

Algoritmos para Resolução de Problemas

Site: [AVA - CTEAD/IFPA](#)

Curso: NC/FIC/PW - 01.Ambiente Virtual de Aprendizagem - AVA

Livro: Algoritmos para Resolução de Problemas

Impresso por: VERENA CRISTINA BATISTA DE SOUSA (NC2020-PW-A)

Data: quinta, 28 Mai 2020, 12:10

Sumário

1. ALGORITMOS PARA RESOLUÇÃO DE PROBLEMAS

- 1.1. Definição do problema
- 1.2. Análise do problema
- 1.3. Construção do modelo
- 1.4. Programação
- 1.5. Execução
- 1.6. Documentação
- 1.7. Referências Bibliográficas

1. ALGORITMOS PARA RESOLUÇÃO DE PROBLEMAS

Em qualquer atividade de programação de computadores, desejamos resolver algum problema e utilizar como ferramenta o computador, com o objetivo de acelerar a resolução do problema e garantir que todas as atividades previamente programadas sejam executadas. Nesta atividade, descrevemos os objetos manipulados pelo programa, e em seguida, descrevemos, numa linguagem apropriada, um algoritmo que efetue a manipulação desses objetos.[2]

Para resolução de problemas, visando a implementação dessa solução em um computador, convém seguirmos os seguintes passos:

1.1. Definição do problema

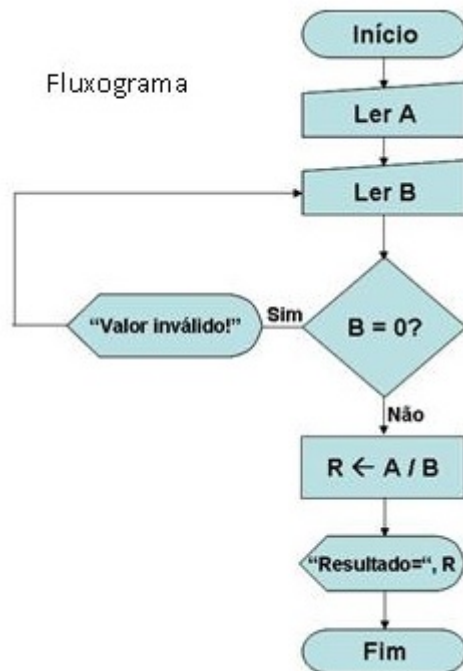
Neste item, devemos verificar se o enunciado do problema se apresenta de forma clara e completa. Precisamos observar cuidadosamente se não temos dúvida ou interpretação ambígua do que está sendo solicitado.

1.2. Análise do problema

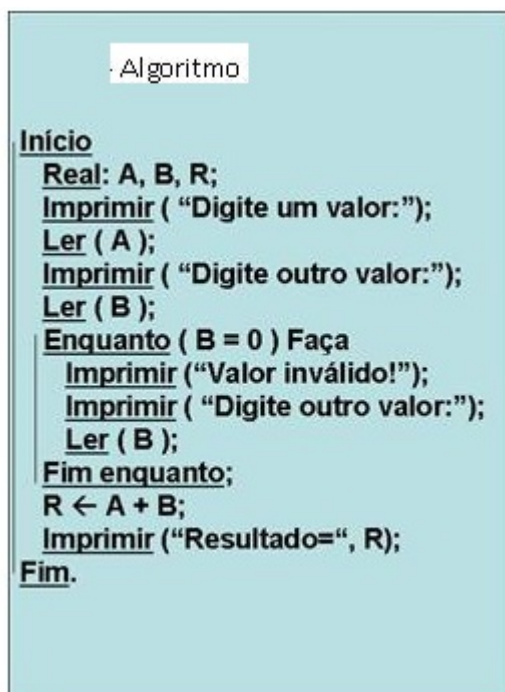
A análise é dividida em três fases. Na primeira, determinamos os dados de entrada, ou seja, quais os dados são necessários para resolver o problema, sem os quais não teríamos como resolvê-lo. Em seguida, faço a definição de como vou resolver o problema, criando fórmulas, definindo regras, estabelecendo condições, entre outros elementos necessários para se chegar ao resultado esperado. Finalmente, na terceira fase, especifico qual será a saída, ou seja, o resultado que será mostrado ao usuário.

1.3. Construção do modelo

Durante muitos anos, no início do desenvolvimento de programas para computadores, foi utilizado fluxograma para construção do modelo, uma representação gráfica, por meio de símbolos geométricos, para representar o fluxo de uma solução lógica de determinado problema, como pode ser verificado no exemplo mostrado na figura abaixo:



No ambiente atual de desenvolvimento de programas para uso em computadores, utilizamos como ferramenta para descrição do modelo um algoritmo, que é a descrição de uma sequência de passos que levam a execução de uma tarefa em tempo determinado, conforme exemplo mostrado na figura abaixo:



1.4. Programação

A programação de computadores envolve a transformação do modelo criado no passo anterior em uma linguagem conhecida pela máquina, utilizando para isso o que chamamos de linguagem de programação. A **linguagem de programação** é um conjunto de termos (vocabúlos) e de regras (sintaxe) que permitem a formulação de instruções a um computador.

O processo de programação gera um arquivo de dados chamado **programa-fonte** ou **código-fonte**. O código-fonte é o conjunto de palavras ou símbolos escritos de forma ordenada, contendo instruções em uma das linguagens de programação existentes, de maneira lógica.

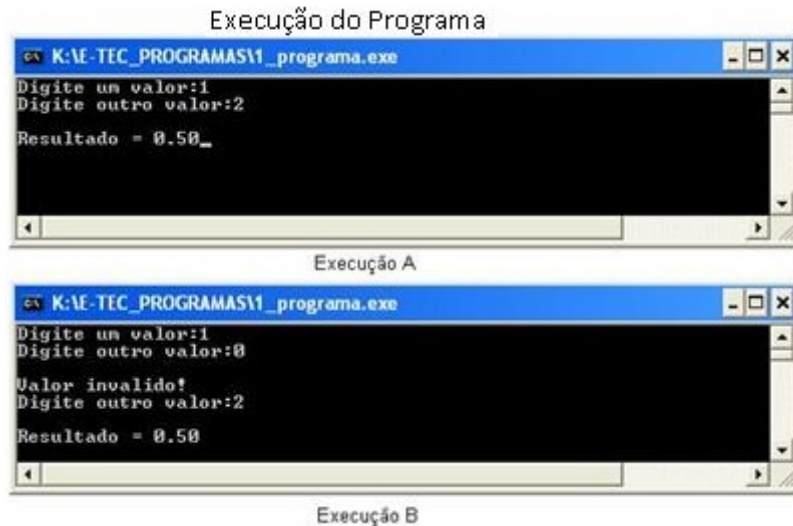
A figura abaixo mostra a programação do algoritmo mostrado durante a Construção do Modelo, utilizando a linguagem de programação C.

```
Programa escrito em C
1 #include <stdio.h>
2 #include <conio.h>
3 main ()
4 {
5     float A,B,R;
6     printf ("Digite um valor:");
7     scanf ("%f", &A);
8     printf ("Digite outro valor:");
9     scanf ("%f", &B);
10    while (B==0)
11    {
12        printf("\nValor invalido!");
13        printf("\nDigite outro valor:");
14        scanf ("%f", &B);
15    }
16    R=A/B;
17    printf("\nResultado = %.2f",R);
18    getch();
19 }
```

1.5. Execução

Executar um programa significa submeter o programa ao computador, fazendo com que o programa e os dados necessários para execução do programa sejam colocados na memória principal do computador e essas informações serão manipuladas pelo processador, executando todas as operações definidas pelo programador.

A figura abaixo apresenta a execução do programa mostrado na Programação, considerando inicialmente a execução com entrada de valores válidos (A) e posteriormente a execução do programa com a primeira entrada inválida, seguida de outra entrada válida (B).



1.6. Documentação

A documentação representa a inserção no código-fonte de informações necessárias ao entendimento de cada parte do programa escrito, de forma a facilitar a leitura do programa por qualquer pessoa que tenha acesso ao código-fonte.

Guimarães e Lages (1985) informam que dificilmente se encontrará um programador que ponha em dúvida a importância da boa documentação na estrutura dos programas. Todos possuem a certeza que a boa documentação facilita a vida de todos: de quem tem a responsabilidade de programar, dos que realizam a tarefa de modificar os programas, de quem usa o programa e quem administra. Para a boa programação convém tornarmos o código do programa e sua documentação algo indivisível, como partes integrantes de um todo, forçosamente elaborados ao mesmo tempo, dependentes um do outro.[4]

A figura abaixo apresenta o programa descrito na fase da Programação com modificações, incluindo agora comentários sobre o programa. Dois tipos de comentários são utilizados: comentário de bloco e comentário de linha.

```
divisao.cpp
1  /*Programa 1
2  Autor: Nome do Autor
3  Data: 15/05/2020
4  Objetivo do programa: realizar a divisão entre
5  dois números
6  */
7  //Definição das includes
8  #include <stdio.h>
9  #include <conio.h>
10 main() //Função principal
11 {
12     float A,B,R; //Declaração de variáveis
13     //Entrada do primeiro valor
14     printf("Digite um valor:");
15     scanf("%f",&A);
16     //Entrada do segundo valor
17     printf("Digite outro valor:");
18     scanf("%f",&B);
19     //Estrutura de repetição para forçar o usuário
20     //a digitar um valor válido para a variável B
21     while(B==0)
22     {
23         printf("\nValor inválido!");
24         printf("Digite outro valor:");
25         scanf("%f",&B);
26     }
27     //Cálculo da divisão
28     R=A/B;
29     //Exibição do resultado
30     printf("\nResultado = %.2f",R);
31     getch();
32 }
```

O **comentário de bloco** é representado pelos caracteres `/*` no início do comentário e `*/` no final do comentário, fechando o bloco. Todas as informações escritas dentro do bloco não serão interpretadas pelo compilador no momento da transformação do código-fonte em código-objeto, representando simples comentário.

(Glossário) **COMPILADOR** - Programa que, a partir de um código escrito em uma linguagem de alto nível, o **código fonte**, cria um programa semanticamente equivalente porém escrito em código de máquina, **código objeto**.

O **comentário de linha** é representado pelos caracteres `//` à esquerda do comentário. Todas as informações escritas **na linha** e à direita do `//` serão consideradas como comentário.

1.7. Referências Bibliográficas

- [1] FARRER, Harry. et. al. **Programação estruturada de Computadores. Algoritmos estruturados**. 2 Ed. Rio de Janeiro: LTC Livros Técnicos e Científicos Editora Ltda, 1989.
- [2] MARTINS, J. Pavão. **Introdução à programação usando Pascal**. Lisboa: Editora McGraw-Hill de Portugal Lda, 1994.
- [3] ARAÚJO, Everton Coimbra de. **Algoritmos: Fundamentos e Prática**. 2 Ed. ampl. e atual. Florianópolis: VisualBooks Editora, 2005.
- [4] GUIMARÃES, Ângelo de Moura. LAGES, Newton Alberto de Castilho. **Algoritmos e estruturas de dados**. Rio de Janeiro: Campus, 1985.
- [5] SCHILDT, Herbert. **C Completo e total**. São Paulo: MAKRON BOOKS, 1997.
- [6] LOUDON, Kyle. **Dominando algoritmos com C**. São Paulo: CIENCIA MODERNA COMPUTAÇÃO, 2000.
- [7] LAUREANO, Marcos. **Programando em C para Linux, Unix e Windows**. Rio de Janeiro: Brasport Livros, 2005.
- [8] MEDINA, Marco. FERTIG, Cristina. **Algoritmos e programação: teoria e prática**. São Paulo: NOVATEC INFORMATICA, 2005.
- [9] FERREIRA, Aurélio Buarque de Holanda. **Novo Dicionário Aurélio Século XXI**. Rio de Janeiro: Nova Fronteira, 1999.
- [10] INFORMAÇÃO. In: Wikipédia: a enciclopédia livre. Disponível em: <<http://pt.wikipedia.org/wiki/Informa%C3%A7%C3%A3o>>. Acesso em: 15/07/2009.
- [11] PEREZ, Anderson Luiz Fernandes. **Linguagens de programação: sintaxe e semântica de linguagens de programação e conceitos de linguagens compiladas e interpretadas**. Disponível em <<http://www.univasf.edu.br/~anderson.perez/ensino/intprog/>> acesso em 16/07/2009.
- [12] BARBOSA, Lisbete Madsen. **Ensino de algoritmos em cursos de computação**. São Paulo: EDUC, 2001.
- [13] BIT. In: Wikipédia: a enciclopédia livre. Disponível em: <<http://pt.wikipedia.org/wiki/Bit>>. Acesso em: 15/07/2009.
- [14] CÓDIGO-FONTE. In: Wikipédia: a enciclopédia livre. Disponível em: <<http://pt.wikipedia.org/wiki/C%C3%B3digo-fonte>>.
- Acesso em: 15/07/2009.
- [15] PI (II). In: Wikipédia: a enciclopédia livre. Disponível em: <<http://pt.wikipedia.org/wiki/Pi>>. Acesso em: 17/07/2009.
- [16] BUGS. In: Wikipédia: a enciclopédia livre. Disponível em: <<http://pt.wikipedia.org/wiki/Bug>> . Acesso em: 15/08/2009.
- [17] FREEDMAN, Alan. **Dicionário de Informática**. São Paulo: Makron Books, 1995.
- [18] RAIZ QUADRADA. Wikipédia a Enciclopédia Livre. Disponível em <http://pt.wikipedia.org/wiki/Raiz_quadrada> acesso em 30/07/2009.