



Residência  
Tecnológica  
em Sistemas  
Embarcados

# Introdução a Linguagem C

Unidade 3 | Capítulo 1 | Aula 1

Executores:



Coordenação:



Iniciativa:



# Sumário

- Revisão de lógica de Programação;
- Introdução a Linguagem C;
- Declaração de Variáveis, contantes e tipos de dados;
- Bibliotecas;
- Entrada e saída com linguagem C;
- Operadores e atribuições;
- Exercícios Resolvidos.

# Revisão de lógica de Programação

Dividir o problema apresentado em três fases fundamentais de análise:



**Método de construção de um Algoritmo:**

1. Entender o problema;
2. Definir dados de entrada;
3. Definir o processamento;
4. Definir dados de saída;
5. Construir o algoritmo usando a sintaxe adequada;
6. Validar o algoritmo (Testar).

**OBS: Pela lógica computacional a entrada dos dados serão processados para se obter uma saída.**

# Revisão de lógica de Programação

## Exemplo 1:

Calcular o delta =  $b^2 - 4*a*c$  de uma equação do segundo grau.

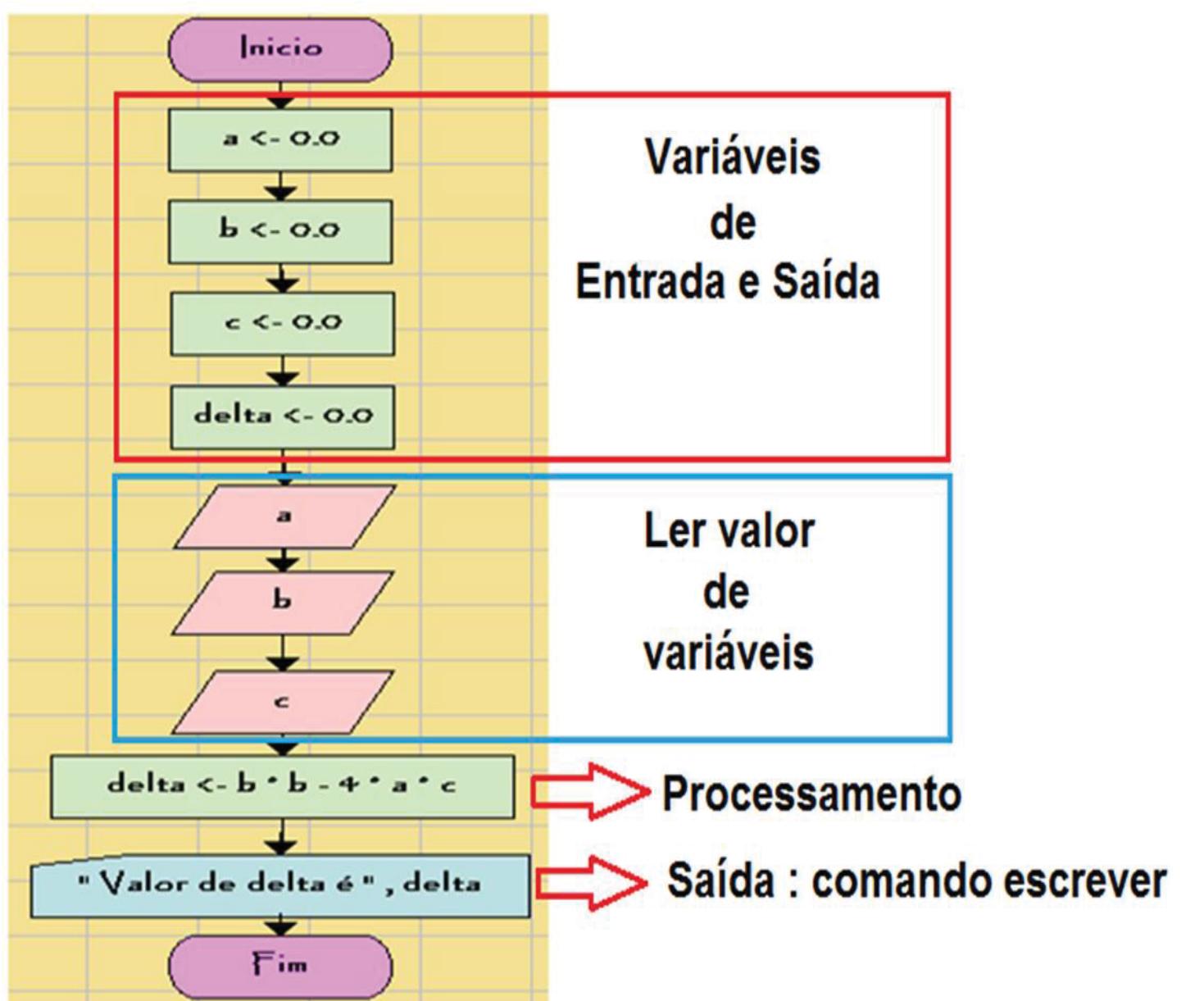
**Para montar o algoritmo proposto, faremos três perguntas:**

1. Quais são os dados de entrada ?
2. Qual será o processamento a ser utilizado ?
3. Quais serão os dados de saída ?

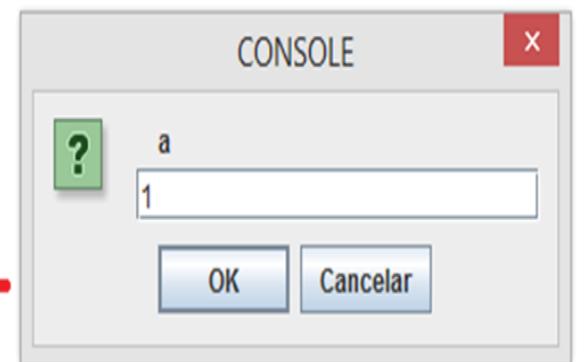
# Revisão de lógica de Programação

## Exemplo 1:

Calcular o delta =  $b^2 - 4*a*c$  de uma equação do segundo grau.



```
inicio
    // variaveis de entrada e saída
    real a,b,c
    real delta
    ler a,b,c
    // Processamento
    delta <- b*b - 4*a*c
    // saída do algoritmo
    escrever "valor de delta é ",delta
fim
```



```
1
2
3
```

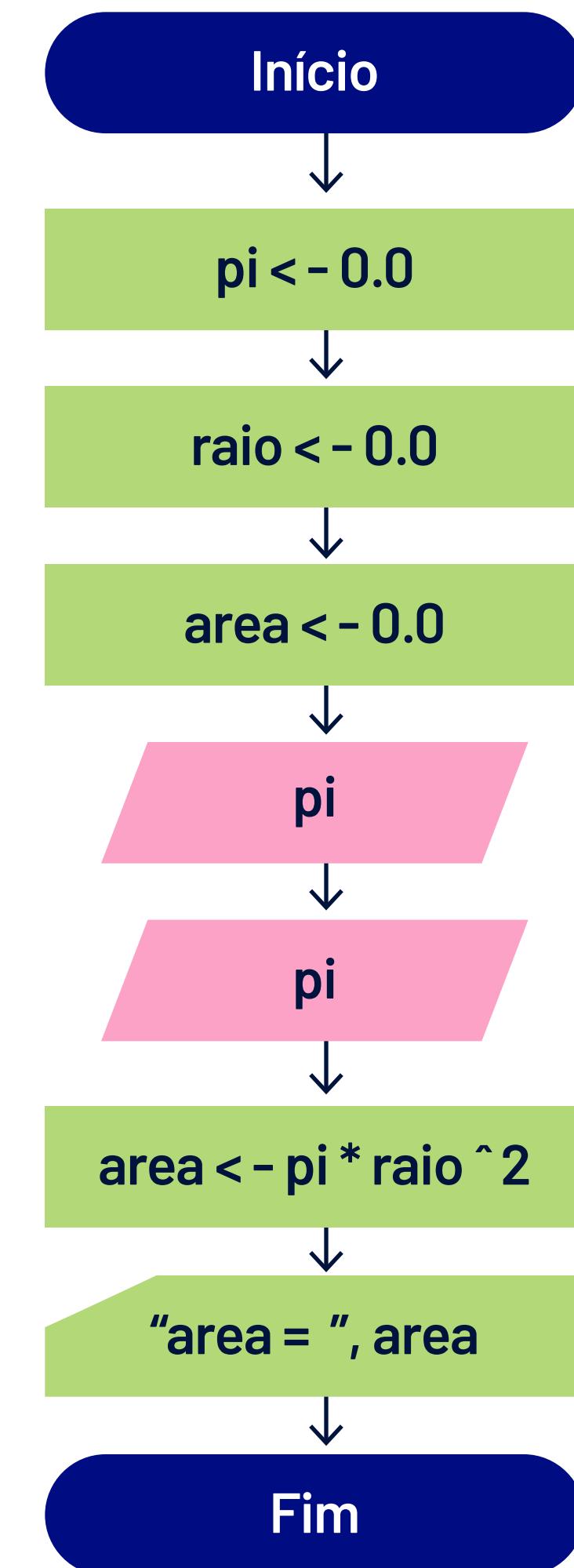
valor de delta é -8.0

# Revisão de lógica de Programação

## Exemplo 2:

Calcular a área de uma circunferência  
( área = PI\*R<sup>2</sup> ).

```
inicio
    // VARIÁVEIS DE ENTRADA E SAÍDA
    real pi, raio
    real area
    ler pi
    ler raio
    // PROCESSAMENTO
    area <- pi * raio ^ 2
    // SAÍDA
    escrever "area = ", area
fim
```



# Introdução a Linguagem C

## Características:

Linguagem Baixo nível;

Portabilidade;

Tipagem Estática;

Case Sensitive;

Ponteiros.

## Estrutura básica de um programa em C:

```
#include <bibliotecas>

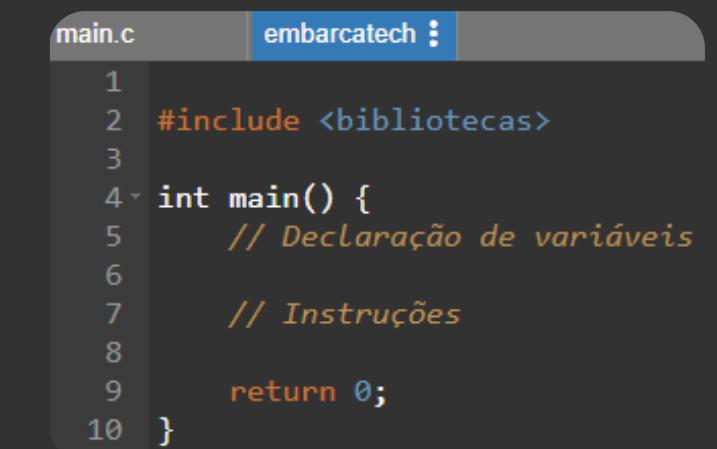
int main(){

    // Declaração de variáveis

    // Instruções

    return 0;

}
```



```
main.c | embarcatech ::

1 #include <bibliotecas>
2
3 int main(){
4     // Declaração de variáveis
5
6     // Instruções
7
8     return 0;
9 }
10 }
```



## Compilador:

[https://www.onlinegdb.com/online\\_c\\_compiler](https://www.onlinegdb.com/online_c_compiler)

# Declaração de Variáveis

**Sintaxe 1:** tipo\_variavel nome\_variavel ;

Variáveis básicas		Modificadores de tipo		Outros modificadores	
char	caractere	signed	Valores negativos e positivos	static	Mantém seu valor entre chamada de funções.
int	inteiro	unsigned	Valores não negativos	const	Valor de uma variável não pode ser modificado após a inicialização
float	real	short	Diminui o tamanho de um tipo inteiro		
		long	Aumenta o tamanho de um tipo inteiro		

**Sintaxe 2:** <modificador> tipo\_variavel nome\_variavel ;

```
main.c      embarcatech ::

1 int main() {
2     int age = 30; // Variável inteira com valor inicial 30
3     const float pi = 3.14159; // Constante pi com valor fixo
4     static int count = 0; // Variável estática, incrementada a cada chamada da função
5 }
```

# Bibliotecas

**Definição :** Conjunto de funções pre-definidas.

Tipo	Aplicação	Funções
Gerenciamento de arquivos , entrada e saída:	stdio.h	printf, scanf, fopen, fclose, etc.
Manipulação de strings:	string.h	strcpy, strcat, strlen, etc.
Matemática:	math.h	sin, cos, sqrt, pow, etc.
Alocação de memória:	stdlib.h	malloc, calloc, free
Conversão de tipos:	stdlib.h	atoi, atof
Controle de processos:	stdlib.h	exit, system
Manipulação de tempo:	time.h	time, localtime, difftime

```
main.c      embarcatech ::

1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     double x = 2.0;
6     double resultado = sqrt(x);
7     printf("A raiz quadrada de %.2lf é %.2lf\n", x, resultado);
8     return 0;
9 }
```

# Entrada e saída com linguagem C

Tipo	Função	Exemplo
Entrada	scanf(formato, lista de endereços)	<a href="https://onlinegdb.com/zErmU1j08">https://onlinegdb.com/zErmU1j08</a>
Saída	printf(formato , argumentos)	<a href="https://onlinegdb.com/JBYSb1bh7">https://onlinegdb.com/JBYSb1bh7</a>

```
main.c
1 #include <stdio.h>
2 #include <string.h>
3
4 int main() {
5
6     char nome[50];
7     int idade;
8     float altura;
9
10
11    printf("Digite seu nome: ");
12    scanf("%s", nome);
13
14    printf("Digite sua idade: ");
15    scanf("%d", &idade);
16
17    printf("Digite sua altura: ");
18    scanf("%f", &altura);
19
20    printf("\nSeus dados:\n");
21    printf("Nome: %-20s Idade: %d\n", nome, idade);
22    printf("Altura: %.2f metros\n", altura);
23
24 }
```

## Formatos

- %d Inteiro decimal
- %f Ponto flutuante
- %c Caractere
- %s String
- %x Inteiro Hexadecimal

## Formatos

- \n Nova linha
- \t Tabulação
- \\" Barra invertida
- \\" Aspas

# Entrada e saída com linguagem C

Tipo	Operador	Explicação	Exemplo
Aritméticos	+	Adição	valor = 3 + 5; // 8
	-	subtração	valor = 5 - 5; // 0
	*	multiplicação	valor = 3 * 5; // 15
	/	divisão	valor = 30 / 5; // 6
	%	módulo	valor = 30 % 5; // 0
Atribuição	=	Atribuição simples	valor = 30;
Incremento e decremento	++	incremento	valor++
	--	decremento	valor--

# Operadores e Atribuição

Tipo	Operador	Explicação	Exemplo
Lógicos	&&	And	(valor>18)&&(valor<23)
		or	(valor>18)   (valor<23)
	!	negação	(valor != 15)
Relacionais	==	Igual relacional	(valor == 15)
	!=	diferente	(valor != 15)
	<	Menor que	(valor < 15)
	>	Maior que	(valor > 15)
	>=	Maior ou igual a	(valor >= 15)
	<=	Menor ou igual a	(valor <= 15)

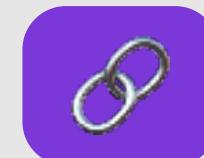
# Exercícios Resolvidos

# Exercícios Resolvidos

## Exemplo 1:

Calcular o delta =  $b^2 - 4*a*c$  de uma equação do segundo grau.

```
main.c      | embarcatech :|  
1 #include <stdio.h>  
2 #include <math.h>  
3  
4 int main() {  
5     double a, b, c, delta;  
6  
7     printf("Digite os coeficientes da equacao do segundo grau (a, b, c): ");  
8     scanf("%lf %lf %lf", &a, &b, &c);  
9  
10    delta = pow(b, 2) - 4 * a * c;  
11  
12    printf("O valor de delta e: %.2lf\n", delta);  
13}
```



Link:

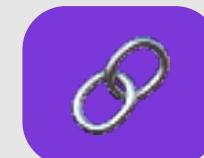
<https://www.onlinegdb.com/dyMkiemss>

# Exercícios Resolvidos

## Exemplo 2:

Calcular a área de uma circunferência  
( área = PI\*R<sup>2</sup> ).

```
main.c
1 #include <stdio.h>
2 #include <math.h>
3
4 #define PI 3.14159
5
6 int main() {
7     double raio, area;
8
9     printf("Digite o valor do raio da circunferencia: ");
10    scanf("%lf", &raio);
11
12    area = PI * pow(raio, 2);
13
14    printf("A area da circunferencia e: %.2lf\n", area);
15 }
```



Link:

<https://onlinegdb.com/Lr40x4jW3>

# Principais aspectos da aula:

```
main.c      embarcatech ::  
1 int main() {  
2     int age = 30; // Variável inteira com valor inicial 30  
3     const float pi = 3.14159; // Constante pi com valor fixo  
4     static int count = 0; // Variável estática, incrementada a cada chamada da função  
5 }
```

```
main.c      embarcatech ::  
1 #include <stdio.h>  
2 #include <math.h>  
3  
4 int main() {  
5     double a, b, c, delta;  
6  
7     printf("Digite os coeficientes da equacao do segundo grau (a, b, c): ");  
8     scanf("%lf %lf %lf", &a, &b, &c);  
9  
10    delta = pow(b, 2) - 4 * a * c;  
11  
12    printf("O valor de delta e: %.2lf\n", delta);  
13 }
```



Residência  
Tecnológica  
em Sistemas  
Embarcados

# Obrigado!

Executores:



Coordenação:



Financiadores:

