**Lógica – Algorítmos e Programação de Computadores**

**- Módulo III**

Valmir Santana

**Caderno de Anotações**

**-Agosto/2020**

**Introdução a Linguagem de Programação e Algorítmos - Construção**

**Linguagem de programação, dados, proposições e algoritmos (ações)**

## **Algorítmo**

sequência finita de ações para resolver um problema estabelecido. Essa sequência finita de ações será traduzida em um conjunto de instruções de uma linguagem de programação (o programa ou software) que o computador executará para a solução do problema.

O processo de desenvolvimento do algoritmo e a sua tradução para uma linguagem de programação é denominado de construção de programa de computador e é uma das principais tarefas dos profissionais da área de Tecnologia da Informação (TI).

O estudo de Lógica de Programação deve se preocupar com o desenvolvimento das seguintes habilidades:  
• Conhecimento dos conceitos básicos que estejam relacionadas com algoritmos.  
• Entendimento das técnicas básicas de construção de algoritmos.  
• Análise de algoritmos.  
• Entendimento dos problemas de acordo com a sua complexidade.

A finalidade de um computador é receber, manipular e armazenar dados, por meio de programa (ASCENCIO; CAMPOS, 2007).  
O programa de computador (ou, simplesmente, *software*) contém um conjunto de  
instruções que objetivam tratar os dados para atingir um objetivo preestabelecido (SOUZA et al., 2012). Esse programa é desenvolvido em uma linguagem de programação. O resultado desse desenvolvimento é denominado de código fonte, pois ele é a origem para a transformação do mesmo em uma linguagem entendível pelo computador.  
Para que o código-fonte possa ser entendido pelo computador, é necessário sua transformação, que depende da linguagem escolhida.

Os processos de transformação de um código fonte são a ***compilação-linkeditação*** e a ***interpretação*** (CAPRON; JOHNSON, 2004; NORTON, 1996).

As principais diferenças entre o processo de ***interpretação*** e ***compilação-linkedição*** são, de acordo com Capron e Johnson (2004) e Norton (1996):

* Para execução do programa interpretado é necessário fazer uma avaliação da sintaxe antes para a execução depois, enquanto no compilado-linkeditado, é só executar o programa.
* O programa executável, o originado do processo compilado-linkeditado, para ser executado, requer apenas que o computador tenha o mesmo sistema operacional no qual foi gerado, enquanto o interpretado requer o interpretador também.

## **Dados**

***Qualquer elemento identificado em sua forma bruta*** que, por si só, não conduz a uma compreensão de determinados fatos e situações. Ele representa eventos ocorridos na empresa ou circunstâncias físicas. Todo e qualquer programa, independentemente do seu objetivo, faz tratamento de dados (a entrada) para gerar informações (a saída), de acordo com as instruções contidas nele.



A partir dos dados é que é possível gerar ou criar informação, que é o dado configurado de forma adequada ao entendimento e à utilização do ser humano.

São dois os tipos básicos de dados: **Numérico** e **Não numérico**.

**Dado Numérico** - É a representação de números sem a necessidade de utilização de uma tabela de conversão, normalmente, representado por um ou dois bytes (agrupamento de 8 bits). Os dados numéricos dividem-se em dois grupos: os inteiros e os decimais (ASCENCIO; CAMPOS, 2007; FORBELONE; EBERSPÄCHER, 2005).  
Qualquer número do conjunto dos números inteiros (*ℤ*) é um número inteiro, ou seja, qualquer número negativo (< 0) ou neutro (= 0) ou positivo (> 0) que não possua a parte decimal.  
São exemplos de números inteiros: -300; -2; 0; 9; 126.

Um número decimal, também denominado de ponto flutuante, é aquele que possui, além da parte inteira, a parte decimal (utiliza-se a vírgula com delimitador da casa decimal), ou seja, pertence ao conjunto dos números reais (ℝ).  
São exemplos de números decimais: -300,0; -2,0; -1,85; -0,3; 1,125; 78,4; 2158,387.

**Dado Não Numérico -**É um dado que não é numérico puro e é composto de uma sequência de caracteres (1 ou mais), sendo cada um dos caracteres representados em um byte.  
São exemplos de dados não numéricos: "123"; "AB2"; "EDUARDO"; "44,33"; "@33#ABC".  
 Normalmente, quando se deseja representar um conteúdo numérico como não numérico, escrevesse este conteúdo entre aspas ("123" ou "44,33").  
 Cada um dos caracteres é convertido de acordo com um padrão internacional adotado para a codificação de caracteres. A tabela de conversão mais utilizada, mas que não é a única, é a ASCII (*American Standard* *Code for Information Interchange*).