

# Fundamentos de Programação

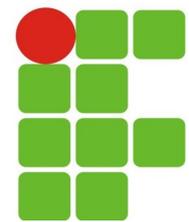
## Programação com sequência

**Prof. M.Sc.: João Paulo Q. dos Santos**

**E-mail: [joao.queiroz@ifrn.edu.br](mailto:joao.queiroz@ifrn.edu.br)**

**Página: <http://docente.ifrn.edu.br/joaoqueiroz/>**

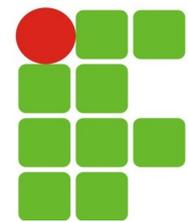
*"Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar."*



# Etapas de ação de um computador

- Entrada de dados:
  - O computador recebe os dados do mundo externo, podendo armazená-los na memória principal para realizar algum tipo de processamento, ou armazenar na memória secundária para uso futuro;
- Processamento de dados:
  - Por meio de um programa executando em sua memória, faz a transformação dos dados entrados ou previamente armazenados, tornando elementos que possam ser usados como fontes de informação para o mundo externo;
- Saída de dados:
  - O computador envia os dados processados para a memória ou armazena para o mundo externo.

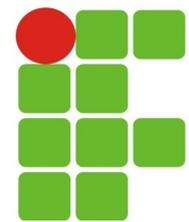
Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar."



# Tipo de dados primitivos

- Os dados são elementos do mundo exterior, que representam dentro de um computador as informações manipuladas pelos seres humanos;
- Eles podem ser classificados em três tipos primitivos: numéricos, caracteres e lógicos;
- **Inteiro:**
  - São caracterizados como tipos inteiros os dados numéricos positivos ou negativos, excluindo-se destes qualquer número fracionário (conjunto de números reais).
- **Real:**
  - São caracterizados como tipos reais os dados numéricos positivos, negativos e números fracionários. (Observe que quando estamos tratando números em programas devemos utilizar a notação inglesa e substituir a vírgula pelo ponto).

"Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar."



# Tipo de dados primitivos

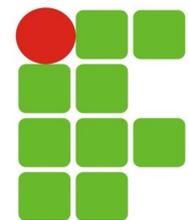
- **Caractere:**

- São caracteres delimitados entre aspas (“ ”). Eles são representados por letras, números e símbolos . Este tipo de dado é também conhecido como: alfanumérico, string, carácter ou cadeia.

- **Lógico:**

- São caracterizados como tipos lógicos os dados com valores binários do tipo sim e não, verdadeiro e falso, 1 e 0, sendo que este tipo de dado poderá representar apenas um dos valores. Ele é chamado por alguns de tipo booleano, devido à contribuição do filósofo e matemático inglês George Boole na área da lógica matemática.

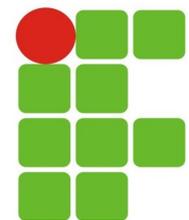
“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”



# Uso de variáveis

- Variável é tudo que está sujeito a variações, que é incerto , instável ou inconstante;
- Quando se fala de computadores, temos que ter em mente que o volume de informações a serem tratadas é grande e diversificado;
- Desta forma, os dados a serem processados serão bastante variáveis;
- Todo dado a ser armazenado na memória de um computador deve ser previamente identificado, ou seja, primeiro é necessário saber qual o seu tipo para depois fazer o seu armazenamento adequado;
- Estando armazenado o dado desejado, ele poderá ser utilizado e manipulado a qualquer momento.

“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”

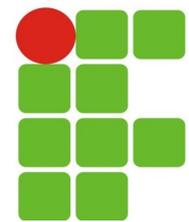


# Uso de variáveis

- Para utilizar o conceito de variável, imagine que a memória de um computador é um grande arquivo com várias gavetas, sendo que cada gaveta pode apenas armazenar um único valor (seja ele numérico, lógico ou caractere);
- Se for um grande arquivo com várias gavetas, você há de concordar que é necessário identificar com um nome a gaveta que se pretende utilizar. Desta forma o valor armazenado pode ser utilizado a qualquer momento;



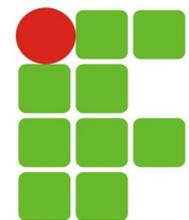
“Ve  
nce,  
mas sim aquele que nunca para de lutar.”



# Uso de variáveis

- O nome de uma variável é utilizado para sua identificação e posterior uso dentro de um programa;
- Sendo assim, é necessário estabelecer algumas regras de utilização das variáveis:
  - Nomes de uma variável poderão ser atribuídos com um ou mais caracteres;
  - O primeiro caractere do nome de uma variável não poderá ser, em hipótese alguma, um número; sempre deverá ser uma letra;
  - O nome de uma variável não poderá possuir espaços em branco;
  - O nome de uma variável não poderá ser uma palavra reservada (uma instrução ou comando);
  - Não poderão ser utilizados outros caracteres a não ser letras, números e sublinhado.

“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”



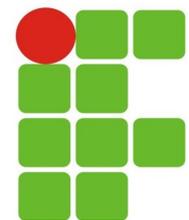
# Uso de variáveis

- São nomes válidos de variáveis: NOMEDOUSUARIO, telefone, x, z, delta\_25, z1, entre outros. São nomes inválidos de variáveis: NOME DO USUARIO, 25\_delta, telefone#

**<tipo>** **<nome>** = **<valor>**

- Ex.: **int inteiro** = **10**;

“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”



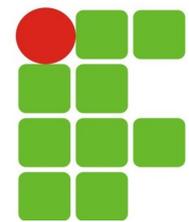
# Uso de constantes

- Tem-se como definição de constante tudo aquilo que é fixo, estável ou imutável;

```
const <tipo> <nome> = <valor>;
```

- Ex.: **const float PI = 3.14159265;**

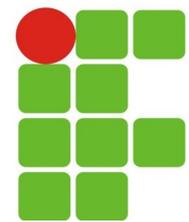
“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”



# Operadores Aritméticos

- Tanto variáveis como constantes poderão ser utilizadas na elaboração de cálculos matemáticos, ou seja, na elaboração de expressões aritméticas, desde que estejam estabelecidas como do tipo real ou inteira, e para que isto ocorra é necessário à utilização de operadores aritméticos;
- Os operadores aritméticos são classificados em duas categorias, sendo binários ou unários:
- São binários quando atuam em operações de exponenciação, multiplicação, divisão, adição e subtração;
- São unários quando atuam na inversão de um valor, atribuindo a este o sinal positivo ou negativo;

“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”

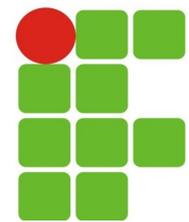


# Operadores Aritméticos

- Veja em seguida, a tabela de prioridade matemática existente quando da utilização destes operadores:

Tabela de operadores aritméticos					
Operador	Operação	Descrição	Tipo	Prioridade	Resultado
+	"+" ou "n"	Manutenção de sinal	Unário	-	Positivo
-	-n	Inversão de sinal	Unário	-	Negativo
←	$x \leftarrow n$	Atribuição do valor "n" a "x"	Binário	-	Positivo ou Negativo
↑	$x \uparrow n$	Exponenciação de $x^n$	Binário	1	Inteiro ou Real <sup>15</sup>
↑(1/n)	$x \uparrow (1/n)$	Radiciação de $\sqrt[n]{x}$	Binário	1	Real
/	$x/n$	Divisão de "x" por "n"	Binário	2	Real
*	$x * n$	Multiplicação de "x" por "n"	Binário	2	Inteiro ou Real

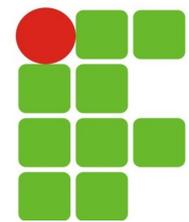
Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar."



# Operadores Aritméticos

Tabela de operadores aritméticos					
Operador	Operação	Descrição	Tipo	Prioridade	Resultado
+	$x + n$	Adição de "x" com "n"	Binário	3	Inteiro ou Real
-	$x - n$	Subtração de "n" de "x"	Binário	3	Inteiro ou Real
div	$x \text{ div } n$	Divisão de "x" por "n"	Binário	4	Inteiro

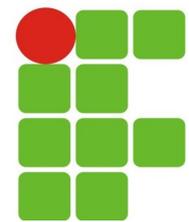
“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”



# Expressões aritméticas

- É muito comum lidarmos com expressões aritméticas;
- Uma vez que a maior parte do todo trabalho computacional está relacionado e envolve a utilização de cálculos;
- Estas expressões são definidas pelo relacionamento existente entre variáveis e constantes numéricas por meio da utilização dos operadores aritméticos;
- Considere a fórmula:  $AREA = \pi \cdot RAI O^2$ .
- Para o cálculo da área de uma circunferência, onde estão presentes as variáveis  $AREA$  e  $RAIO$ , a constante  $\pi$  (3.14159) e os operadores aritméticos de multiplicação e também a operação de potência, que eleva o valor da variável  $RAIO$  ao quadrado;

“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”

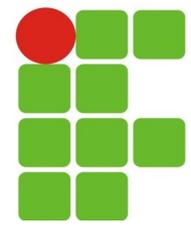


# Expressões aritméticas

- As expressões aritméticas escritas em computação seguem um formato um pouco diferente da forma conhecida em matemática;
- Por exemplo a expressão:  $X = \{ 43 \cdot [ 55 : ( 30 + 2 ) ] \}$  é escrita na forma computacional como:  $X \leftarrow (43 * (55 / (30 + 2)))$ .
- Perceba que as chaves e colchetes são abolidos, utilizando-se em seu lugar apenas os parênteses;
- É também substituído o sinal de “igual a” (=) pelo simbolo ( $\leftarrow$ ) “seta para a esquerda”, que indica a operação de atribuição;
- E se a fórmula a ser utilizada fosse para efetuar o cálculo da área de um triângulo, em que é necessário efetuar a multiplicação da base pela altura e em seguida dividir pela constante 2, como ficaria? Observe abaixo a fórmula padrão:

$$AREA = \frac{BASE \cdot ALTURA}{2}$$

“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”

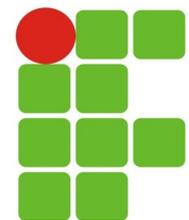


# Expressões aritméticas

- Ela deveria ser escrita como:

$$\text{AREA} \leftarrow (\text{BASE} * \text{ALTURA}) / 2.$$

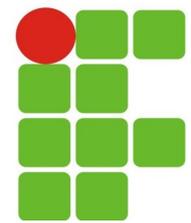
“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”



# Instruções básicas

- As instruções são representadas pelo conjunto de palavras-chaves (vocabulário) de uma linguagem de programação que tem por finalidade comandar, em um computador, o seu funcionamento e a forma como os dados armazenados devem ser tratados;
- Deve-se ainda considerar que existem várias linguagens de programação, como: Pascal, C++, Java, entre outras, sendo que uma determinada instrução para realizar uma tarefa em um computador poderá ser escrita de forma diferente, dependendo da linguagem utilizada.

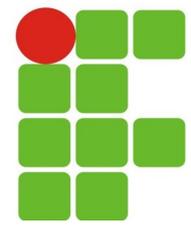
“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”



# Entrada, processamento e saída

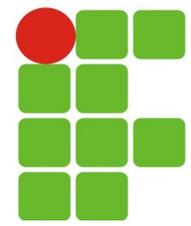
- Para criar um programa que seja executável dentro de um computador, deve-se ter em mente três pontos de trabalho: a entrada de dados, o seu processamento e a saída deles;
- Todo programa estará trabalhando com estes três conceitos;
- Se os dados forem entrados de forma errada, serão conseqüentemente processados de forma errada e resultarão em respostas erradas;
- Desta forma, dizer a alguém que foi erro do computador é ser um tanto “mediocre”;
- E isto é o que mais ouvimos quando nosso saldo está errado e vamos ao banco fazer uma reclamação, ou quando recebemos uma cobrança indevida;
- Se houve algum erro, é porque foi causado por falha humana. Realmente é impossível um computador errar por vontade própria, pois vontade é uma coisa que os computadores não tem.

“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”



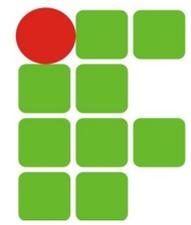
# Code::Blocs IDE

“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”



# Exercício de Aprendizagem

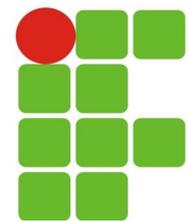
“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”



# Exemplo 1

Desenvolver a lógica para um programa que efetue o cálculo da área de uma circunferência, apresentado a medida da área calculada:

“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”

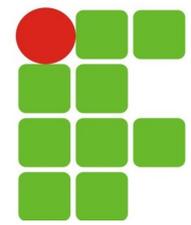


# Algoritmo

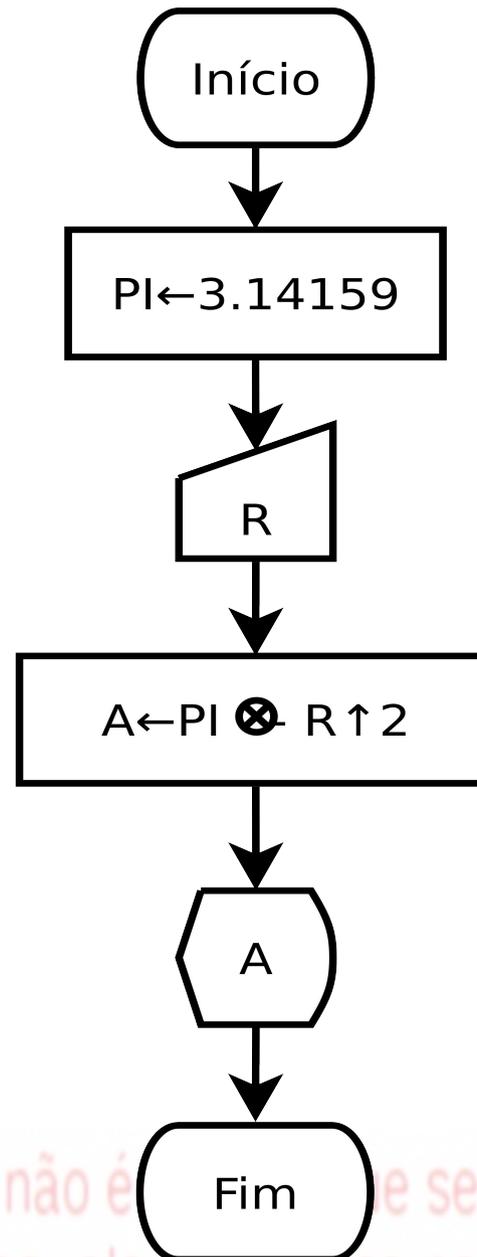
- Para efetuar o cálculo da área de uma circunferência é necessário conhecer a fórmula que executa este cálculo, sendo esta:
  - $A = \pi.R^2$ , em que  $A$  é a variável que conterà o resultado do cálculo da área,  $\pi$  é o valor de  $PI$  (3.14159, sendo uma constante na fórmula) e  $R$  o valor do raio.
- Sendo assim, basta estabelecer:
  - Ler um valor para o raio, no caso variável  $R$ ;
  - Estabelecer que  $\pi$  possui o valor de 3.14159;
  - Efetuar o cálculo da área, elevando ao quadrado o valor de  $R$  e multiplicando por  $\pi$ ;
  - Apresentar o valor da variável  $A$ ;
- A fórmula para o cálculo da área passará a ser escrita como:

$$A \leftarrow 3.14159 * R^2$$

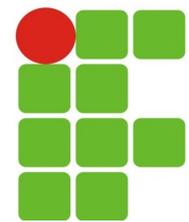
“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”



# Diagrama de Bloco



"Vencedor não é quem sempre vence,  
mas sim aquele que nunca para de lutar."



# Código

```
#include <iostream>

using namespace std;

int main() {

    const float PI = 3.14159;
    float A, R;

    cout<<"Digite R: ";

    cin >> R;

    A = PI*(R*R);

    cout<<"A área é: "<<A;

    return 0;
}
```

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

using namespace std;

int main() {

    const float PI = 3.14159;
    float A, R;

    cout<<"Digite R: ";

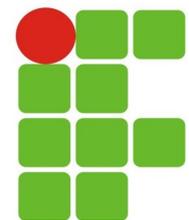
    cin >> R;

    A = PI*pow(R,2);

    cout<<"A área é: "<<A;

    return 0;
}
```

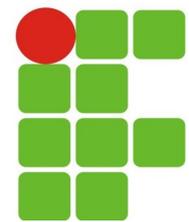
"Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar."



## Exemplo 2

Construir um programa que efetue o cálculo do salário líquido de um professor. Para fazer este programa, você deverá possuir alguns dados, tais como: valor da hora aula, número de horas trabalhadas no mês e percentual de desconto do INSS. Em primeiro lugar, deve estabelecer qual será o seu salário bruto para efetuar o desconto a ter o valor do salário líquido.

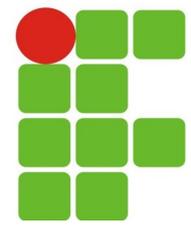
“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”



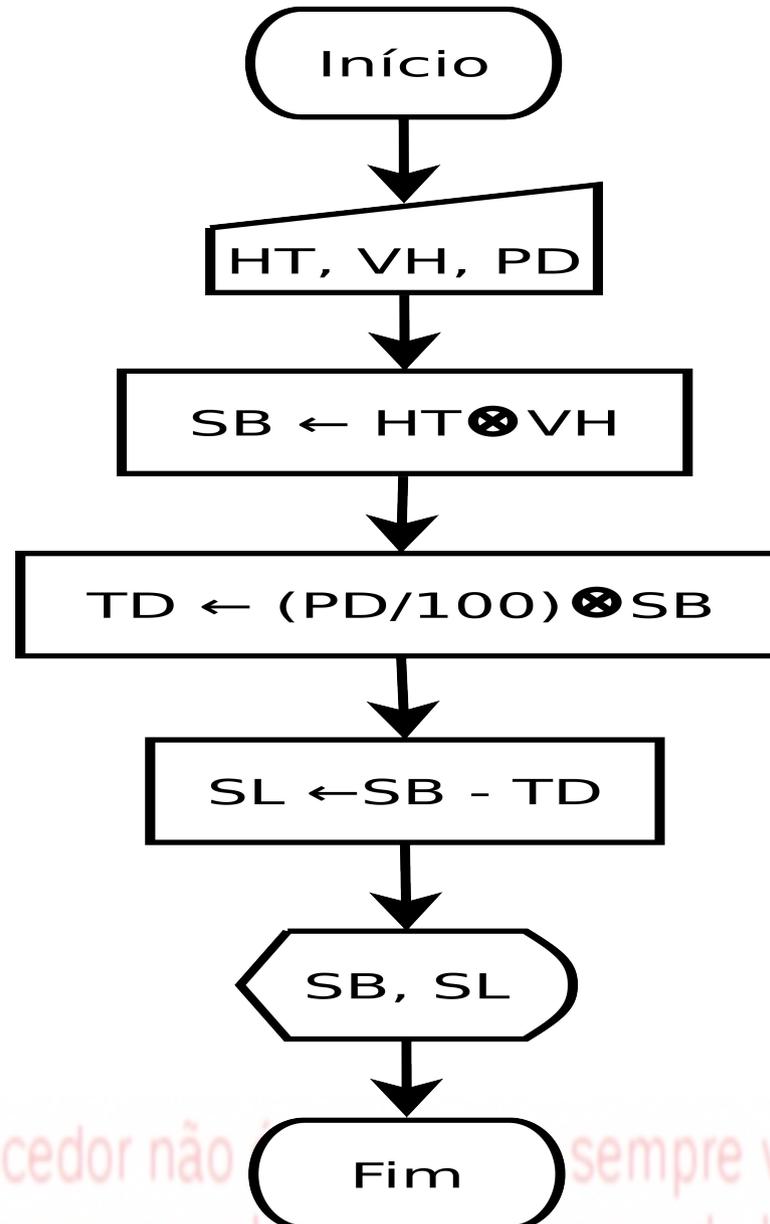
# Algoritmo

- Estabelecer a leitura da variável HT (horas trabalhadas no mês);
- Estabelecer a leitura da variável VH (valor hora aula);
- Estabelecer a leitura da variável PD (percentual de desconto);
- Calcular o salário bruto (SB), sendo este a multiplicação das variáveis HT e VH;
- Calcular o total de desconto (TD) com base no valor de PD dividido por 100;
- Calcular o salário líquido (SL), deduzindo o desconto do salário bruto;
- Apresentar os valores dos salários bruto e líquido: SB e SL.

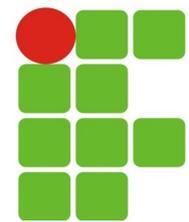
“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”



# Diagrama de Bloco



"Vencedor não é sempre aquele que sempre vence,  
mas sim aquele que nunca para de lutar."



# Código

```
#include <iostream>

using namespace std;

int main() {

    int HT;
    float VH, PD, TD, SB, SL;

    cout<<"Digite HT: ";
    cin >> HT;

    cout<<"Digite VH: ";
    cin >> VH;

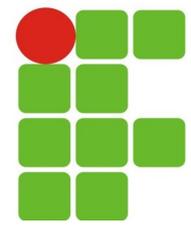
    cout<<"Digite PD: ";
    cin >> PD;

    SB = HT * VH;
    TD = (PD/100) * SB;
    SL = SB - TD;
```

```
    cout<<"Salario Bruto é: "<<SB<<endl;
    cout<<"Salario Liquido é: "<<SL<<endl;

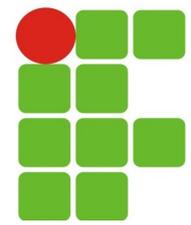
    return 0;
}
```

Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar."



# Perguntas?

“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”



# Lista de exercício

“Vencedor não é aquele que sempre vence,  
mas sim aquele que nunca para de lutar.”