

CURSO DE INFORMÁTICA

Algoritmos

Ricardo José Cabeça de Souza

Parte 2

Sumário

UNIDADE I - ALGORITMOS

- 1.4 Resolução de Problemas
 - 1.4.1 - Definição do Problema
 - 1.4.2. Análise do Problema
 - 1.4.3 Construção do Modelo
 - 1.4.4 Programação
 - 1.4.5 Execução
 - 1.4.6 Documentação
- 1.5 Resolvendo Problemas
 - 1.5.1 Entrada
 - 1.5.2 Como Resolver o Problem
 - 1.5.3 Saída
 - 1.5.4 Seqüência das Operações Realizadas

REFERÊNCIAS BÁSICAS

REFERÊNCIAS COMPLEMENTARES

1.4 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:

1.4.1 - 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.

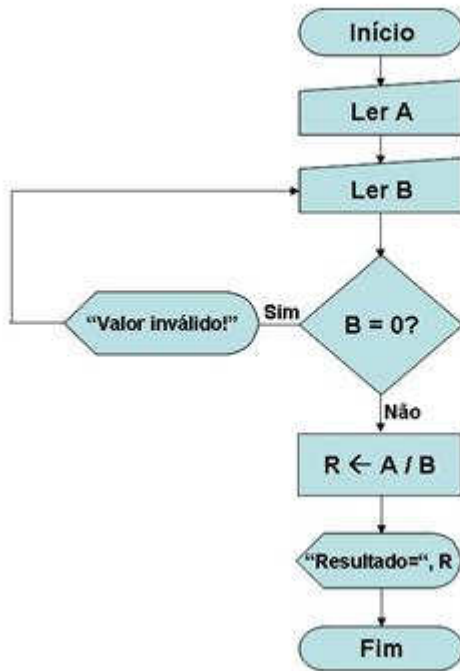
1.4.2. Análise do problema

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.

1.4.3 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 1.4.

Figura 1.4 - Fluxograma



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 seqüência de passos que levam a execução de uma tarefa em tempo determinado, conforme exemplo mostrado na figura 1.5.

Figura 1.5 - Algoritmo

```
Início  
Real: A, B, R;  
Imprimir ( "Digite um valor:");  
Ler ( A );  
Imprimir ( "Digite outro valor:");  
Ler ( B );  
Enquanto ( B = 0 ) Faça  
  Imprimir ("Valor inválido!");  
  Imprimir ( "Digite outro valor:");  
  Ler ( B );  
Fim enquanto;  
R ← A + B;  
Imprimir ("Resultado=", R);  
Fim.
```

1.4.4 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 (vocábulos) e de regras (sintaxe) que permitem a formulação de instruções a um computador. [3]

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.[14]

A figura 1.6 mostra a programação do algoritmo mostrado na figura 1.5, utilizando a linguagem de programação C.

Figura 1.6 – 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 }
```

1.4.5 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 1.7 apresenta a execução do programa mostrado na figura 1.6, 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).

Figura 1.7 – Execução do Programa



1.4.6 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 1.8 apresenta o programa descrito na figura 1.3 com modificações, incluindo agora comentários sobre o programa. Dois tipos de comentários são utilizados: comentário de bloco e comentário de linha.

Figura 1.8 – Programa-Fonte Comentado

```
1 /*Programa 1
2 Autor: Ricardo Souza
3 Data: 15/08/2009
4 Objetivo: Realizar a divisão de dois números
5 */
6 //Definição das Includes
7 #include <stdio.h>
8 #include <conio.h>
9
10 main () //Função Principal
11 {
12 float A,B,R; //Declaração de Variáveis
13 //Entrada do 1º Valor
14 printf ("Digite um valor:");
15 scanf ("%f", &A);
16 //Entrada do 2º Valor
17 printf ("Digite outro valor:");
18 scanf ("%f", &B);
19 //Estrutura de repetição enquanto B = 0
20 while (B==0)
21 {
22 printf ("\nValor invalido!");
23 printf ("\nDigite outro valor:");
24 scanf ("%f", &B);
25 }
26 //Cálculo da divisão
27 R=A/B;
28 //Exibição do resultado
29 printf ("\nResultado = %.2f",R);
30 getch();
31 }
```

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.

1.5 RESOLVENDO PROBLEMAS

Com base nos itens mostrados anteriormente, vamos exemplificar o raciocínio lógico para se resolver determinado problema.

O problema: desejo um algoritmo para realizar a soma de dois números.

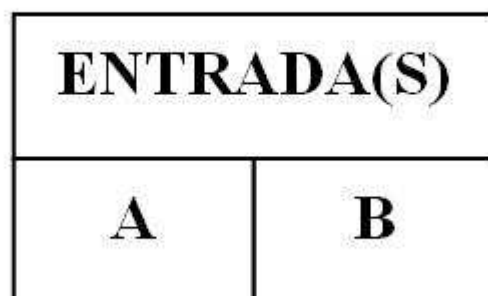
Seguindo a orientação descrita no item 1.4.1, inicialmente preciso verificar a definição do problema. O que desejo é realizar a soma entre dois números, sejam eles quais forem. Logo, tenho certeza, sem qualquer sombra de dúvida ou interpretação equivocada, a ação que preciso realizar.

Em seguida, realizamos a análise do problema. Essa análise envolve três pontos essenciais: quais os dados são necessários para resolução do problema (**entrada**), como efetivamente vou realizar a resolução do problema (**operações lógicas e/ou aritméticas**) e como o resultado será apresentado ao usuário (**saída**).

1.5.1 Entrada

Inicialmente, verifico quais os dados são necessários para realização dessa tarefa, definida neste ponto com a(as) entrada (entradas). Considerando que desejo realizar a soma entre dois números, logicamente preciso dos dois números. Continuando a análise, considerando ainda que posso ter como entrada qualquer número, **não posso** definir dois números específicos, exatos, fixos, como por exemplo, 8 e 10. Então, represento esses números através de um símbolo, representados aqui pelas letras **A** e **B**, conforme mostrado na figura 1.9.

Figura 1.9 – Representação da(s) Entrada(s)



1.5.2 Como resolver o problema

Continuando a análise, passo então a pensar em como vou resolver o problema.

Considerando que desejo realizar uma soma entre dois números, preciso de uma fórmula matemática para realizar a ação, como a apresentada na figura 1.10 a seguir:

Figura 1.10 – Representação da Operação

$$\mathbf{A + B}$$

Contudo, preciso especificar como vou apresentar o resultado. A simples execução da operação mostrada no quadro acima não é suficiente. Em geral, tenho normalmente duas opções: posso exibir o resultado para o usuário, utilizando neste caso a unidade de saída do computador (vídeo) ou armazenar o resultado em algum lugar da memória principal (criar um espaço na memória para armazenar o resultado). Vou escolher a opção de guardar o resultado na memória do computador e para isso vou utilizar o símbolo **R** para representar esse lugar, complementando a fórmula inicial, ficando o resultado da operação conforme mostrado na figura 1.11:

Figura 1.11 – Resultado da Operação

COMO?
R ← A + B

1.5.3 Saída

Definidas as entradas especificadas no item 1.5.1, pensado em como resolver o problema da soma dos dois números definido no item 1.5.2, só nos resta apresentar o resultado da operação armazenado no símbolo **R**, criado especificamente para essa tarefa.

Figura 1.12 – Representação da Saída



1.5.4 Seqüência das operações realizadas

Baseado nos itens expostos acima, podemos agora fazer um esboço da seqüência das operações realizadas para execução da tarefa, como a seguir:

Seqüência de operações:

- a) Entrar com o primeiro valor (A);
- b) Entrar com o segundo valor (B);
- c) Realizar a operação matemática:

$$\mathbf{R \leftarrow A + B}$$

- d) Apresentar o resultado da operação (R).

A seqüência descrita acima chama-se **algoritmo**. Observamos que o algoritmo segue uma seqüência lógica na realização de cada tarefa, finita e com a apresentação do resultado esperado, sejam quais forem os valores definidos para A e B.

É possível também, em alguns casos, alterar a ordem de execução das tarefas. Por exemplo, podemos alterar a execução das tarefas dos itens

a) e b) sem haver alteração do resultado esperado. Contudo, não é possível alterar a ordem de outros itens sem o comprometimento do resultado esperado. Como executar a soma sem a entrada dos valores? Como mostrar o resultado sem a execução da soma?

Para realizar a verificação da seqüência de operações descritas, confirmando se realmente executa a tarefa solicitada, no caso, a soma de dois números, podemos **simular** a execução das tarefas, executando cada passo do algoritmo definindo valores reais. Esse processo chama-se **teste de mesa**.

TESTE DE MESA		
ALGORITMO	SIMULAÇÃO	COMPUTADOR VIRTUAL
a) Entrar com o primeiro valor (A);	Na memória principal do computador é criada uma variável com o nome A, representada pela figura ao lado e o valor informado pelo usuário (por exemplo, o número 5) é armazenado neste espaço;	<p>A</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">5</div>
b) Entrar com o segundo valor (B);	Na memória principal do computador é criada uma variável com o nome B, representada pela figura ao lado e o valor informado pelo usuário (por exemplo, o número 10) é armazenado neste espaço;	<p>A B</p> <div style="display: flex; gap: 10px;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">5</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">10</div> </div>

ALGORITMO	SIMULAÇÃO	COMPUTADOR VIRTUAL
c) Realizar a operação matemática: $R \leftarrow A + B$	É criada uma variável R e o resultado da operação de soma entre o valor de A(5) com o valor de B(10) é armazenado na variável R;	

ALGORITMO	SIMULAÇÃO	COMPUTADOR VIRTUAL
d) Apresentar o resultado da operação (R).	No monitor do usuário é exibido o valor armazenado na variável R. No exemplo será exibido o valor igual a 15.	

Veja que utilizamos a linguagem natural para descrever as ações a serem executadas, passo a passo. Contudo, isso fica impraticável se considerarmos que cada um pode escrever da forma que desejar. Para facilitar a escrita de algoritmos, considerando que o texto da seqüência de ações a serem realizadas pode ser escrita de qualquer forma, utilizaremos uma linguagem específica, chamada de **pseudocódigo**, estabelecendo padrão de escrita para os algoritmos desenvolvidos daqui pra frente.

(Glossário) **PSEUDO**- Falso; imaginário.

REFERÊNCIAS BÁSICAS

- [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 (Π). 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.

REFERÊNCIAS COMPLEMENTARES

SCHILDT, Herbert. **C Completo e total**. São Paulo: MAKRON BOOKS, 1997.

DAMAS, Luís. **Linguagem C**. 10. ed. Rio de Janeiro: LTC, 2007.

LOUDON, Kyle. **Dominando algoritmos com C**. São Paulo: CIENCIA MODERNA COMPUTAÇÃO, 2000.

JAMSA, Kris. **Programando em C/C++: a bíblia**. São Paulo: Makron Books, 2000.

LOPES, Anita. **Introdução a programação: 500 algoritmos resolvidos**. Rio de Janeiro: Campus, 2002.

LAUREANO, Marcos. **Programando em C para Linux, Unix e Windows**. Rio de Janeiro: BRASPORT LIVROS, 2005.

DEITEL, Paul J.; DEITEL, Harvey M. **C++ como programar**. Porto Alegre: Bookman, 2001.

PINTO, Wilson Silva. **Introdução ao desenvolvimento de algoritmos e estrutura de dados**. São Paulo: Érica, 1990. KERNIGHAN, Brian W.; RITCHIE, Dennis M. **C: a linguagem de programação**. Rio de Janeiro: Campus, 2000. (005.133 K39c).

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo. **Estudo dirigido de algoritmos**. São Paulo: Érica, 1997. SALVETTI, Dirceu Douglas; BARBOSA, Lisbete Madsen. **Algoritmos**. São Paulo: Makron Books, 1998. (005.1. S183A).

MIZRAHI, Victorine Viviane. **Treinamento em linguagem C: curso completo**. Módulo I. São Paulo: McGraw-Hill, 1990.