



Algoritmos e Programação I

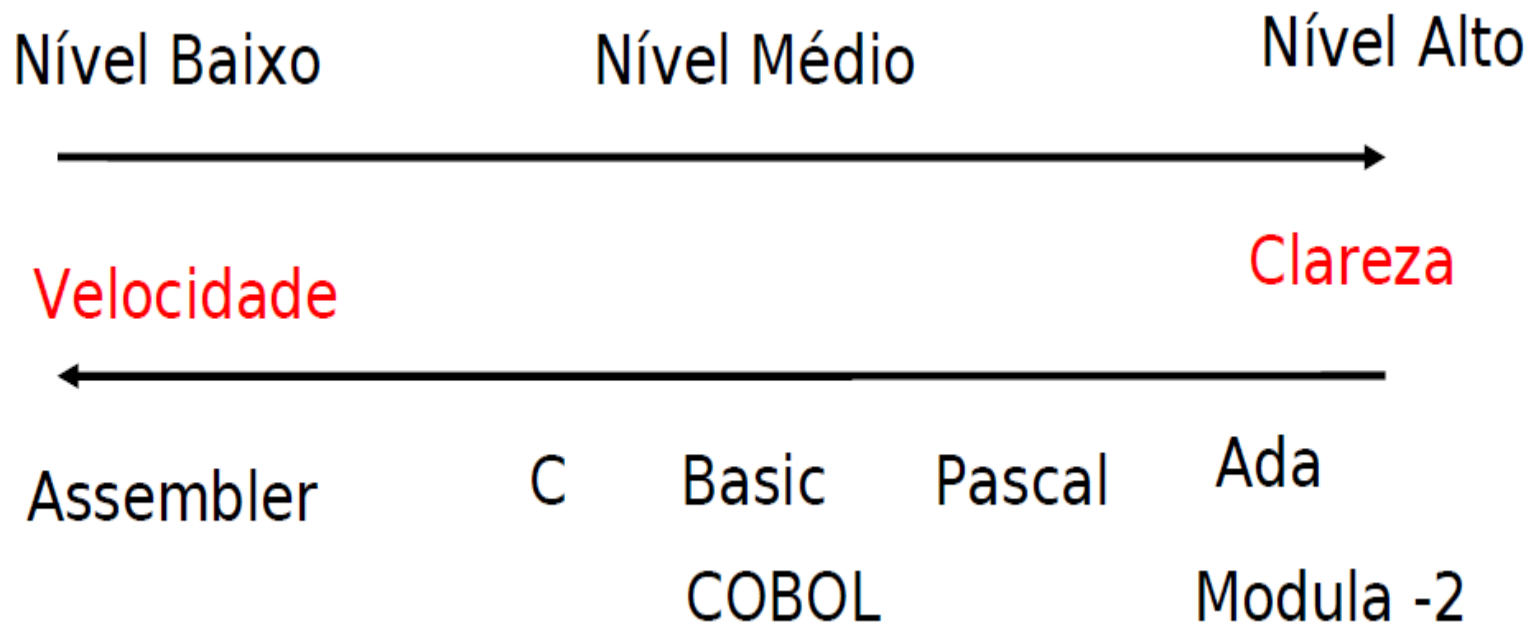


Agenda

- Construção de Algoritmo
 - Linguagem C;
 - Estrutura de um programa;
 - Atribuições;
 - Comandos do sistema;
 - Estrutura sequencial.

Linguagem C

Escrita em 1973 por Dennis Ritchie
utilizada para escrever o S.O Unix.



Linguagem C

- **Portabilidade** entre máquinas e sistemas operacionais;
- Programas **Estruturados**;
- Código **compacto** e rápido, quando comparado ao código de outras linguagem de complexidade análoga.

Estrutura de um programa

Tomando como base a linguagem C, possui a seguinte estrutura:

1. Diretivas para o pré-processador;
 2. Declaração de variáveis globais;
 3. Funções
- ```
main()
{
 • Declaração de variáveis locais da função main;
 • Comandos.
}
```

## Diretivas para o pré-processador

Incluir **biblioteca** com funções pré-definidas:

| Diretivas                              | Funções da biblioteca      |
|----------------------------------------|----------------------------|
| <code>#include &lt;stdio.h&gt;</code>  | Funções de Entrada e Saída |
| <code>#include &lt;stdlib.h&gt;</code> | Funções padrão             |
| <code>#include &lt;math.h&gt;</code>   | Funções matemáticas        |
| <code>#include &lt;system.h&gt;</code> | Funções do sistema         |

## Atribuição

- Permite que se forneça um valor a uma certa variável, onde o tipo do valor tem que ser compatível com o tipo da variável;

Identificador ← expressão

**Identificador:** É o nome da variável a qual está sendo atribuído o valor da expressão.

← : É o símbolo de atribuição;

**expressão:** pode ser aritmética, lógica ou literal de cuja avaliação é obtido o valor a ser atribuído a variável.

# Atribuição

Exemplos em Algoritmos:

|                                             |                                         |
|---------------------------------------------|-----------------------------------------|
| $K \leftarrow 1$                            | $A \leftarrow B$                        |
| $COR \leftarrow \text{"VERDE"}$             | $MEDIA \leftarrow SOMA/N$               |
| $TESTE \leftarrow \underline{\text{falso}}$ | $SIM \leftarrow X < 0 \text{ e } Y = 5$ |



## Atribuição

Exemplos em Linguagem C:

|                        |                                |
|------------------------|--------------------------------|
| $K = 1$                | $A = B$                        |
| $COR = \text{"VERDE"}$ | $MEDIA = SOMA/N$               |
| $TESTE = \text{false}$ | $SIM = (X < 0 \ \&\& \ Y = 5)$ |

## Comandos de Saída

- escreva lista\_de\_variaveis e/ou constantes;
- escreva: é uma palavra reservada da linguagem que permite a saída de dados;
- lista\_de\_variaveis: nomes das variáveis, cujos conteúdos serão mostrados ao usuário através do meio de saída ou gravados em disco. Além dos conteúdos das variáveis, o valor de um constante pode ser emitido diretamente.
- Exemplo: escreva “Media da Turma: ”, MEDIA

## Função printf

Função “printf”: – possibilita a saída de valores segundo um determinado formato

```
printf (formato, lista de constantes/variáveis/expressões...);
```

```
printf ("%d %g", 33, 5.3);
```

tem como resultado a impressão da linha: **33 5.3**

```
printf ("Inteiro = %d Real = %g", 33, 5.3);
```

com saída: Inteiro = 33 Real = 5.3

# Função printf

## Especificação de formato:

- %c** especifica um **char**
- %d** especifica um **int**
- %f** especifica um **float**
- %s** especifica uma **cadeia de caracteres**

## Comandos de Entrada

- leia lista\_de\_variaveis;
- leia: é uma palavra reservada da linguagem que permite a entrada de dados;
- lista\_de\_variaveis: nomes das variáveis, separadas por vírgulas, nas quais são armazenados os valores provenientes do dispositivo de entrada.
- Exemplo: leia NOME, NOTA

# Função scanf

**Função “scanf”:**

**– captura valores fornecidos via teclado**

```
scanf (formato, lista de endereços das variáveis...);
```

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

valor inteiro digitado pelo usuário é armazenado na variável n

## Função scanf

### Especificação de formato:

- %c** especifica um **char**
- %d** especifica um **int**
- %f** especifica um **float**
- %s** especifica uma **cadeia de caracteres**

## Função scanf

Função “scanf” (cont.):

- caracteres diferentes dos especificadores no formato servem para cercar a entrada
- espaço em branco dentro do formato faz com que sejam "pulados" eventuais brancos da entrada
- %d, %f automaticamente pulam os brancos que precederem os valores numéricos a serem capturados

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



## Estrutura Sequencial

Conjunto de **ações** que serão executadas em sequencia linear de forma **horizontal**, na ordem que forem escritas.

As ações serão seguidas por **ponto e vírgula(;)** .

```
float peso;
printf("Digite o peso:");
scanf("%f", &peso)
```

# Estrutura Sequencial

## Primeiro Programa - Exemplo

```
algoritmo "subtracao"
var
 numero1, numero2, diferenca : inteiro
inicio
 escreva("Digite o primeiro número inteiro: ")
 leia(numero1)
 escreva("Digite o primeiro número inteiro: ")
 leia(numero2)
 diferenca <- numero1 - numero2
 escreva("Resultado da diferenca = ", diferenca)
fimalgoritmo
```

# Estrutura Sequencial

## Primeiro Programa - Exemplo

```
#include <stdio.h>
main()
{
 int numero1, numero2, diferenca;
 printf ("Digite o primeiro número inteiro: ");
 scanf ("%d", &numero1);
 printf ("Digite o segundo número inteiro: ");
 scanf ("%d", &numero2);
 diferenca = numero1 - numero2;
 printf ("Resultado da diferenca = %d", diferenca);
}
```

# Estrutura Sequencial

## Primeiro Programa - Exemplo

| <u>numero1</u> | <u>numero2</u> | <u>diferenca</u> | <u>saída</u> |
|----------------|----------------|------------------|--------------|
| ????           | ????           | ????             |              |

# Estrutura Sequencial

## Primeiro Programa - Exemplo

```
printf ("Digite o primeiro número inteiro: ");
```

| <u>numero1</u> | <u>numero2</u> | <u>diferenca</u> | <u>saída</u>          |
|----------------|----------------|------------------|-----------------------|
| ????           | ????           | ????             | Digite o primeiro ... |

# Estrutura Sequencial

## Primeiro Programa - Exemplo

```
scanf ("%d", &numero1);
```

| <u>numero1</u> | <u>numero2</u> | <u>diferenca</u> | <u>saída</u>          |
|----------------|----------------|------------------|-----------------------|
| ????           | ????           | ????             | Digite o primeiro ... |
| 171            | ????           | ????             |                       |

# Estrutura Sequencial

## Primeiro Programa - Exemplo

```
printf ("Digite o segundo número inteiro: ");
```

| <u>numero1</u> | <u>numero2</u> | <u>diferenca</u> | <u>saída</u>          |
|----------------|----------------|------------------|-----------------------|
| ????           | ????           | ????             | Digite o primeiro ... |
| 171            | ????           | ????             | Digite o segundo ...  |

# Estrutura Sequencial

## Primeiro Programa - Exemplo

```
scanf ("%d", &numero2);
```

| <u>numero1</u> | <u>numero2</u> | <u>diferenca</u> | <u>saída</u>          |
|----------------|----------------|------------------|-----------------------|
| ????           | ????           | ????             | Digite o primeiro ... |
| 171            | ????           | ????             | Digite o segundo ...  |
| 171            | 10             | ????             |                       |



# Estrutura Sequencial

## Primeiro Programa - Exemplo

```
diferenca = numero1 - numero2;
```

| <u>numero1</u> | <u>numero2</u> | <u>diferenca</u> | <u>saída</u>          |
|----------------|----------------|------------------|-----------------------|
| ????           | ????           | ????             | Digite o primeiro ... |
| 171            | ????           | ????             | Digite o segundo ...  |
| 171            | 10             | 161              |                       |

# Estrutura Sequencial

## Primeiro Programa - Exemplo

```
printf ("Resultado da diferenca = %d", diferenca);
```

| <u>numero1</u> | <u>numero2</u> | <u>diferenca</u> | <u>saída</u>          |
|----------------|----------------|------------------|-----------------------|
| ????           | ????           | ????             | Digite o primeiro ... |
| 171            | ????           | ????             | Digite o segundo ...  |
| 171            | 10             | 161              | Resultado ... é 161   |

## Estrutura Sequencial

### Exercício

1. Ler dois números e imprimir a soma. Antes do resultado, deverá aparecer a mensagem, SOMA:.
2. Ler dois números e imprimir a média aritmética com a mensagem, MÉDIA: antes do resultado.
3. Entrar com dois números e imprimir a seguinte saída:  
Dividendo:  
Divisor:  
Quociente:  
Resto:

## Bibliografia

FORBELLONE, A. L. V. e EBERSPACHER, H. F. **Lógica de Programação – A Construção de Algoritmos e Estrutura de Dados**. 3ª Edição. Prentice Hall. 2005. Editora, 1999

FARRER, H. **Algoritmos Estruturados**. 3 ed. Rio de Janeiro: LTC

CORMEN, T. H. e LEISERSON, C. E. **Algoritmos - teoria e prática**. 1. ed. Rio de Janeiro: Campus, 2002