

INSTITUTO FEDERAL  
BAIANO

# Algoritmos e Introdução à Programação

---

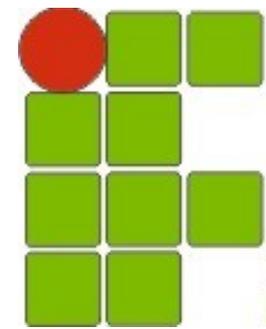
## Lógica e Linguagem de Programação

*Prof. José Honorato Ferreira Nunes*

**honoratonunes@softwarelivre.org**

**<http://softwarelivre.org/zenorato/honoratonunes>**





INSTITUTO FEDERAL  
BAIANO



---

# Linguagem C

*Prof. José Honorato Ferreira Nunes*

**honoratonunes@softwarelivre.org**

**<http://softwarelivre.org/zenorato/honoratonunes>**

# Resumo da aula

---

- Introdução a Linguagem C
- Variáveis e C
- Tipos de dados em C
- Operadores e Expressões em C
- Comandos de Entrada e Saída em C

# Introdução a Linguagem C

---

A linguagem C foi criada por Dennis Ritchie, em 1972, no centro de Pesquisas da Bell Laboratories. Sua primeira utilização importante foi a reescrita do Sistema Operacional UNIX, que até então era escrito em assembly.

Em meados de 1970 o UNIX saiu do laboratório para ser liberado para as universidades. Foi o suficiente para que o sucesso da linguagem atingisse proporções tais que, por volta de 1980, já existiam várias versões de compiladores C oferecidas por várias empresas, não sendo mais restritas apenas ao ambiente UNIX, porém compatíveis com vários outros sistemas operacionais.

# Introdução a Linguagem C

---

O C é uma linguagem de propósito geral, sendo adequada à programação estruturada. No entanto é mais utilizada escrever compiladores, analisadores léxicos, bancos de dados, editores de texto, etc..

A linguagem C pertence a uma família de linguagens cujas características são: portabilidade, modularidade, compilação separada, recursos de baixo nível, geração de código eficiente, confiabilidade, regularidade, simplicidade e facilidade de uso.

# Introdução a Linguagem C - sintaxe

---

A sintaxe são regras detalhadas para cada construção válida na linguagem C.

Estas regras estão relacionadas com os tipos, as declarações, as funções e as expressões.

Os tipos definem as propriedades dos dados manipulados em um programa.

As declarações expressam as partes do programa, podendo dar significado a um identificador, alocar memória, definir conteúdo inicial, definir funções.

As funções especificam as ações que um programa executa quando roda.

A determinação e alteração de valores, e a chamada de funções de I/O são definidas nas expressões.

# Introdução a Linguagem C - sintaxe

---

As funções são as entidades operacionais básicas dos programas em C, que por sua vez são a união de uma ou mais funções executando cada qual o seu trabalho. Há funções básicas que estão definidas na biblioteca C .

As funções `printf()` e `scanf()` por exemplo, permitem respectivamente escrever na tela e ler os dados a partir do teclado. O programador também pode definir novas funções em seus programas, como rotinas para cálculos, impressão, etc.

# Introdução a Linguagem C - sintaxe

---

Uma função importante em todo programa em C é a função main (cuja tradução é principal). Esta será sempre a primeira função do programa a ser executada.

```
main ( )
```

```
{
```

```
}
```

# Variáveis

---

Como o próprio nome sugere, as variáveis, podem conter valores diferentes a cada instante de tempo, ou seja, seu conteúdo pode variar de acordo com as instruções do algoritmo.

As variáveis são referenciadas através de um nome (identificador) criado por você durante o desenvolvimento do algoritmo.

# Variáveis

---

É um local na memória principal, isto é, um endereço que armazena um conteúdo (informação) que pode ser modificado.

Em C, não é possível ter variáveis que comecem com dígito e espaços não são permitidos.

Exemplos de nomes de variáveis indevidas: 2w, peso do aluno, sal/hora.

Observação: em C usualmente são utilizadas variáveis em minúsculo e constantes em maiúsculos.

# Variáveis

---

Para as variáveis, devem ser feitas no início do programa (ou de um bloco) as declarações (de tipo) de variáveis.

Declarada uma variável, toda vez que ela for referenciada em qualquer comando do programa, o computador vai trabalhar com o conteúdo de seu endereço, que é o valor da variável.

Uma variável não pode ter o mesmo nome de uma palavra-chave de C, como por exemplo: main, cout, int, float, char, short, return, case, void.

As variáveis só podem armazenar informações ou dados sempre de um mesmo tipo (inteiro, real, caractere ou char).

# Variáveis

---

## EXEMPLOS DE NOMES VÁLIDOS

nome\_candidato

endereco

RG

mes\_ferias

dataNasc

fone1

## EXEMPLOS DE NOMES INVÁLIDOS

nome candidato

endereço

R.G.

mês\_férias

data-Nasc

1fone

# Tipos de dados

---

Quando declaramos uma variável, precisamos identificar o tipo de informação que desejamos armazenar nela. Existem diversos tipos de dados e muitos deles são comuns na grande maioria das linguagens de programação. No nosso estudo de lógica de programação, porém, utilizaremos apenas alguns dos principais.

# Tipos de dados

---

Os tipos de dados básicos com os quais iremos trabalhar são:

Nome	Tamanho em bits	Faixa de valores
char	8	-128 a 127
int	16	-32.768 a 32.768
float	32	$10^{-38}$ a $10^{38}$
long int	32	-2.147.483.648 a 2.147.483.648
double	64	$10^{-308}$ a $10^{308}$

# Declaração e Atribuição de Variáveis

---

tipo nome\_var;

ou

tipo nome\_var1, nome\_var2, ..., nome\_varn;

Exemplo: int numero, soma;  
          char nome;

variavel = expressão;

Exemplos: a = 5;

# Operadores

---

=	atribuição
!=	diferente
<=	Menor igual
>=	Maior igual
==	igual
%	Resto da divisão entre inteiros

# Operadores

---

## Operadores Aritméticos:

Aritméticos	Tipo	Operação	Prioridade
+	Binário	Adição	5
-	Binário	Subtração	5
%	Binário	Resto da divisão	4
*	Binário	Multiplicação	3
/	Binário	Divisão	3
++	Unário	Incremento	2
--	Unário	Decremento	2
+	Unário	Manutenção do sinal	1
-	Unário	Inversão do sinal	1

# Operadores

---

**Exemplo:**

**a=a+b;**

**a = 4 \* 2 + 3;**

**Observação:**

**a++ é similar a a = a + 1;**

**b-- é similar a b = b -1;**

# Operadores

---

## Outros Operadores:

Relacionais		Lógicos	
>	Maior que	&&	And (e)
>=	Maior ou igual		Or (ou)
<	Menor que	!	Not (não)
<=	Menor ou igual		
==	Igual		
!=	Diferente		

# Comandos de Entrada e Saída

---

Comando de Impressão printf/saída (Escreva).

Através da função pré-definida printf(), cujo protótipo está contido também no arquivo stdio.h. Sua sintaxe é a seguinte:

```
printf( "Expressão" , lista de argumentos );
```

Sintaxe:

<expressão> Mensagens que serão exibidas.

<lista de argumentos> pode conter identificadores de variáveis, expressões aritméticas ou lógicas e valores constantes.

# Comandos de Entrada e Saída

---

<b>Código</b>	<b>Tipo</b>	<b>Elemento armazenado</b>
<b>%c</b>	<b>char</b>	<b>um único caractere</b>
<b>%d</b> ou <b>%i</b>	<b>int</b>	<b>um inteiro</b>
<b>%f</b>	<b>float</b>	<b>um número em ponto flutuante</b>
<b>%lf</b>	<b>double</b>	<b>ponto flutuante com dupla precisão</b>
<b>%e</b>	<b>float ou double</b>	<b>um número na notação científica</b>
<b>%s</b>	<b>.....</b>	<b>uma cadeia de caracteres</b>

# Comandos de Entrada e Saída

---

Primeiro Programa em C!

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

```
main()
```

```
{
```

```
    printf("%s está a %d milhões de milhas do sol","Vênus",67);
```

```
}
```

# Comandos de Entrada e Saída

---

Primeiro Programa em C!

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

```
main()
```

```
{
```

```
    printf("Valor inteiro atribuído foi %d para o caracter  
%c e um float foi de %f \",99,\"a\",1.45);
```

```
}
```

# Comandos de Entrada e Saída

---

Primeiro Programa em C!

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

```
main()
```

```
{
```

```
    printf("Se quisesse imprimir uma string :  
%s", "Minha string!");
```

```
}
```

# Comandos de Entrada e Saída

## Leitura de dados - scanf() - (Leia)

---

Ela é o complemento de printf() e nos permite ler dados formatados da entrada padrão (teclado). Sua sintaxe:

```
scanf("expressão de controle", argumentos);
```

Exemplo:

```
int m;  
scanf ("%d",&m);
```

- %d indicativo do tipo, neste caso do tipo inteiro.
- &m operador utilizado para obter o endereço de memória da variável.

# Comandos de Entrada e Saída

---

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

```
main ()
```

```
{
```

```
int idade;
```

```
printf("Digite a sua idade : ");
```

```
scanf("%d",&idade);
```

```
printf("A sua idade é %d",idade);
```

```
return 0;
```

```
}
```

# Atividades

---

- Crie um algoritmo para calcular a média de consumo de combustível de um veículo qualquer. O usuário deverá informar: quilometragem inicial, quilometragem final e a quantidade de litros consumida durante a viagem.

# Atividades

---

- Crie um algoritmo para armazenar a matrícula, idade e o sexo (M ou F) de um aluno. O programa deve solicitar que o usuário informe os dados acima e depois mostrar essas informações na tela.

# Bibliografia

- ❑ MANZANO, Wilson Y. Yamaturni-São Paulo-SP. **Lógica estruturada para programação de computadores**, Ed. Érica 1997 e 2001.
- ❑ MORAES, Celso Roberto. **Estruturas de Dados e Algoritmos**. Ed. Érica, São Paulo
- ❑ LOPES, Anita. **Introdução à programação**. Rio de Janeiro: Campus, 2002.
- ❑ BENEDUZZI, Humberto M. e METZ, João A. **Lógica e Linguagem de Programação – Introdução ao Desenvolvimento de Software (1ª edição)**. Editora do Livro Técnico, 2010.
- ❑ SEBESTA, Robert W. **Conceitos de linguagens de programação**. 9. ed. Porto Alegre: Bookman, 2003.
- ❑ CORMEN, Thomas H. **Algoritmos: teoria e prática**. Rio de Janeiro: Campus, 2002.
- ❑ ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. **Fundamentos da programação de computadores: algoritmos, Pascal e C/C++ e Java**. 2. ed. São Paulo: Pearson Prentice Hall, 2008.
- ❑ ZIVIANI, Nivio. **Projeto de algoritmos com implementações em Pascal e C**. São Paulo: Pioneira Thomson Learning.