distribuição deste produto

Área = lado2
Perímetro = é a soma de todos os lados




09 – Elabore um algoritmo que calcule a área e o perímetro de um retângulo.
Área = b x h
Perímetro = é a soma de todos os lados




## Constantes

São como variáveis em que uma vez definido o valor, este não pode ser mudado pelo programa. Exemplo:

```
const double PI = 3.1415;
```

Antes do tipo da variável deve ser informado a palavra-chave const.

Por convenção e para diferenciar uma variável normal de uma constante, ela deve ser escrita com letra maiúscula.

Outro ponto é que uma constante deve sempre ser inicializada na declaração, caso contrário, será gerado um erro.

## Tipagem implícita

Observação: Este tipo de tipagem diminui o controle do programando sobre a aplicação.

Não é necessário informar o tipo da variável (a partir da versão 3.0).

Observação: É necessário sempre informar o valor da variável no momento que é declarada caso contrário teremos um erro.

```
var n = 10;
Console.WriteLine(n);
Console.WriteLine(n.GetType());
```

## Tipagem implícita fraca

```
dynamic @var = 10.55;
Console.WriteLine(@var);
Console.WriteLine(@var.GetType());

@var = "Fulano da Silva";
Console.WriteLine(@var);
Console.WriteLine(@var.GetType());

@var = 5;
Console.WriteLine(@var);
Console.WriteLine(@var.GetType());

@var = true;
Console.WriteLine(@var);
Console.WriteLine(@var.GetType());

@var = 'c';
Console.WriteLine(@var);
Console.WriteLine(@var.GetType());
```

Proibida venda

## Introdução às Strings - Interpolação

Uma string é na verdade uma sequência de caracteres (uma matriz de caracteres).

Deve ser declarada usando aspas duplas.

```
string str = "Um exemplo de string";
Console.WriteLine(str);
```

Você pode concatenar sequências, como este:

```
string str1 = "Um exemplo ";
string str2 = "de string! ";
string str3 = "Concatenação de strings!!!";
```

```
Console.WriteLine(str1 + str2 + str3);
Console.WriteLine(str1 + " - " + str2 + " - " + str3);
Console.WriteLine("{0} - {1} - {2}", str1, str2, str3);
Console.WriteLine($"{str1} - {str2} - {str3}");
```

Strings são imutáveis, significando que elas não podem ser alteradas uma vez que tenha sido criada. Métodos que atuam em strings realmente retornam novas strings.

## Caracteres de escape

- Quebra de linha – “\n”
- Tabulação – “\t”
- Insere uma barra – “\\”
- Insere um aspas – “\”” ou “\’”

## Exemplo:

```
Console.WriteLine("Quebra\nDe\nLinha");
Console.WriteLine("Quebra\nDe\nLinha\n\tcom\n\ttabulação");
Console.WriteLine("Exemplo \\ de caracter de escape");
Console.WriteLine("Exemplo \" de caracter de escape\"");
Console.WriteLine("Exemplo \' de caracter de escape\'");
```

## Operadores reduzidos

+=, -=, *=, /= e %=

## Incremento e decremento

Pré incremento

```
++contador;
```

Pré decremento

```
--contador;
```

Pós incremento

```
contador++;
```

pós decremento

```
contador--;
```

proibido a venda e a distribuição deste produto

A diferença é que com **++** o cálculo é feito mais rápido que os demais, sendo muito utilizado nos laços (tratado mais adiante).

## Funções Matemáticas Básicas

**Sugestão:**

- Utilize double para não perder o valor das casas decimais.

### Potência:

```
double numero = 5.0;
Console.WriteLine( Math.Pow(numero,2) );
Console.WriteLine( Math.Pow(numero,3) );
```

### Raiz quadrada:

```
double numero = 25.0;
Console.WriteLine( Math.Sqrt(numero) );

numero = 81.0;
Console.WriteLine( Math.Sqrt(numero) );
```

### Raiz cúbica:

```
double numero = 25.0;
Console.WriteLine( Math.Cbrt(numero) );

numero = 81.0;
Console.WriteLine( Math.Cbrt(numero) );
```

### Constante PI:

```
Console.WriteLine(  Math.PI  );
```

### Orientação...

Para qualquer exercício siga os seguintes passos

a) Primeiro leia o exercício por inteiro.
b) Entenda o exercício, de nada adianta criar um código se você não sabe o que é preciso fazer.
c) Defina quais serão as entradas.
d) Defina como serão as saídas.
e) Execute o código

## Lista de exercício 02

10- Crie um Algoritmo que receba um número e imprima sua raiz quadrada.

SAÍDA:
A raiz quadrada de no número .... é ...

11 – Crie um Algoritmo que receba um número e imprima seu valor elevado a 2, elevado a 3, elevado a 4 e elevado a 5.

SAÍDA:
O número digitado foi ... E seu valor elevado a 2 é ..., elevado a 3 é ..., ...

12 – Crie um Algoritmo que receba um número e imprima sua raiz quadrada e sua raiz cúbica.

SAÍDA:
O número digitado foi ...
Sua raiz cúbica é ...
Sua raiz quadrada é ...

13 – Entrar com quatro números e imprimir a média ponderada, sabendo-se que os pesos são respectivamente 1, 2, 3 e 4.

14- Elabore um algoritmo que entre com valor do raio e calcule a área e o perímetro do círculo correspondente.

- A fórmula para se calcular a área da circunferência é : $\mathrm{A} = \pi * \mathrm{raio}^2$
- A fórmula para se calcular o perímetro da circunferência é : $\mathrm{A} = 2 * \pi * \mathrm{r}$

15 - Entrar com os lados A, B e C de um paralelepípedo.

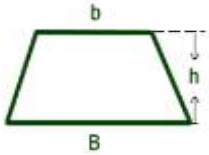
Calcular e imprimir o volume.

- Volume = A * B * C

16 - Elabore um algoritmo que calcule a área de um trapézio qualquer (figura meramente ilustrativa).

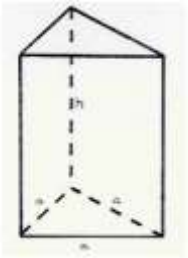
Proibido a venda e a distribuição deste produto

Área = $\frac{[(B+b)\times h]}{2}$




17 - Construa um algoritmo que possa calcular o volume de um prisma de base triangular (figura meramente ilustrativa).

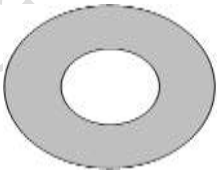
Volume = área da base x altura.



18 - Crie um algoritmo que possa calcular a área de uma coroa de forma circular (figura meramente ilustrativa).

Área da circunferência = PI * raio$^2$

Área = (Área da circunferência Maior) – (Área da circunferência menor)



19 - Elabore um algoritmo que possa calcular o volume de um cilindro (figura meramente ilustrativa).

Área da base = área da circunferência

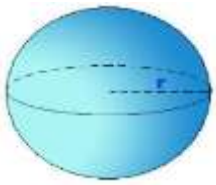
Volume = área da base x altura

20 - Elabore um algoritmo para calcular o volume e a área de uma esfera(figura meramente ilustrativa).

Área = $4 \times \pi \times r^2$

Volume = $\frac{4}{3} \times \pi \times r^3$

Proibido a venda e a distribuição deste produto



## Estruturas condicionais e condições lógicas

Muitas vezes precisamos verificar se determinadas condições são verdadeiras ou falsas.

Por exemplo: verificar se, entre dois números digitados pelo usuário, o primeiro é maior que o segundo ou se o número digitado é positivo.

Para estas situações utilizamos as estruturas condicionais.

As estruturas condicionais permitem que você possa verificar se uma determinada expressão lógica (condição) é verdadeira ou não.

Se a expressão for verdadeira ela executa um determinado trecho do código e em caso contrário poderá ser executado outro trecho de código ou simplesmente não executar nenhum trecho de código.

Uma expressão condicional sempre irá retornar um valor booleano, em outras palavras ela sempre irá retornar um valor **verdadeiro** ou **falso**.

| Operador | Significado |
|---|---|
| == | Igual a |
| != | Diferente de |
| > | Maior que |
| < | Menor que |
| >= | Maior ou igual a |
| <= | Menor ou igual a |

Proibido a venda e distribuição

## Operador Ternário

```
double nota = 5.00;
string texto = (nota >= 6.00) ? "Aprovado" : "Reprovado";
Console.WriteLine(texto);

nota = 9.99;
Console.WriteLine((nota >= 6.00) ? "Aprovado" : "Reprovado");

Console.WriteLine("Digite a nota do aluno: ");
```

```
nota = double.Parse(Console.ReadLine());


Console.WriteLine($"A nota do aluno é {nota}.");
Console.WriteLine((nota >= 6.00) ? "Aprovado" : "Reprovado");
```

## Lista de exercício 03

21 – Entrar com quatro notas de um aluno e imprimir a média ponderada destas notas, sabendo-se que os pesos são respectivamente 3, 5, 6 e 6.
Imprima também se o aluno está aprovado ou reprovado sabendo-se que para ser aprovado a média deve ser maior que 6,00.

22 – Entrar com três números e imprimir a média aritmética. Imprima também se o aluno está aprovado ou reprovado sabendo-se que para ser aprovado a média deve ser maior ou igual à 7,50.

23 - O algoritmo deve receber um número qualquer.
Se o número for par o algoritmo deve imprimir seu valor e seu valor elevado ao quadrado.
Se o número for ímpar o algoritmo deve imprimir seu valor e seu valor elevado ao cubo.

Proibida a distribuição deste produto

# IF

O **if** é um condicional avalia uma expressão lógica booleana.

Se o resultado for verdadeiro, o código contido dentro do bloco **if** será executado.

### Observe:

```
int a = 10;
int b = 100;

if ( a < b ) {
    Console.WriteLine("B é maior");
}
```

### IF/Else

```
int a = 10;
int b = 100;

if ( a < b ){
    Console.WriteLine("B é maior");
} else {
        Console.WriteLine("A é maior");
}
```

## Else-if

É possível avaliar diversos **else ifs** com uma determinada expressão. Como neste exemplo:

```
double valor = 1200;

if(valor < 500){
    valor += 50;
} else if(( valor >= 500) &&( valor < 600)) {
    valor += 100;
} else if ((valor >= 500) &&( valor < 700)) {
    valor += 110;
} else {
    valor += 250;
}
```

Lista de exercício 04

24 -Antes do racionamento de energia ser decretado, quase ninguém falava em quilowatts, mas agora, todos incorporaram essa palavra em seu vocabulário.

Sabendo-se que 100 quilowatts de energia custa um sétimo do salário mínimo, faça um algoritmo que receba o valor do salário mínimo e a quantidade de quilowatts gasta por uma residência.

Calcule e imprima:

- O valor em reais de cada quilowatt.
- O valor em reais a ser pago.
- O novo valor a ser pago por essa residência com um desconto de 10%.

25 - Criar um algoritmo que calcule e imprima a área e a hipotenusa de um triângulo retângulo.
Observação: Os valores devem ser positivos.

26 - Digitar um valor qualquer e imprimir se o valor digitado é “Par” ou “Ímpar”.
Atenção: os números dever ser maiores que zero.

27 - Ler um número qualquer positivo e maior que zero.

Se o valor do número elevado à quarta for par e múltiplo de cinco o algoritmo deverá imprimir a seguinte mensagem:

- “O número digitado foi __.”
- “Seu valor elevado ao quadrado é __.”
- “Seu valor elevado ao cubo é __.”
- “Seu valor elevado a sétima é __.”

Se o valor for ímpar o algoritmo deverá imprimir a seguinte mensagem:

- “O número digitado foi __.”
- “Sua raiz quadrada é __.”

Proibido a venda e a distribuição deste p

- “Sua raiz cúbica é __.”
- “Sua raiz a sétima é __.”

**Observação:** Se o usuário digitar um valor inválido o algoritmo deverá emitir uma mensagem de erro.

28 - Ler três números.

- Exibir os três números informando se eles são positivos, negativos ou nulos.
- Informar o maior número.

29 - Construir um algoritmo que leia dois valores numéricos e efetue sua adição.

- Caso o resultado da adição seja maior que 10, exibir os números digitados, o valor da adição e a raiz cúbica da adição.
- Caso contrário exibir somente os valores digitados e o valor da adição.

30 - Ler uma temperatura em graus Celsius e apresentá-la convertida em graus Fahrenheit.

A fórmula de conversão é: F = C*(9.0/5.0)+32.0, sendo F a temperatura em Fahrenheit e C a temperatura em Celsius.

31 - Efetuar a leitura de três valores e apresentar como resultado final a soma dos quadrados dos três valores lidos. Apresentar também se a soma é um número par ou ímpar.

32 - Receber o salário de um funcionário, exibir o valor deste salário, calcular e mostrar seu novo salário, sabendo que ele recebeu um aumento de 25%.

33 - Receber a altura do degrau de uma escada e a altura que o usuário deseja alcançar subindo a escada. Calcular e mostrar quantos degraus o usuário deverá subir para atingir seu objetivo, sem se preocupar com a altura do usuário.

Proibida a venda e a distribuição deste produto

Proibido a venda e a distribuição deste produto