

[◀ VOLTAR](#)

# Conceitos e princípios da lógica de programação

Rever ou apresentar para quem não conhece os princípios da lógica de programação utilizada por todas as linguagens de programação.

## NESTE TÓPICO

- > DISPOSITIVOS DE ENTRADA E SAÍDA
- > UNIDADE CENTRAL DE PROCESSAMENTO
- > MEMÓRIA PRINCIPAL [Marcar tópico](#)
- > ALGORITMO
- > LINGUAGEM DE

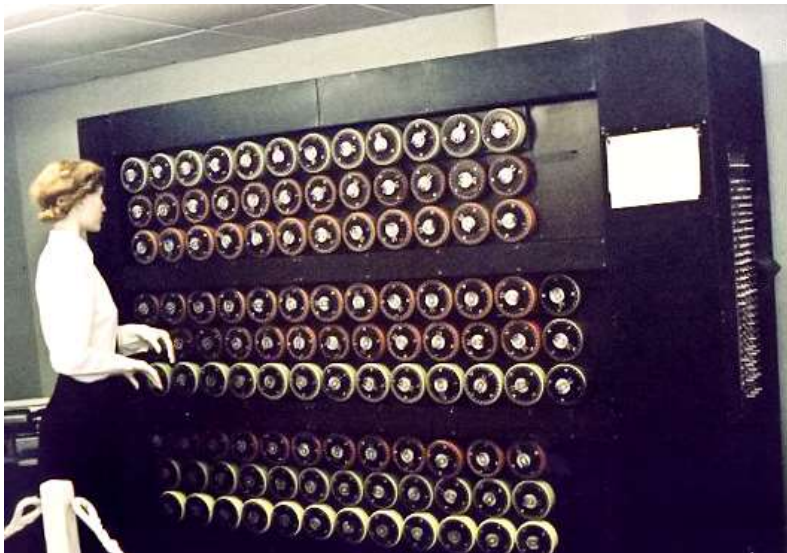


Olá alunos,

Antes de iniciarmos a programação, precisamos entender o que é lógica. É utilizar o raciocínio de forma ordenada para resolver um problema. No nosso caso, na área de TI, é necessário empregarmos a lógica para resolver problemas do mundo real, instruindo um computador, ou seja, temos que pegar o problema e colocá-lo em um dispositivo que tem uma arquitetura e uma organização de hardware para processá-lo.

A partir da década de 40, Allan Turing e Von Neumann, iniciaram a criação de uma máquina (sugiro assistirem o filme: “O jogo da imitação”) com a seguinte organização: **Memória Principal** (conhecida como memória RAM), **Unidade Central de Processamento** (conhecida como CPU ou Processador) e dispositivos de **Entrada e Saída de dados**, do original em inglês: Input e Output (I/O).

Veja a figura abaixo:



O computador de Allan M Turing

Portanto, podemos considerar o seguinte fluxo da área de TI:

**ENTRADA** ———> **PROCESSAMENTO** ———> **SAÍDA**

Onde:

**ENTRADA:** Entrada de dados: o insumo, a matéria prima da área de TI.

**PROCESSAMENTO:** Processar dados para resolver o problema.

**SAÍDA:** Dados processados, a informação (o produto acabado da área de TI).  
Daí temos a ciência da informação que é a informática.

Em relação ao **HARDWARE**, que é a parte física da computação, temos:

## DISPOSITIVOS DE ENTRADA E SAÍDA

Fazem a comunicação entre o homem e a máquina.

Os dados podem vir de diversos dispositivos de **ENTRADA**. Tais como: teclado, leitor óptico ou de um hd (hard disk) que possui dados armazenados como um banco de dados.

Os dados processados ou a informação podem ser mostrados através de diversos dispositivos de **SAÍDA** tais como: a impressora, a tela de um monitor ou a tela de um celular, etc.

## UNIDADE CENTRAL DE PROCESSAMENTO

Este dispositivo é que irá fazer o trabalho “braçal”, receber as instruções e os dados em linguagem de máquina (linguagem binária) e processá-los. Possui basicamente uma Unidade de Controle (**UC**) que interpreta as instruções, uma Unidade Lógico e Aritmética (**ULA**) que processará as instruções e os dados e uma área de armazenamento conhecida como registradores.

## MEMÓRIA PRINCIPAL

Armazena os dados e instruções que serão processados pela CPU, Von Neumann e Allan Turing criaram o conceito de “programa armazenado”, ou seja, todo o processo deve ser armazenado na memória principal para ser executado pela CPU. O processo é composto de um conjunto de instruções finito (sempre haverá uma primeira e uma última linha de instrução), conhecido como: programa, software, aplicativo, e os dados que ficam em **VARIÁVEIS**, lembrando que a memória principal é um dispositivo volátil: o armazenamento é temporário, estes dados só ficam na memória principal enquanto as linhas de instruções estão sendo processadas, mas os dados que precisam ficar de forma permanente é necessário ser armazenados em um **BANCO DE DADOS**, em dispositivos tipo hd (hard disk).

Veja na figura abaixo a simulação da alocação de variáveis na memória principal:



Variáveis na Memória Principal

Em relação ao **SOFTWARE**, que é a parte lógica, temos o trabalho do programador: criar um conjunto de instruções para agir no hardware. Para isso, é necessário utilizar uma sequência lógica para resolver o problema, portanto, o software é resultado de um trabalho intelectual.

É necessário utilizarmos técnicas para resolver o problema de forma computacional:

- Entender o problema, identificar os dados de entrada.
- Desenvolver o **algoritmo** (linguagem descritiva) ou utilizar o fluxograma (linguagem visual), recurso muito utilizado nas décadas de 60 e 70.
- Utilizar uma **linguagem de programação** para aplicar o algoritmo (codificação) e gerar o programa (software, aplicativo).

Para a criação de um software mais complexo (um sistema de informação, por exemplo), podemos utilizar as técnicas:

- Levantamento de dados e de requisitos, diagramação (diagramas da UML, por exemplo), codificação, testes e implantação do sistema.

## ALGORITMO

É uma sequência de instruções ordenadas de forma **lógica** para a resolução de um determinado problema.

Para desenvolvermos um algoritmo, basta utilizar uma língua, como o português, conhecido também como português estruturado ou pseudo-linguagem.

- Utilizar frases com apenas um verbo.
- Utilizar frases simples e curtas.
- Utilizar palavras sem sentido dúbio.
- Ser objetivo.

## LINGUAGEM DE PROGRAMAÇÃO

**Algoritmos** escritos em códigos para instruir o computador. Após compilado, gera o programa de computador.

Logo, um **PROGRAMA DE COMPUTADOR** é um conjunto de instruções em uma **sequência lógica** para resolver um problema ou realizar uma determinada tarefa.

Será necessário utilizarmos vários recursos para construirmos o algoritmo:

### OPERADORES ARITMÉTICOS:

- + (soma)
- (subtração)
- \* (multiplicação),

/ (divisão).

#### OPERADORES RELACIONAIS:

> (maior que)

< (menor que)

>= (maior ou igual)

<= (menor ou igual)

= (igual)

!= (diferente)

#### OPERADORES LÓGICOS:

**E (AND):** Retorna verdadeiro se ambas ou mais expressões forem verdadeiras.

**OU (OR):** Retorna verdadeiro se uma das expressões forem verdadeiras.

**NÃO (NOT):** Inverte a expressão se verdadeira passa para falso e vice-versa.

#### VARIÁVEIS

Armazenam dados de forma temporária na memória principal.

**VARIÁVEIS DE ENTRADA:** Armazenam dados oriundos de usuários ou de discos (banco de dados).

**VARIÁVEIS DE SAÍDA:** Armazenam dados processados como resultados.

**CONSTANTES** à Variáveis que armazenam valores fixos.

## TIPOS PRIMITIVOS DE VARIÁVEIS

- Pré-definidos pela linguagem:

**INTEIRO:** Admitem números inteiros.

**REAL:** Admitem números com partes fracionárias.

**CARACTERES:** Admitem valores alfanuméricos.

**LÓGICO:** Admitem valores lógicos (true ou false).

## COMANDOS DE I/O (INPUT/OUTPUT)

Pré-definidos pela linguagem:

**LER:** Receber dados de um dispositivo de entrada (teclado, mouse, leitor óptico, etc).

**ESCREVER:** Envia dados de uma variável de saída para a tela ou impressora.

## SINAIS DE ATRIBUIÇÃO

Para atribuir (carregar) um valor em uma variável podemos utilizar o sinal “=” ou utilizar os sinais “:=” como são utilizados nas linguagens COBOL, BASIC, PASCAL, ADA.

Exemplos:

X := 5

X = 5

## SINAIS DE COMPARAÇÃO DE IGUALDADE

Para comparar a igualdade entre valores de variáveis ou entre valores e variáveis, utilizando o sinal de “=” como na linguagem PASCAL ou os sinais “==” como em C, C# e Python.

Exemplos:

X = 5

X = Y

X == 5

X == Y

## ESTRUTURAS DE DECISÃO

Um bloco que executa um comando ou vários comandos a partir de uma condição, se esta for verdadeira.

Exemplos:

**SE** *condição for verdadeira* **ENTÃO**

..... *comandos;*

**FIM SE**

Opcional com **SENÃO** (obriga a realizar determinado comando se a condição **NÃO** for verdadeira):

**SE** *condição for verdadeira* **ENTÃO**

..... *comandos;*

**SENÃO** *comandos;*

**FIM SE**

## ESTRUTURAS DE REPETIÇÃO

É utilizada quando necessitamos repetir um conjunto de instruções sob uma condição (estrutura condicionada) ou não (estrutura determinada).

Exemplo com **ESTRUTURA CONDICIONADA**:

**ENQUANTO** *condição for verdadeira* **FAÇA**

*comando1;*

*comandoN;*

**FIM;**

Exemplo com **ESTRUTURA NÃO CONDICIONADA OU DETERMINADA:**

**PARA** *valor inicial* **ATÉ** *valor final* **FAÇA** (passos)

*comando1;*

*comandoN;*

**FIM**

Em lógica podemos dizer que “2 é maior do que 1” ou que “1 é menor do que 2”.

## SAIBA MAIS...

Dê uma olhada nos links abaixo para saber mais sobre computadores e lógica de programação:

[https://iow.unirg.edu.br/public/profarqs/2804/0272700/1.A\\_Historia\\_dos\\_computadores\\_e\\_da\\_computacao\\_-\\_imprimir.pdf](https://iow.unirg.edu.br/public/profarqs/2804/0272700/1.A_Historia_dos_computadores_e_da_computacao_-_imprimir.pdf)  
([https://iow.unirg.edu.br/public/profarqs/2804/0272700/1.A\\_Historia\\_dos\\_computadores\\_e\\_da\\_computacao\\_-\\_imprimir.pdf](https://iow.unirg.edu.br/public/profarqs/2804/0272700/1.A_Historia_dos_computadores_e_da_computacao_-_imprimir.pdf))

<http://www.ams.eti.br/livros/Fabiano.pdf>  
(<http://www.ams.eti.br/livros/Fabiano.pdf>)

Neste tópico vimos a estrutura básica de um computador e como teremos que instruí-lo utilizando primeiramente a lógica de programação e depois uma linguagem de programação para gerarmos o programa que irá instruir o computador para resolver um problema do mundo real. E para resolvermos estes problemas, será necessário utilizarmos vários operadores e as estruturas de decisão e de repetição.

# Quiz

Exercício Final

Conceitos e princípios da lógica de programação

INICIAR ➤

## Referências

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. *Lógica de programação: a construção de algoritmos e estruturas de dados*. 3. ed. São Paulo: Prentice Hall, 2005. 218 p.

MENEZES, N. N. C. *Introdução à programação com Python: algoritmos e lógica de programação para iniciantes*. 2. ed. São Paulo: Novatec, 2014. 328 p.

MANZANO, J. N. G.; OLIVEIRA, J. F. *Algoritmo: lógica para o desenvolvimento de programação*. São Paulo: Érica, 2004. 336 p.

GUGIK, Gabriel. Disponível em: <<https://iow.unirg.edu.br/public/profarqs/2804/0272700/1>>.

A\_Historia\_dos\_computadores\_e\_da\_computacao>. Acesso em: Junho/2018.



Avalie este tópico



Biblioteca

(<https://www.uninove.br/conheca-a->

a-



Índice

Histórico e características da linguagem Python

© Todos os direitos reservados

Ajuda?  
PRÓXIMO  
(<https://ava.uninove.br/ava-uninove-br/conheca-a->

Python





uninove/biblioteca/sobre-  
a-  
biblioteca/apresentacao/)  
Portal Uninove  
(<http://www.uninove.br>)  
Mapa do Site