

# Professor José de Assis

## Introdução à Lógica de Programação

De forma bem simples podemos definir a programação como uma sequência de instruções e passos lógicos com o objetivo de realizar alguma tarefa.

### Linguagem de programação

Uma linguagem de programação é uma espécie de idioma artificial desenvolvido para expressar instruções e operações que podem ser executadas por máquinas como por exemplo o computador. A linguagem tem uma sintaxe própria que é formada por um conjunto de símbolos e regras que definem sua estrutura e o significado de seus elementos e expressões.

### Linguagem de máquina

São códigos diretamente interpretados pelo microprocessador de um computador ou um microcontrolador. A linguagem de máquina trabalha com dois níveis de voltagem. Tais níveis, por abstração, se simbolizam com o zero (0) e o um (1), também conhecido como sistema binário.

### Linguagem de alto nível

Linguagem de alto nível é como conhecemos as linguagens de programação. As linguagens de programação estão longe da linguagem de máquina e mais próximas à linguagem humana. Desse modo, as linguagens de alto nível não estão diretamente relacionadas à arquitetura do computador. O programador de uma linguagem de alto nível não precisa conhecer características do processador, como instruções e registradores por exemplo para desenvolver os programas. Essas características são abstraídas na linguagem de alto nível. Embora trabalhe próxima ao hardware, a linguagem C é considerada uma linguagem de alto nível.

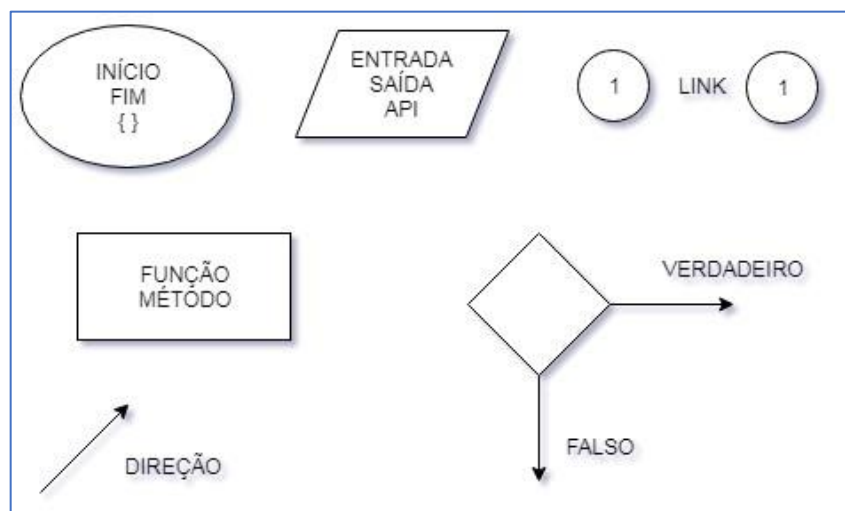
### Algoritmo

O conceito de algoritmo é frequentemente comparado a uma receita culinária. Um algoritmo não representa, necessariamente um programa de computador, e sim os passos necessários para realizar uma tarefa.

## Fluxograma

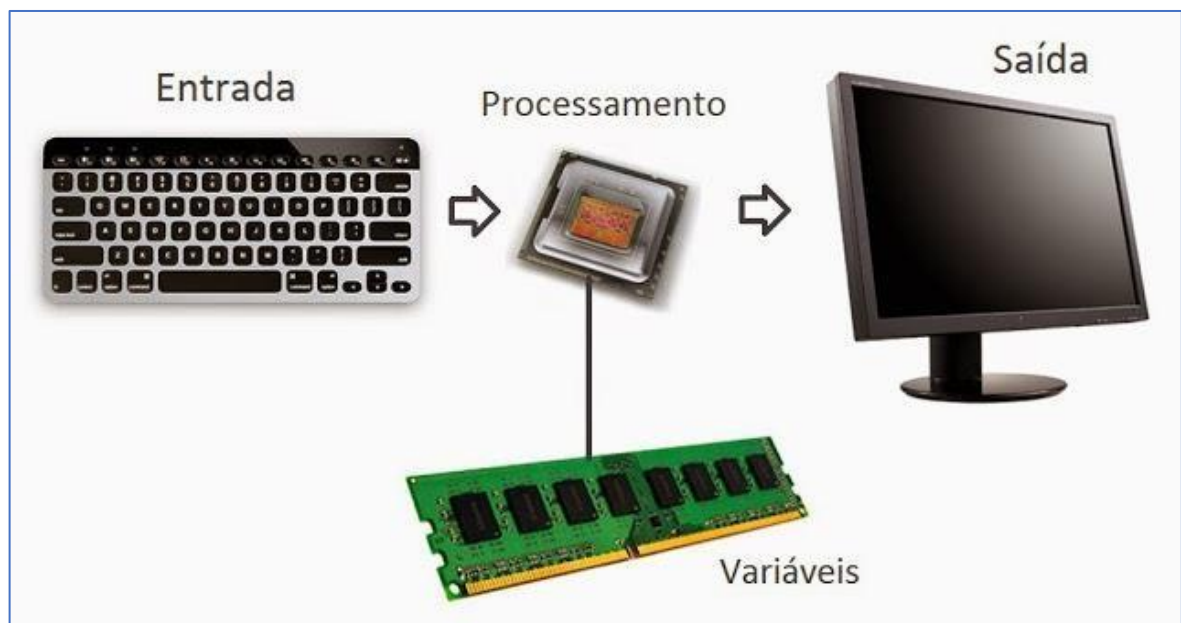
É a representação gráfica do projeto. Costumo fazer uma comparação com o electricista que ao analisar uma planta eléctrica consegue entender o projeto. Usamos o fluxograma para validar requisitos e para apresentar um projeto a diferentes desenvolvedores.

Principais símbolos:



*Figura 1 - Fluxograma - Professor José de Assis*

## Como funciona o computador:



*Figura 2 - Funcionamento do computador - Professor José de Assis*

## Variáveis

Na programação utilizamos as variáveis para armazenar dados na memória. Estes dados podem ser alterados de acordo com o tempo, como por exemplo o armazenamento do valor de um sensor de temperatura.

### Tipos de dados

Basicamente temos 3 tipos de dados que podem ser armazenados nas variáveis:

- String (consiste em letras e outros caracteres)
- Numérico (lida com números inteiros e não inteiros)
- Booleano (verdadeiro ou falso)

### Tipos de variáveis usadas na Linguagem C

Tipo	Formato	Tipo de dado	Tamanho(bytes)	Escala
<b>char</b>	%c	caractere	1	-128 a 127
<b>int</b>	%d	inteiro	2	-32768 a 32767
<b>float</b>	%f	real	4	3.4E-38 a 3.4E+38
<b>double</b>	%lf	real	8	1.7E-308 a 1.7E+308
<b>char[ ]</b>	%s	string		

### Regras para nomear variáveis

1. Todos os nomes diferenciam letras maiúsculas de minúsculas.
2. O nome deve começar com uma letra.
3. Não usar caracteres especiais nem deixar espaços.
4. Não usar palavras chaves ou reservadas da linguagem de programação.
5. Definir nomes de variáveis que descrevam o tipo de informação que serão armazenadas (código limpo).

Observação: Se o nome da variável for composto por mais de uma palavra, use uma letra maiúscula para a primeira letra da segunda palavra (notação do camelo).

Exemplo: `mediaFinal` ao invés de `media final`

## Operadores Aritméticos

São usados junto com as variáveis para processar as informações.

Operador	Propósito	Exemplo	Resultado
=	Atribuir um valor	x = 10	x = 10
+	Somar	x = 10 + 5	x = 15
-	Subtrair	x = 10 - 5	x = 5
*	Multiplicar	x = 10 * 5	x = 50
/	Dividir	x = 10 / 5	x = 2
%	Obter o resto da divisão	x = 10 % 3	x = 1
++	Adicionar 1 ao valor	x = 10; x++	x = 11
--	Subtrair 1 do valor	x = 10; x --	x = 9

## Operadores de comparação

Comparador	Nome
>	Maior que
>=	Maior ou igual a
<	Menor que
<=	Menor ou igual a
==	Igual a
!=	Diferente de

## Dicas para iniciar no mundo da programação:

- Entenda o problema (não adianta sentar-se na frente do computador se não entender o problema)
- Desenvolva pensando no usuário final (pense no usuário que nunca viu um computador na vida)
- Simplifique o código (use o mínimo possível de variáveis, expressões, estruturas etc.)