**Texto base**

**1**

**Introdução - Algoritmos e Fluxogramas**

Gilberto Alves Pereira

***Resumo***

*Nesse texto discutimos os conceitos básicos envolvidos quando falamos inicialmente sobre lógica de programação: Algoritmos, o funcionamento de um software, Linguagem de Programação e por fim falamos um pouco sobre a ferramenta que usaremos em nosso curso - o Fluxograma.*

**1.1. Algoritmo**

O termo **algoritmo** pode ser visto desde o **século IX**. Foi nesta época que o cientista, astrônomo e matemático persa *Abū ‘Abd Allāh Muhammad ibn Mūsā al-Khwārizmī* usou pela primeira vez o termo para indicar **regras** de operações aritméticas utilizando algarismos indo-arábicos (DIERBACH 2012).

No século XII, Adelardo de Bath traduziu o termo para o latim **Algorithmi**. De lá para cá, o termo evoluiu bastante, incluindo todos os **procedimentos definidos para resolver problemas** ou **realizar tarefas**.

DIERBACH (2012) A formalização da noção de algoritmo ocorreu em 1936 com os trabalhos de **Alan Turing** e **Alonzo Church**, que desenvolveram independentemente os modelos de **Máquinas de Turing** e **Cálculo Lambda**.

Do ponto de vista computacional, um **algoritmo** pode ser visto como um **conjunto de regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema em um número finito de passos**.

DIERBACH (2012) *Donald* ***Knuth***, um dos pesquisadores mais respeitados em algoritmos, indica uma lista de cinco propriedades que são **requisitos para algoritmos**:

* **Finitude**: um algoritmo deve sempre terminar após um número finito de etapas (ou passos).
* **Definição**: cada passo de um algoritmo deve ser definido com precisão. As ações a serem executadas deverão ser especificadas rigorosamente e sem ambiguidades.
* **Entrada**: valores que são dados ao algoritmo antes que ele inicie.
* **Saída**: os valores resultantes das ações do algoritmo a partir de uma determinada entrada.
* **Eficácia**: todas as operações a serem realizadas pelo algoritmo devem ser suficientemente básicas para poderem, em princípio, ser feitas com precisão e em um período de tempo finito por um homem usando papel e lápis.



**Figura 1.1. Esquema do funcionamento de um algoritmo. Fonte Autor.**

Embora os requisitos de Knuth sejam intuitivos, falta-lhes **rigor formal**. A formalidade pode ser conseguida com o uso de **lógica**. Assim, vamos exigir que um algoritmo seja **uma sequência lógica de passos com começo, meio e fim.**

Comumente, esta lógica é conhecida como **lógica de programação** e isto ocupará grande parte de nossa disciplina.

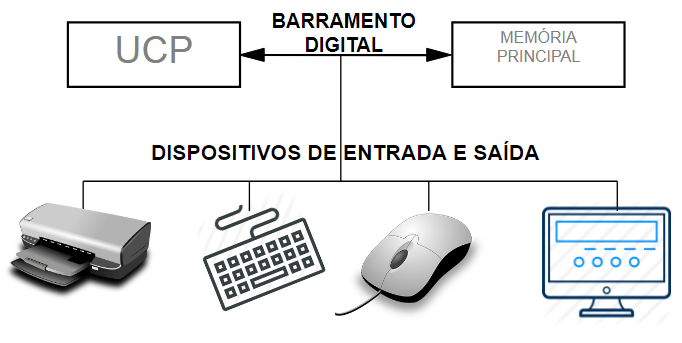
**1.2. Funcionamento de um Software**

De uma maneira simplificada, um software para funcionar precisa de um ambiente chamado Sistema Digital. O sistema digital é composto minimamente por uma unidade de processamento de instruções (UCP), memória (volátil - RAM), e de dispositivos de entrada e saída. As informações dentro do sistema digital estão em formato binário (zeros e uns).

O software é um conjunto de instruções para o processador que fica armazenado na memória. Esse conjunto de instruções basicamente fazem o seguinte:

1. Obtém dados de algum dispositivo de entrada (teclado, mouse, HD, placa de rede,etc) e armazena na memória;
2. Com as informações na memória efetua cálculos - o que chamamos de processamento, armazenando esses resultados na memória;
3. Por fim, envia esses resultados desses cálculos da memória para algum dispositivo de saída (impressora, tela, placa de rede,etc).

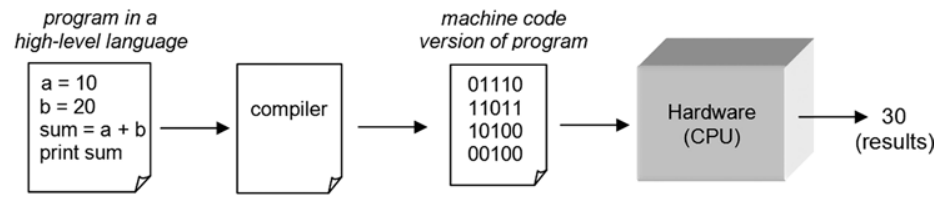
Portanto, um software basicamente lida com esses componentes (dispositivos de entrada e saída, memória e cálculos) frequentemente. Basicamente, quando construímos um software usamos comandos que lidam com esses componentes (entrada, saída, memória e processamento).



**Figura 1.2. Sistema Digital. Fonte Autor.**

**1.3. Sobre Linguagens de Programação e Algoritmos**

A Linguagem de Programação auxilia no processo de conversão para a linguagem binária do processador, através de uma série de regras que eliminam as ambiguidades.



**Figura 1.3. Conversão de algoritmo em código para linguagem binária. Fonte DIERBACH 2012.**

Algoritmos resolvem qualquer problema. As Linguagens resolvem problemas utilizando os recursos do Sistema Digital (dispositivos de entrada, memória, processamento e dispositivos de saída).

Nem todos Algoritmos podem ser implementados por Linguagens.

**1.4. O processo para resolver problemas computacionais**

A resolução de um problema computacional não envolve simplesmente o ato de programar um computador. Ele é um **processo**, sendo que a programação é apenas um dos passos. Antes de escrever um programa é preciso entender e pensar a solução do problema, depois desenvolver o projeto, escrever e testar a solução encontrada.

Uma vez que o algoritmo tenha sido definido, podemos **simulá-lo** com ferramentas como Scratch, VisuAlg ou Raptor, representá-los em **Fluxograma** ou **implementá-lo** diretamente em alguma **linguagem de programação** (Python, C, C++, Java, Pascal, PHP, dentre outras). Nossa disciplina será focada em **Fluxograma**.

**1.5. Sobre Fluxograma**

Fluxograma é uma forma padronizada e eficaz para se representar os passos lógicos de um determinado processo (algoritmo), é uma técnica de representação gráfica na qual se utilizam símbolos previamente convencionados, permitindo a descrição clara e precisa do fluxo, ou sequência de um processo, bem como sua análise e redesenho.

* **Vantagens**: maior clareza na visualização do fluxo de execução (“imagens valem mais do que mil palavras”).
* **Desvantagens**: requer conhecimento de convenções gráficas e dificuldade para fazer correções.

O fluxograma é utilizado para diversas finalidades, onde se tem a necessidade de controlar processos, visualizar as operações que são realizadas, as entradas e saídas de uma operação, entre outras. Os fluxogramas possuem símbolos que permitem descrever fluxos de processos (não são necessariamente usados para programas de computador).

O fluxograma é representado por símbolos e cada um é utilizado para uma finalidade específica, o que torna o fluxograma mais fácil de compreender, pois a representação segue um padrão pré-estabelecido.

Neste curso, usaremos apenas uma pequena parcela dos símbolos disponíveis para uso nos fluxogramas, mostrados na tabela a seguir:

**Tabela 1.1. Símbolos do fluxograma e seus significados.**

| SÍMBOLO | SIGNIFICADO | DESCRIÇÃO |
| --- | --- | --- |
|  | Terminação | Utilizado para indicar o início e fim ou de um fluxograma. |
|  | Processo | Utilizado para indicar uma ação, isto é, um processo a ser executado. |
|  | Decisão | Utilizado para comparar dados e desviar o fluxo dependendo da avaliação do teste lógico para verdadeiro ou falso. |
|  | Entrada manual | Utilizado para entrada manual de dados através do teclado. |
|  | Processo pré-definido ou módulo | Utilizado para indicar uma chamada a uma sub-rotina ou função. |
|  | Exibir na tela | Utilizado para exibir dados na tela. |
|  | Conector | Utilizado para continuar o fluxograma em outra parte na mesma página. |
|  | Conector de página | Utilizado para continuar o fluxograma em outra página. |
|  | Seta de fluxo de dados | Utilizado para indicar o sentido do fluxo de dados conectando os símbolos existentes. |

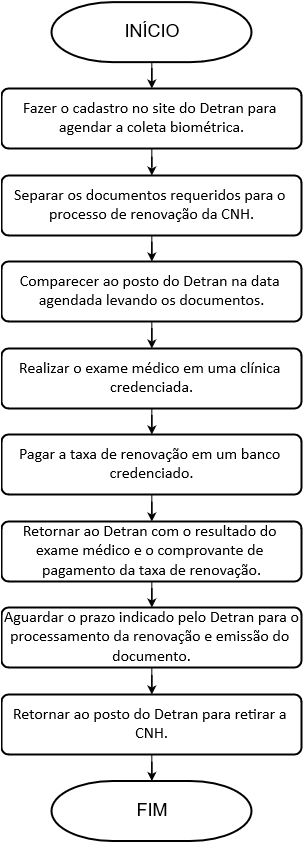
**Exemplo 1:**

Imagine que sua carteira de motorista tenha vencido. Para poder continuar dirigindo, você precisa renová-la, então o primeiro passo é definir o problema: **Renovar a CNH.**

Para resolver este problema, devemos seguir os seguintes passos:

1. Fazer o cadastro no site do Detran para agendar a coleta biométrica;
2. Separar os documentos requeridos para o processo de renovação da CNH;
3. Comparecer ao posto do Detran selecionado na data agendada e levar os documentos;
4. Realizar o exame médico em uma clínica credenciada;
5. Pagar a taxa de renovação em um banco credenciado;
6. Retornar ao posto do Detran com o resultado do exame médico e o comprovante de pagamento da taxa de renovação;
7. Aguardar o prazo indicado pelo Detran para o processamento da renovação e emissão do documento;
8. Retornar ao posto do Detran para retirar a CNH.

Esses oito passos podem ser representados no fluxograma abaixo, e vale lembrar que todo fluxograma deve ter **início** e **fim**.



**Figura 1.4. Fluxograma referente ao exemplo 1. Fonte Autor.**

Para esse exemplo, além dos símbolos de início e fim, utilizamos somente o de processo, para indicar as ações a serem realizadas e em qual ordem, de acordo com o fluxo indicado pelas setas.

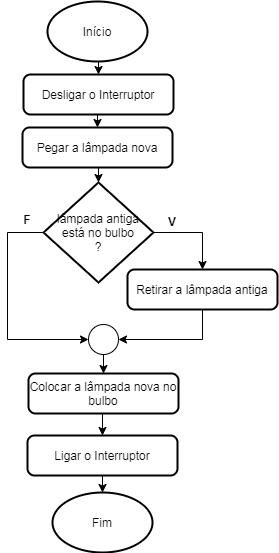
Mas se você tiver algum dos documentos requeridos ou não houver datas disponíveis para agendamento, o que fazer nestas situações? Veremos como resolver problemas desse tipo nas próximas aulas, com estruturas de seleção e repetição.

**Exemplo 2:**

Fluxograma para troca de uma lâmpada. Nesse exemplo, estamos usando um símbolo diferente e novo em relação ao exemplo anterior, o símbolo de decisão.

**Figura 1.5. Símbolo de decisão. Fonte Autor.**

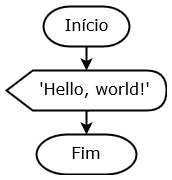
Este símbolo nos permite criar dois caminhos em nosso fluxograma. Nesse caso, um dos caminhos não fazemos nada e o outro retiramos a lâmpada do bulbo. Tomamos o caminho dependendo da resposta a pergunta - Verdadeiro ou Falso. Isso será visto com maiores detalhes nas próximas aulas.



**Figura 1.6. Fluxograma referente ao exemplo 2. Fonte Autor.**

**Exemplo 3:**

Para exibir uma mensagem na tela, utilizamos o símbolo **Exibir**. Esse símbolo é uma Saída de Dados, que pode ser utilizado para mostrar o resultado de um processo, um nome ou até mesmo para pedir para que o usuário digite alguma coisa.



**Figura 1.7. Fluxograma referente ao exemplo 3. Fonte Autor.**

* Nesse fluxograma, temos os símbolos de início e fim, que são colocados por padrão.
* O texto ‘Hello, world!’ será mostrado na tela para o usuário. É uma forma de saída de dados.
* Você pode criar o seu fluxograma em [www.draw.io](http://www.draw.io), uma aplicação online *open source*.

**1.6. Referências**

DIERBACH, C. “Introduction to Computer Science Using Python: A Computational Problem-Solving Focus”1st Edition, New York: Wiley, 2012.