

Lógica de Programação

Aula 01:
Algoritmos e lógica de programação.

Lógica de Programação

O que são algoritmos?

Lógica de Programação

Algoritmos

- **Definição**

“Sequência **finita** de instruções **bem definidas e excluídas de dúvidas ou incertezas**, as quais podem ser **executadas** em um **período de tempo definido** e com uma **quantidade de esforço definida**.”

Lógica de Programação

Algoritmos

- **Definição**
 - ✓ Não representa, necessariamente, um programa de computador mas sim os passos necessários para a realização de uma tarefa;
 - ✓ Ex: Receita do bolo;

Lógica de Programação

Algoritmos

- **Receita do bolo**

**Farinha de trigo
Ovos
Leite
Fermento
Açúcar
Manteiga**



Receita do Bolo



Lógica de Programação

Algoritmos

- Receita do bolo

Algoritmo

Farinha de trigo
Ovos
Leite
Fermento
Açúcar
Manteiga

Receita do Bolo



Ingredientes

Resultado

Lógica de Programação

Algoritmos

- **Receita do bolo - Algoritmo**
 1. Bater duas claras;
 2. Adicionar duas gemas;
 3. Adicionar um xícara de açúcar;
 4. Adicionar duas colheres de manteiga;
 5. Adicionar farinha;
 6. Adicionar fermento;
 7. Colocar tudo em uma forma;
 8. Levar ao forno em baixa temperatura.

Lógica de Programação

Algoritmos

- **Fases para elaboração de soluções para problemas:**
 - ✓ Análise do problema;
 - ✓ Concepção de um algoritmo;
 - ✓ Obtenção da solução.



Lógica de Programação

Algoritmos

- **Passos para a construção de um algoritmo:**
 - ✓ Quais os passos necessários para a construção de um algoritmo?

Lógica de Programação

Algoritmos

- **Passos para a construção de um algoritmo:**
 - ✓ Definir o problema;
 - ✓ Compreender o problema;
 - ✓ Realizar um levantamento da situação;
 - ✓ Identificar os dados de entrada;
 - ✓ Identificar os dados de saída;

Lógica de Programação

Algoritmos

- **Passos para a construção de um algoritmo:**
 - ✓ Elaborar o que deve ocorrer com as entradas para se obter a saída desejada:
 - Usar a estratégia de dividir o problema;
 - Seguir as regras e limitações;
 - Identificar todas as ações a se realizar
 - Eliminar ambiguidades.

Lógica de Programação

Algoritmos

- **Passos para a construção de um algoritmo:**
 - ✓ Escolher uma linguagem para escrever a solução do problema;
 - ✓ Construir o algoritmo;
 - ✓ Testar o algoritmo;
 - ✓ Executar o algoritmo;
 - ✓ Verificar se o problema foi resolvido.

Lógica de Programação

O que é lógica de programação?

Lógica de Programação

Algoritmos

- **Lógica de Programação**
 - ✓ Qualquer problema a ser resolvido necessita de uma solução **lógica** e funcional;
 - ✓ Portanto é necessário aprender a elaborar soluções lógicas e funcionais;
 - ✓ Tais soluções podem ser previstas pela **lógica de programação**.

Lógica de Programação

Algoritmos

- **Lógica de Programação**
 - ✓ Consiste em analisar e compreender claramente os diversos passos necessários na elaboração de uma solução para um determinado **problema**, garantindo que o resultado final seja válido e satisfatório.

Lógica de Programação

Algoritmos

- **Lógica de Programação**
 - ✓ Pensar para elaborar uma solução;
 - ✓ Transformação do conhecimento previamente obtido para criar algo novo, ou seja, a solução do problema proposto;
 - ✓ Mudar de $A = A, A, A \dots A$ para $A = B + C$.

Lógica de Programação

Algoritmos

- **Varias soluções para um problema:**
 - ✓ Diferentes algoritmos podem realizar a mesma tarefa usando um conjunto diferenciado de instruções em mais ou menos tempo, espaço ou esforço do que outros;

Lógica de Programação

Algoritmos

- **Varias soluções para um problema:**
 - ✓ Um algoritmo para se vestir pode especificar que você vista primeiro as meias e os sapatos antes de vestir a calça enquanto outro algoritmo especifica que você deve primeiro vestir a calça e depois as meias e os sapatos.

Lógica de Programação

Algoritmos

- **Características importantes em um algoritmo quanto a lógica:**
 - ✓ Finitude;
 - ✓ Definição;
 - ✓ Entradas;
 - ✓ Saídas;
 - ✓ Efetividade.

Lógica de Programação

Algoritmos

- **Característica: Finitude**
 - ✓ Um algoritmo deve sempre terminar após um número finito de passos.

Lógica de Programação

Algoritmos

- **Característica: Definição**
 - ✓ Cada passo de um algoritmo deve ser precisamente **definido**;
 - ✓ As ações devem ser definidas rigorosamente e excluídas de dúvidas ou incertezas com relação a sua interpretação.

Lógica de Programação

Algoritmos

- **Característica: Entradas**
 - ✓ Um algoritmo deve ter uma ou mais entradas fornecidas antes do algoritmo iniciar.

Lógica de Programação

Algoritmos

- **Característica: Saídas**
 - ✓ Um algoritmo deve ter uma ou mais saídas, que tenham uma relação específica com as entradas e com o problema a ser resolvido.

Lógica de Programação

Algoritmos

- **Característica: Efetividade**
 - ✓ Um algoritmo deve ser efetivo, ou seja, todas as operações devem ser suficientemente capazes de serem executadas com precisão em um tempo definido.

Lógica de Programação

**Formas de representação de algoritmos
que facilitam a aplicação da lógica.**

Lógica de Programação

Algoritmos

- **Formas de representação de algoritmos:**

Para elaborar a solução de um problema qualquer é necessário, em primeiro lugar, encontrar uma maneira de representa-lo de uma forma clara e precisa;

Lógica de Programação

Algoritmos

- **Formas de representação de algoritmos:**

As formas mais comuns de representação de algoritmos são as seguintes:

- ✓ Linguagem Natural;
- ✓ Diagrama de blocos;
- ✓ Pseudo-linguagem.

Lógica de Programação

Algoritmos

- **Linguagem Natural:**

Os algoritmos são expressos diretamente em linguagem natural.

Lógica de Programação

Algoritmos

- **Linguagem Natural:**

Algoritmo Escovar-os-dentes
colocar creme dental na escova
molhar a escova
escovar os dentes
enxaguar a boca
enxugar a boca
Fim-do-algoritmo

Lógica de Programação

Algoritmos

- **Exercicio 01:**

- ✓ Temos duas vasilhas com capacidades de 9 e 4 litros respectivamente, não há nenhuma marcação de metade ou um terço nas vasilhas. Definir uma sequencia de passos usando as vasilhas de 9 e 4 litros para encher uma terceira vasilha de medida desconhecida com exatamente seis litros de água.

Lógica de Programação

Algoritmos

- **Possível solução:**
 1. Encher a vasilha de 9 litros;
 2. Usando a vasilha de 9 litros, encher a vasilha de 4 litros;

Lógica de Programação

Algoritmos

- **Possível solução:**
 3. Despejar o que sobrou na vasilha de 9 litros (5 litros) na terceira vasilha, observando que falta um litro para completar os seis litros;
 4. Esvaziar a vasilha de 4 litros;
 5. Tornar a encher a vasilha de 9 litros;

Lógica de Programação

Algoritmos

- **Possível solução:**

6. Usando a vasilha de 9 litros encher a vasilha de 4 litros;
7. Esvaziar a vasilha de 4 litros;
8. Usando o que restou na vasilha de 9 litros (5 litros), encher novamente a vasilha de quatro litros;

Lógica de Programação

Algoritmos

- **Possível solução:**

9. Despejar o que sobrou na vasilha de 9 litros (1 litro) na terceira vasilha, que agora tem 6 litros.

Lógica de Programação

Algoritmos

- **Diagrama de bloco:**
 - ✓ Representação gráfica de um processo ou modelo de um sistema complexo;
 - ✓ As relações entre cada subsistema e o fluxo de informação são descritas através de figuras geométricas e ligações;

Lógica de Programação

Algoritmos

- **Diagrama de bloco:**
 - ✓ É a representação gráfica da solução de um problema;
 - ✓ Um exemplo de diagramas de blocos são os **fluxogramas**, usados na elaboração de programas de computadores.

Lógica de Programação

Algoritmos

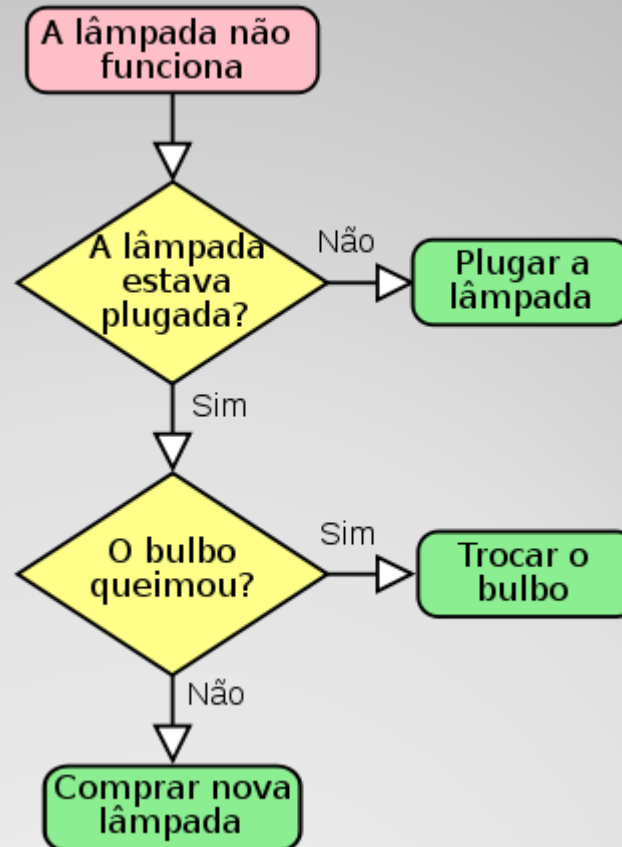
- **Fluxograma:**

Representação gráfica que emprega formas geométricas padronizadas para indicar as diversas ações e as decisões que devem ser executadas para resolver um problema computacional.

Lógica de Programação

Algoritmos

- Fluxograma:



Lógica de Programação

Algoritmos

- **Símbolos básicos do fluxograma:**

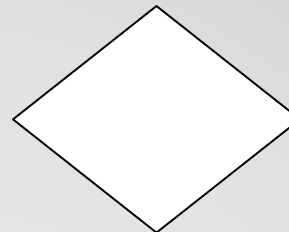
Início ou Fim



Saída



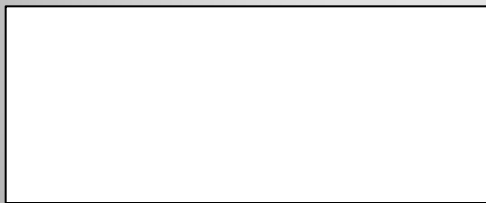
Decisão



Documentos



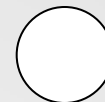
Processamento



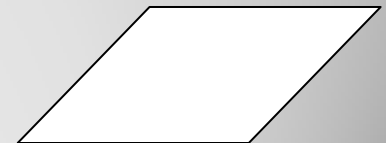
Entrada



Continuação



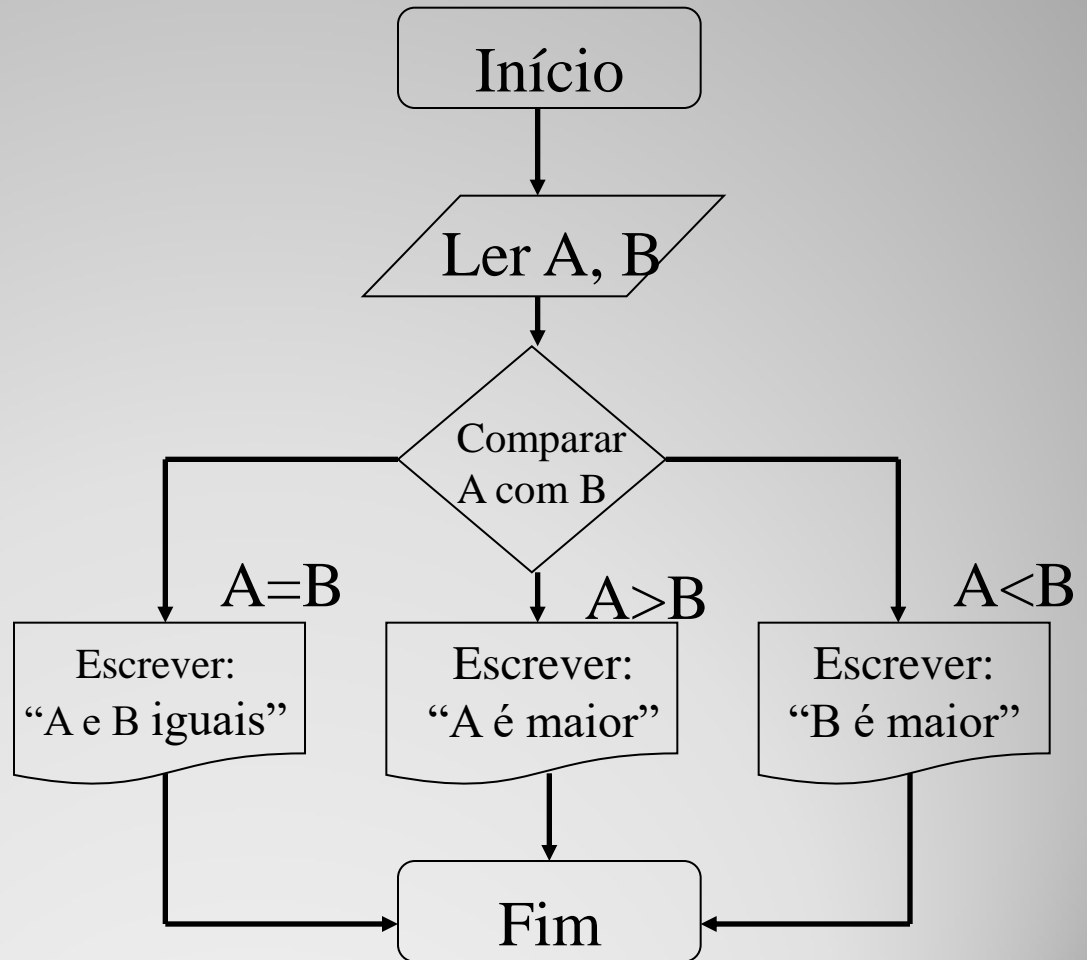
Dados



Lógica de Programação

Algoritmos

- **Exemplo:**
 - ✓ Achar o maior de dois números A e B.



Lógica de Programação

Algoritmos

- **Pseudo-linguagem:**
 - ✓ Linguagem intermediária entre a linguagem natural e uma linguagem de programação para descrever algoritmos;
 - ✓ Exemplo: Portugol, Algol, etc;

Lógica de Programação

Algoritmos

- **Exemplo da Pseudo-Linguagem Portugol:**

```
Programa teste  
início  
    escreva('teste')  
fim
```

Lógica de Programação

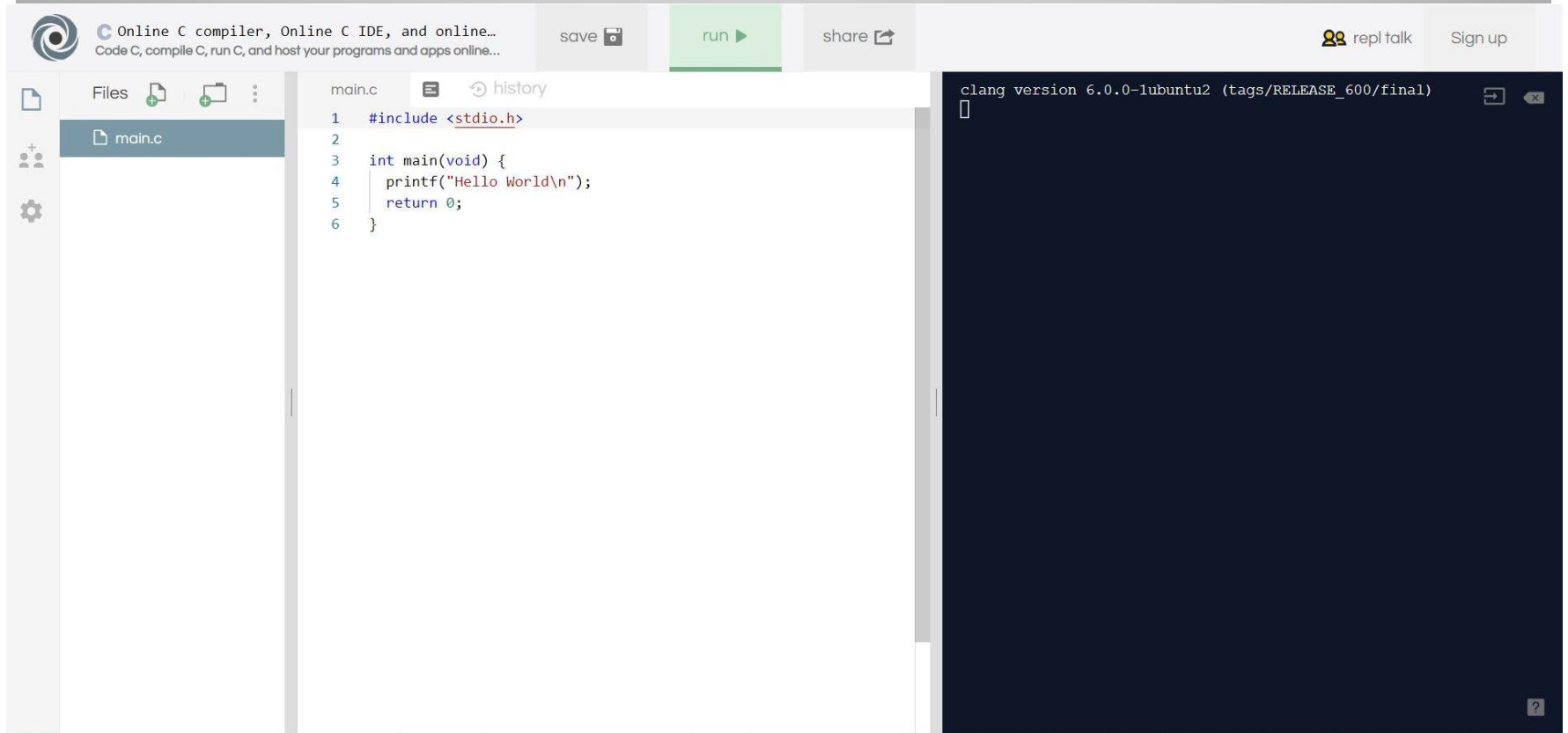
Linguagem C.

Lógica de Programação

O ambiente de programação

Lógica de Programação

<https://repl.it/languages/c>



The screenshot displays the Repl.it online C compiler interface. At the top, a navigation bar includes the Repl.it logo, the text "Online C compiler, Online C IDE, and online... Code C, compile C, run C, and host your programs and apps online...", and buttons for "save", "run", and "share". On the right side of the navigation bar, there are links for "repl talk" and "Sign up".

The main workspace is divided into three panels:

- Files Panel (Left):** Shows a file named "main.c".
- Code Editor (Center):** Displays the C code for "main.c":

```
1 #include <stdio.h>
2
3 int main(void) {
4     printf("Hello World\n");
5     return 0;
6 }
```
- Terminal (Right):** Shows the output of the compilation and execution:

```
clang version 6.0.0-1ubuntu2 (tags/RELEASE_600/final)
[]
```

Lógica de Programação

Meu primeiro programa em Linguagem C

✓ Passos para criar um programa

- Abrir o site;
- Escrever o código;
- Clicar em **execute (run)**;
- Clicar na área escura;
- Interagir com o programa.

Lógica de Programação

A Linguagem C

Lógica de Programação

Linguagem C

✓ A função Main:

- Necessária em todos os programas autoexecutáveis escritos em linguagem C;
- Responsável pelo controle de execução **principal** do programa.

Lógica de Programação

Linguagem C

- ✓ **Tipos de dados: LINGUAGEM C**
 - **Tipos Básicos:**
 - ✓ **char: Character:** O valor armazenado é um caractere;
 - ✓ **int:** Tipo padrão representado por um número inteiro;

Lógica de Programação

Linguagem C

- ✓ **Tipos de dados: LINGUAGEM C**
 - **Tipos Básicos:**
 - ✓ **float:** Número em ponto flutuante de precisão simples (números reais);
 - ✓ **double:** Número em ponto flutuante de precisão dupla;

Lógica de Programação

Linguagem C

- ✓ **Tipos de dados: LINGUAGEM C**
 - **Tipos Básicos:**
 - ✓ **void:** Serve para indicar que um resultado não tem um tipo definido;
 - ✓ Um tipo vazio pode posteriormente ser modificado para um dos tipos anteriores.

Lógica de Programação

Linguagem C

- ✓ **Tipos de dados: LINGUAGEM C**
 - **Tipos Básicos:**
 - ✓ **E o tipo lógico?**
 - ✓ Tudo que é igual a 0 é falso;
 - ✓ Tudo que é diferente de 0 é verdadeiro.

Lógica de Programação

Linguagem C

✓ **Modificadores dos Tipos Básicos:**

- São palavras que alteram o tamanho do conjunto de valores que um tipo pode representar.
- Exemplo: um modificador que permite que possam ser armazenados números inteiros maiores ou que obriga que só números sem sinal possam ser armazenados.

TIPO DE DADO	NUMERO DE BITS	FORMATO PARA LEITURA COM scanf	INTERVALO	
			INÍCIO	FIM
char	8	%C	-128	127
unsigned char	8	%C	0	255
signed char	8	%C	-128	127
int	16	%i	-32.768	32.767
unsigned int	16	%u	0	65.535
signed int	16	%i	-32.768	32.767
short int	16	%hi	-32.768	32.767
unsigned short int	16	%hu	0	65.535
signed short int	16	%hi	-32.768	32.767
long int	32	%li	-2.147.483.648	2.147.483.647
signed long int	32	%li	-2.147.483.648	2.147.483.647
unsigned long int	32	%lu	0	4.294.967.295
float	32	%f	3,40E-38	3.4E+38
double	64	%lf	#NUM!	1,7E+308
long double	80	%Lf	3,4E-4932	3,4E+4932

Lógica de Programação

Linguagem C

✓ Declarações:

- Não é mais necessário o uso da palavra **declaracoes**;
- As variáveis podem ser declaradas em qualquer lugar da função **main()**, mas sempre que possível declare as variáveis no início do código fonte.

Lógica de Programação

Linguagem C

✓ Variáveis:

- Pequeno espaço de memória reservado para receber um dado ou informação que varia no decorrer do tempo.
- Exemplos:
 - `int a;`
 - `float b,c;`
 - `char a`
 - `char b[10];`
 - `int b[10][10];`

Lógica de Programação

Linguagem C

✓ Nomes das variaveis:

- Regras básicas para nomear variáveis:
 - Só pode conter letras e dígitos;
 - O "_" é aceito como uma letra;
 - O primeiro caractere tem que ser letra;
 - Maiúsculas/minúsculas são diferentes;

Lógica de Programação

Linguagem C

✓ Nomes das variaveis:

- Regras básicas para nomear variáveis:
 - Palavras reservadas não podem ser usadas como nome de variáveis;
 - Deve-se escolher nomes que indiquem a função da variável.

Lógica de Programação

Linguagem C

✓ Constantes:

- Pequeno espaço de memória reservado para receber um dado ou informação que **não** varia no decorrer do tempo.
- Exemplos:

➤ `const int TAM=10;` ➤ `int a[TAM];`

Lógica de Programação

Linguagem C

✓ Ponteiros:

- O ponteiro é uma variável que guarda o endereço de uma outra variável.
- A declaração de ponteiros é feita da seguinte forma:

➤ `int *a;` ➤ `*a=10;`

Lógica de Programação

Linguagem C

✓ Comandos:

- Definem entradas de dados;
- Processam as informações;
- Definem saídas de dados.

Lógica de Programação

Linguagem C

✓ Comandos:

- Definem entradas de dados;
- Processam as informações;
- Definem saídas de dados.

Lógica de Programação

Linguagem C

✓ Comando **printf (escreva)**:

- Usado para exibir mensagens e valores de variáveis na tela;
- Exemplos:
 - `printf("Teste");`
 - `printf(A);`
 - `printf("\n Teste",A);`

Lógica de Programação

Caracteres especiais

Código	Significado
<code>\b</code>	Retrocesso (BS)
<code>\f</code>	Alimentação de formulário (FF)
<code>\n</code>	Nova linha (LF)
<code>\r</code>	Retorno de carro (CR)
<code>\t</code>	Tabulação horizontal (HT)
<code>\"</code>	Aspas duplas
<code>\'</code>	Aspas simples
<code>\0</code>	Nulo
<code>\\</code>	Barra invertida
<code>\v</code>	Tabulação vertical
<code>\a</code>	Alerta (beep)
<code>\N</code>	Constante octal (onde N é uma constante octal)
<code>\xN</code>	Constante hexadecimal (onde N é uma constante hexadecimal)

Lógica de Programação

Linguagem C

✓ Comando **scanf (leia)**:

Forma de escrita do comando scanf:

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

Lógica de Programação

Linguagem C

✓ Comando **scanf (leia):**

- Usado para solicitar valores aos usuários que serão enviados direto as variáveis;
- Exemplos:
 - `scanf("%d",&numero_inteiro);`
 - `scanf("%s",&nome);`

Lógica de Programação

Linguagem C

- ✓ **Expressão de controle:**
 - É composta pelos códigos de formatação que indica o tipo do dado a ser lido;
 - Os **argumentos** deve ser constituídos pelos **endereços** das variáveis, por isto usa-se o operador de endereço “&” antes do nome da variável que se deseja atribuir um valor;

Lógica de Programação

Linguagem C

- ✓ **Expressão de controle:**
 - É necessário que o tipo de dado do argumento seja compatível com os códigos de formatação;
 - Também são usados no comando **printf**.

Código de formatação	Descrição
%c	Caracteres simples
%d	Inteiros decimais com sinal
%I	Inteiros decimais com sinal
%e	Notação científica (e minúsculo)
%E	Notação científica (E maiúsculo)
%f	Ponto flutuante decimal
%g	Usa %e ou %f (qual for mais curto)
%G	Isa %E ou %F (qual for mais curto)
%o	Octal sem sinal
%s	Cadeia de caracteres
%u	Inteiros decimais sem sinal
%x	Hexadecimal sem sinal (letras minúsculas)
%X	Hexadecimal sem sinal (letras maiúsculas)
%%	Escreve o símbolo de porcentagem (%)

Lógica de Programação

Exemplos de `printf` e `scanf`

[illegible]

Lógica de Programação

```
#include<stdio.h> /* para usar a função scanf */

void main()
{
    char nome[30];
    int idade;

    printf("Digite seu primeiro nome:");
    scanf("%s",&nome); /*usa-se %s pq se trata de uma string de caracteres*/
    printf("\nDigite a sua idade:");
    scanf("%d", &idade); /*usa-se %d pq se trata de um número inteiro*/

    printf("\n%s tem %d anos!!!",nome,idade);
}
```


Lógica de Programação

Linguagem C

✓ Comandos:

- Definem entradas de dados;
- **Processam as informações;**
- Definem saídas de dados.

Lógica de Programação

Linguagem C

✓ Atribuição:

- Comando que define ou re-define o valor armazenado em uma variável.
- Exemplo:

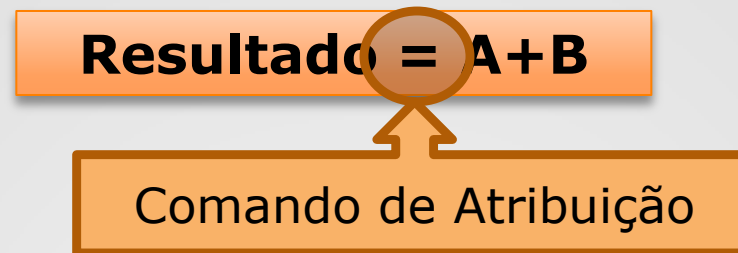
Resultado = A+B

Lógica de Programação

Linguagem C

✓ Atribuição:

- Comando que define ou re-define o valor armazenado em uma variável.
- Exemplo:



Lógica de Programação

Linguagem C

✓ Operadores:

- Aritméticos;
- Lógicos;
- Relacionais.

Lógica de Programação

Linguagem C

✓ Operadores Aritméticos:

Operação	Operador	Exemplo	Resultado
Soma	+	5 + 5	10
Subtração	-	5 - 4	1
Multiplicação	*	10 * 2	20
Divisão	/	10 / 2	5
Modulo	%	7 % 3	1
Incremento	++	A++	A+1
Decremento	--	B--	B-1

Lógica de Programação

Linguagem C

✓ Operadores Lógicos:

Operação	Operador	Exemplo	Resultado
E	&&	1 && 0	falso
E	&&	1 && 1	verdadeiro
OU	 	1 0	verdadeiro
OU	 	1 1	verdadeiro
OU	 	0 0	falso
NÃO	!	! 0	verdadeiro
NÃO	!	! 1	falso

Lógica de Programação

Linguagem C

✓ Operadores Relacionais:

Operação	Operador	Exemplo	Resultado
Maior	>	5 > 4	verdadeiro
Maior ou igual	>=	4 >= 4	verdadeiro
Menor	<	10 < 4	falso
Menor ou igual	<=	10 <= 4	verdadeiro
Igual	==	4 == 10	falso
Diferente	!=	3 != 4	verdadeiro

Lógica de Programação

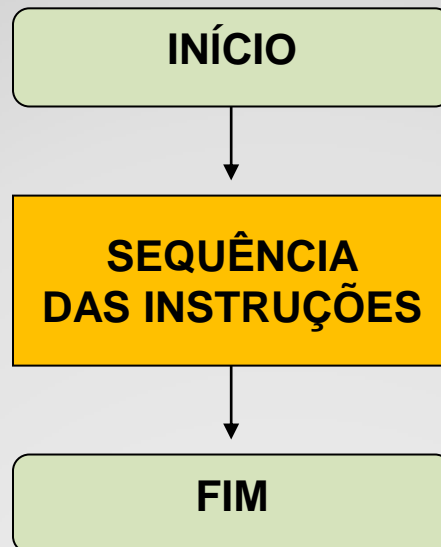
Linguagem C

- ✓ **Tipos de estrutura de controle:**
 - Estrutura de Controle Sequencial;
 - Estrutura de Controle Condicional;
 - Estrutura de Controle de Repetição.

Lógica de Programação

Linguagem C

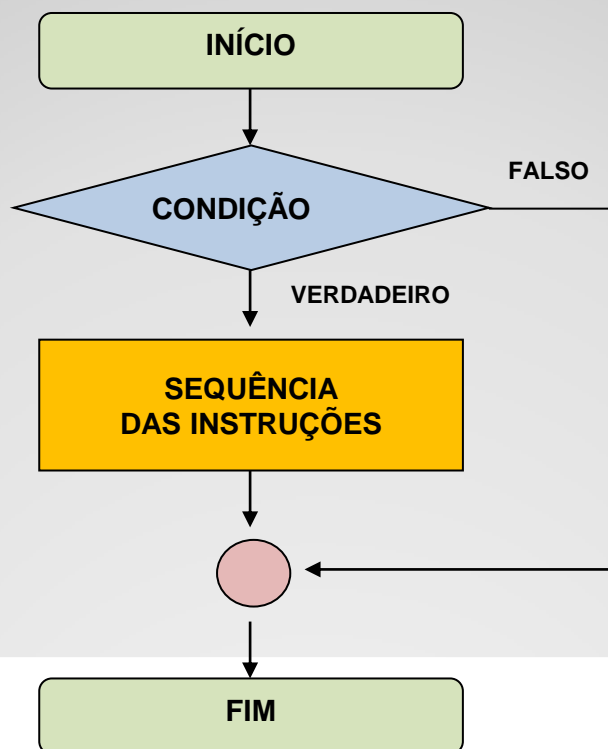
✓ Estrutura de Controle **Sequencial**



Lógica de Programação

Linguagem C

✓ Estrutura de Controle **Condicional**



Lógica de Programação

Linguagem C

✓ Estrutura de Controle Repetitivo

