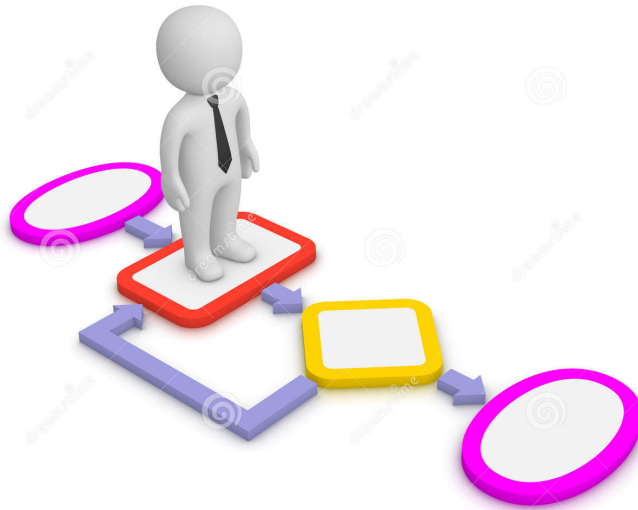


Programação de Soluções Computacionais

Fabício Curvello Gomes



Algoritmos

Conceitos Básicos

O que é Algoritmo?

Um **algoritmo** é uma sequência ordenada e finita de etapas, cuja execução passo a passo resolve um determinado problema.

Observe que a definição de **problema** é bastante genérica, o que nos permite imaginar algoritmos para as mais diversas situações, não necessariamente de caráter algébrico.

Na realidade, nossa própria memória já é recheada de algoritmos para a execução de tarefas diariamente.



O que é Algoritmo? (Cont.)

Eis um exemplo de algoritmo para uma situação rotineira:



- Ex1: Algoritmo para sacar dinheiro em um caixa eletrônico:

1. *Passar o cartão do banco no leitor;*
2. *Digitar senha da conta corrente (ou colocar o dedo no sensor);*
3. *No menu que aparece na tela, escolher a opção “Saque”;*
4. *Digitar o valor desejado e pressionar a tecla “Entrar”;*
5. *Confirmar o saque e pressionar a tecla “Entrar”;*
6. *Ficar esperando em frente ao caixa até sair o dinheiro;*
7. *Retirar o dinheiro.*

Repare que alguns (ou todos os) itens parecem óbvios, mas o algoritmo precisa ser claro assim, de forma que não gere dúvidas, mesmo para quem nunca executou tais ações antes.



Atividade:

Vamos tentar descrever outras situações rotineiras em forma de algoritmos. Descreva em seu caderno as seguintes situações:

1. Algoritmo para tomar banho:
2. Algoritmo para vir da sua casa ao curso:
3. Algoritmo para preparar o café numa cafeteira elétrica:
4. Algoritmo para procurar a explicação de uma matéria em um livro escolar:
5. Algoritmo para trocar uma lâmpada que está queimada no lustre:



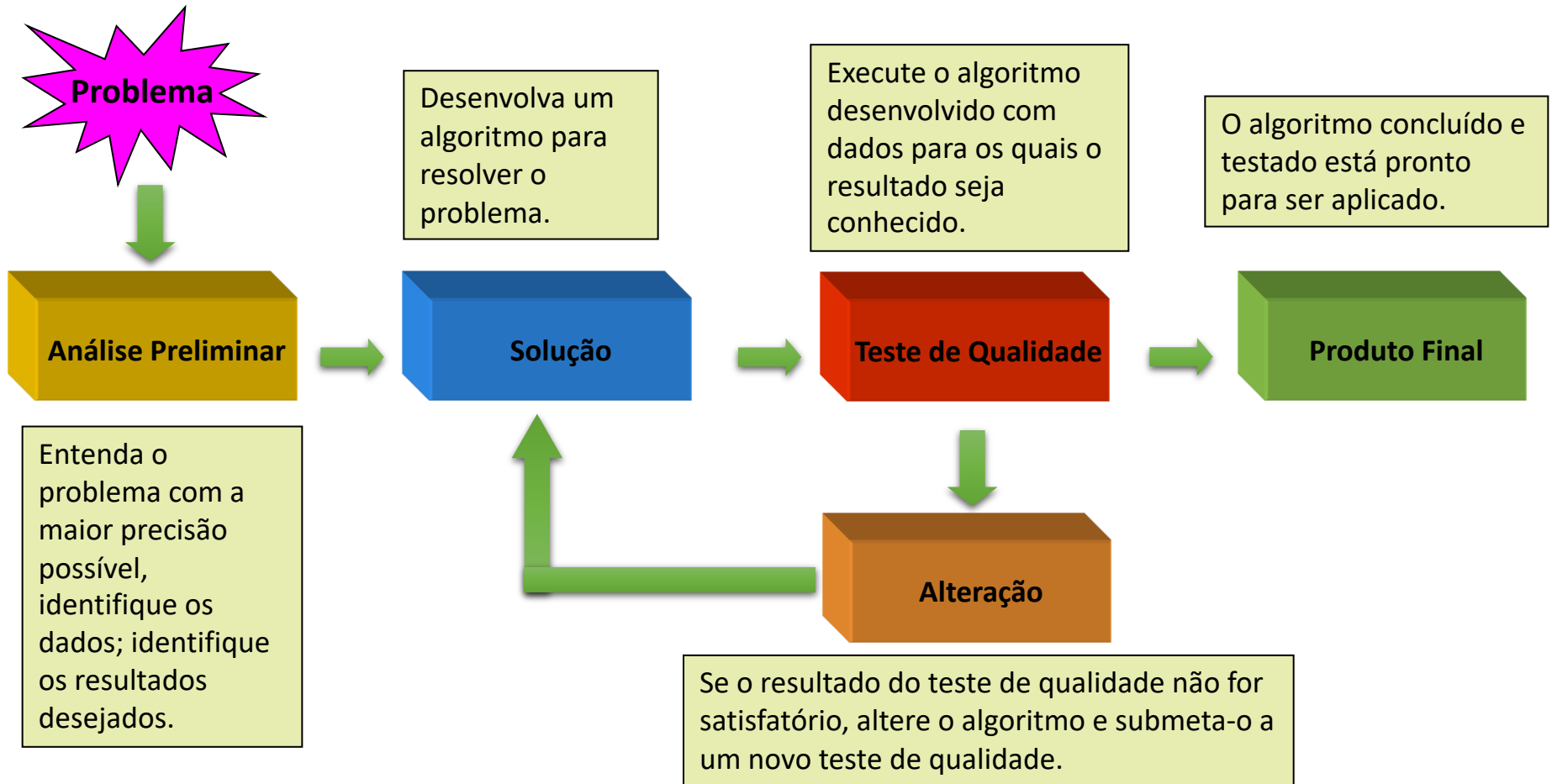
Algumas conclusões sobre a Atividade:

De acordo com as diversas respostas produzidas por todos na turma, pôde-se observar que:

1. Podem existir diversos **algoritmos diferentes**, mas **corretos** para a resolução de um determinado problema.
2. Alguns esquecem detalhes que parecem óbvios, mas este tipo de esquecimento gera erros no algoritmo. **Algoritmos sempre devem ser testados**, e, ao ser percebido algum erro, deve ser refeito, e novamente testado, até que seja considerado correto.
3. O **algoritmo ideal** é o **mais completo** e ao mesmo tempo **menor**.



Visão Esquemática da Construção de Algoritmos



Finalidade dos Algoritmos

Os algoritmos mostram uma sequência lógica que deve ser seguida passo a passo para a resolução de um problema.

Em T.I., a programação exige que o primeiro passo seja um algoritmo do referido programa a ser construído.

Este algoritmo não diz respeito à linguagem de programação que será utilizada na construção do programa. (Java, .NET, PHP, Python, etc...)

Ele deve ser construído em **diagramas de blocos**, e depois, escrito em **texto**, de forma que possa, posteriormente, ser entendido e programado em qualquer linguagem de programação, por qualquer programador.



Estruturação de Algoritmos

A construção de algoritmos é a etapa mais importante em programação. Quando já se sabe a sequência de etapas a realizar, a codificação destes comandos em uma linguagem de programação torna-se mais fácil.

A principal dificuldade do programador iniciante não está no fato de sentar na frente do computador e programar, e sim, saber o quê programar.



Estruturação de Algoritmos

Algoritmos para programas de computador precisam atender aos seguintes objetivos:

- Conversão mais fácil, por utilizar uma notação próxima das linguagens de programação
- Objetividade, evitando interpretação ambígua.
- Ser fácil de entender por um programador.
- E o principal: ter foco no problema.



Lógica de Programação

Construir um algoritmo implica em montar uma solução para um problema.

Como esta solução é feita seguindo uma ordem, o uso do **raciocínio lógico** é importante para chegar a uma solução correta.

Lógica é o estudo das leis do pensamento, determinando se alguma coisa é ou não verdadeira.

A **lógica de programação** é a técnica de encadear pensamentos para atingir determinado objetivo.

Estudar a construção de algoritmos, portanto, também inclui conhecer e praticar a lógica de programação, aprendendo a montar soluções corretas.



Diagrama de Blocos

Iremos utilizar este tipo de representação gráfica de algoritmos, o que facilita a visualização e entendimento do proposto.

Os diagramas de blocos devem, preferencialmente, ser desenhados ao centro da folha, com interpretação de cima para baixo. Em segunda opção, aceita-se o direcionamento da esquerda para a direita.



Utilizaremos como software para a edição de diagramas de blocos, o aplicativo **Dia Portable**.

Este é um **software livre**, e pode ser obtido no seguinte endereço:

<https://br.ccm.net/download/baixaki-2993-dia-portable>



Diagrama de Blocos (Cont.)

Tela do ***Dia Portable***:

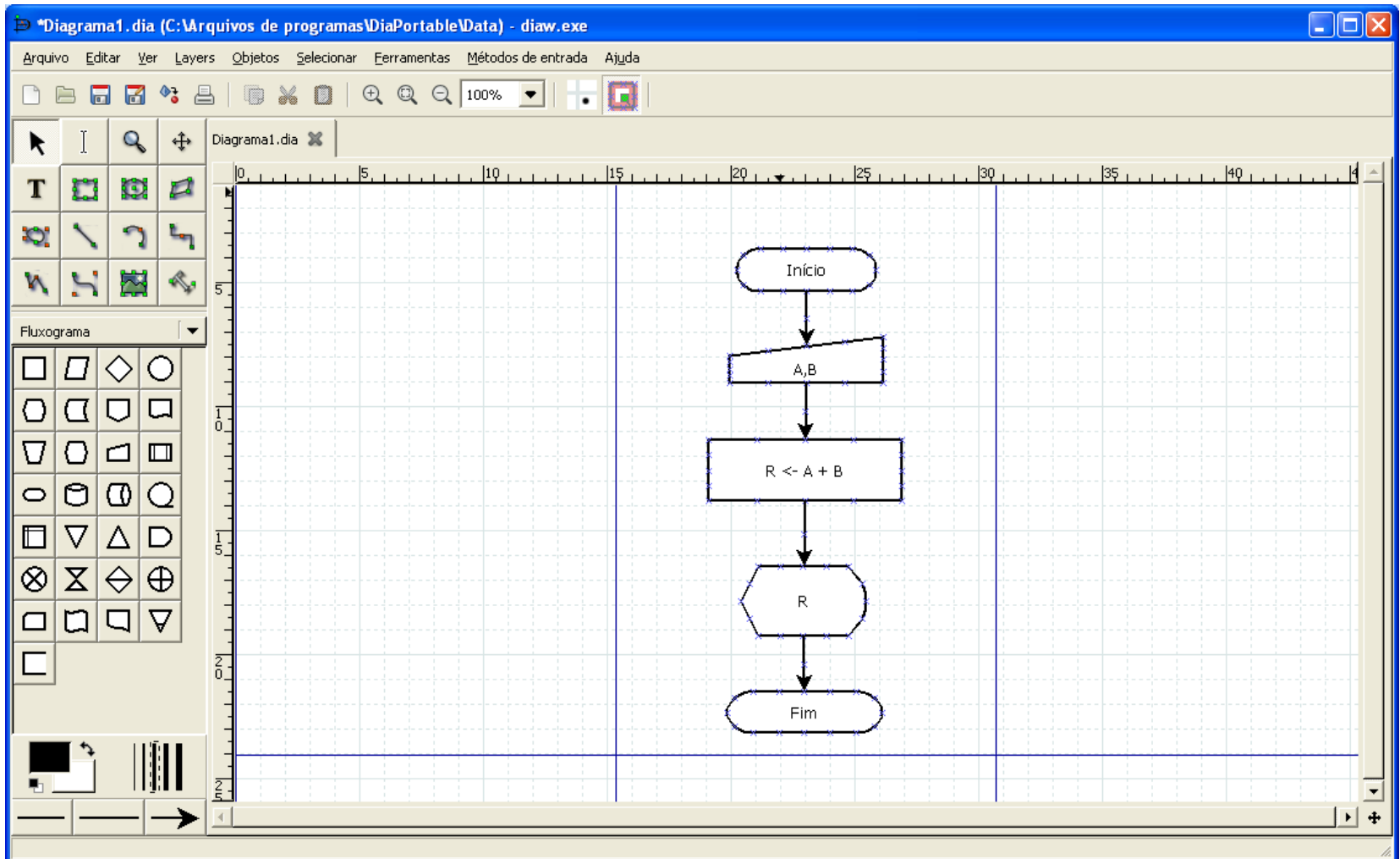


Diagrama de Blocos (Cont.)

Conheça os símbolos mais comuns utilizados em diagramas de blocos:





Símbolo	Significado	Descrição
	Terminal ou Interruptor de Terminal	Representa a definição de início e fim do fluxo lógico de um programa. Também é utilizado na definição de sub-rotinas de procedimento ou de função.
	Entrada Manual	Representa a entrada manual de dados, normalmente efetuada em um teclado de computador.
	Processamento	Representa a execução de uma operação ou grupo de operações que estabelecem o resultado de uma operação lógica ou matemática.
	Exibir	Representa a execução de operação de saída visual de dados em um monitor de computador.



Diagrama de Blocos (Cont.)

Continuação (símbolos mais comuns utilizados em diagramas de blocos):

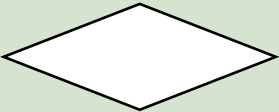

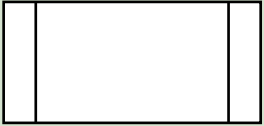

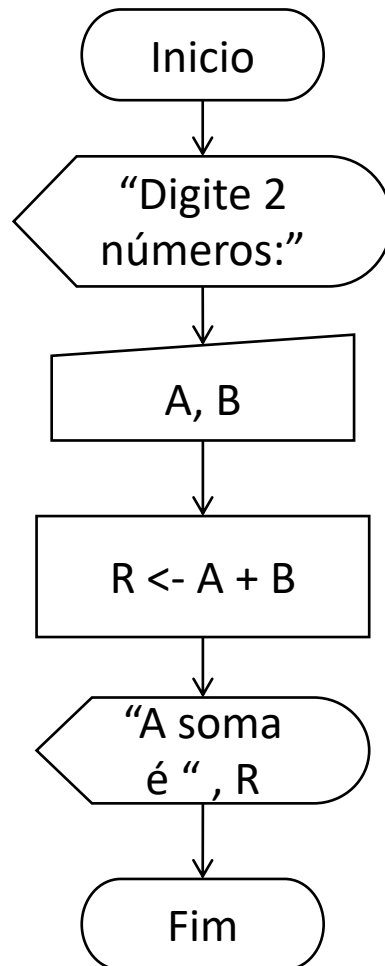
Símbolo	Significado	Descrição
	Decisão	Representa o uso de desvios condicionais para outros pontos do programa de acordo com situações variáveis.
	Preparação	Representa a modificação de instruções ou grupo de instruções existentes em relação à ação de sua atividade subsequencial.
	Processo Predefinido	Definição de um grupo de operações estabelecidas como uma sub-rotina de processamento anexa ao diagrama de blocos.
	Conector	Representa a entrada ou saída em outra parte do diagrama. Pode ser usado na definição de quebras de linha e na continuação da execução de decisões.



Diagrama de Blocos (Cont.)

Os símbolos devem ser interligados por setas, que representam a ação de vínculo existente entre os vários símbolos de um diagrama de blocos.



Pseudocódigo

Como visto, o diagrama de blocos é a primeira forma de notação gráfica.

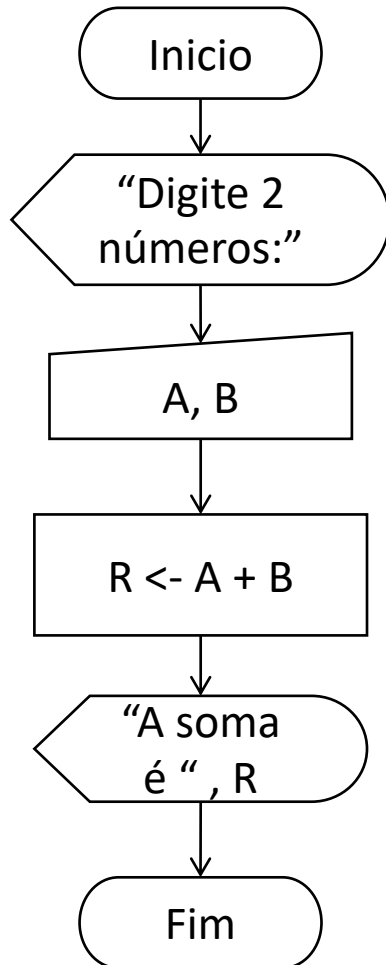
A outra etapa consiste em transcrever o diagrama de uma forma narrativa, denominada **pseudocódigo**, também conhecida como **Português Estruturado** ou chamada por alguns de **Portugol**.

A transcrição narrativa de um diagrama de blocos é baseada em uma técnica de escrita denominada **PDL (Program Design Language – Linguagem de Projeto de Programação)**, usada como referência genérica para uma linguagem de projeto de programação, tendo como finalidade mostrar uma notação para elaborar algoritmos, os quais serão utilizados na **definição, criação e desenvolvimento** de uma linguagem computacional, como *C, C++, C#, PHP, Python, JAVA*, entre outras.



Pseudocódigo (Cont.)

Em seguida, é apresentada a transcrição narrativa do diagrama apresentado anteriormente:

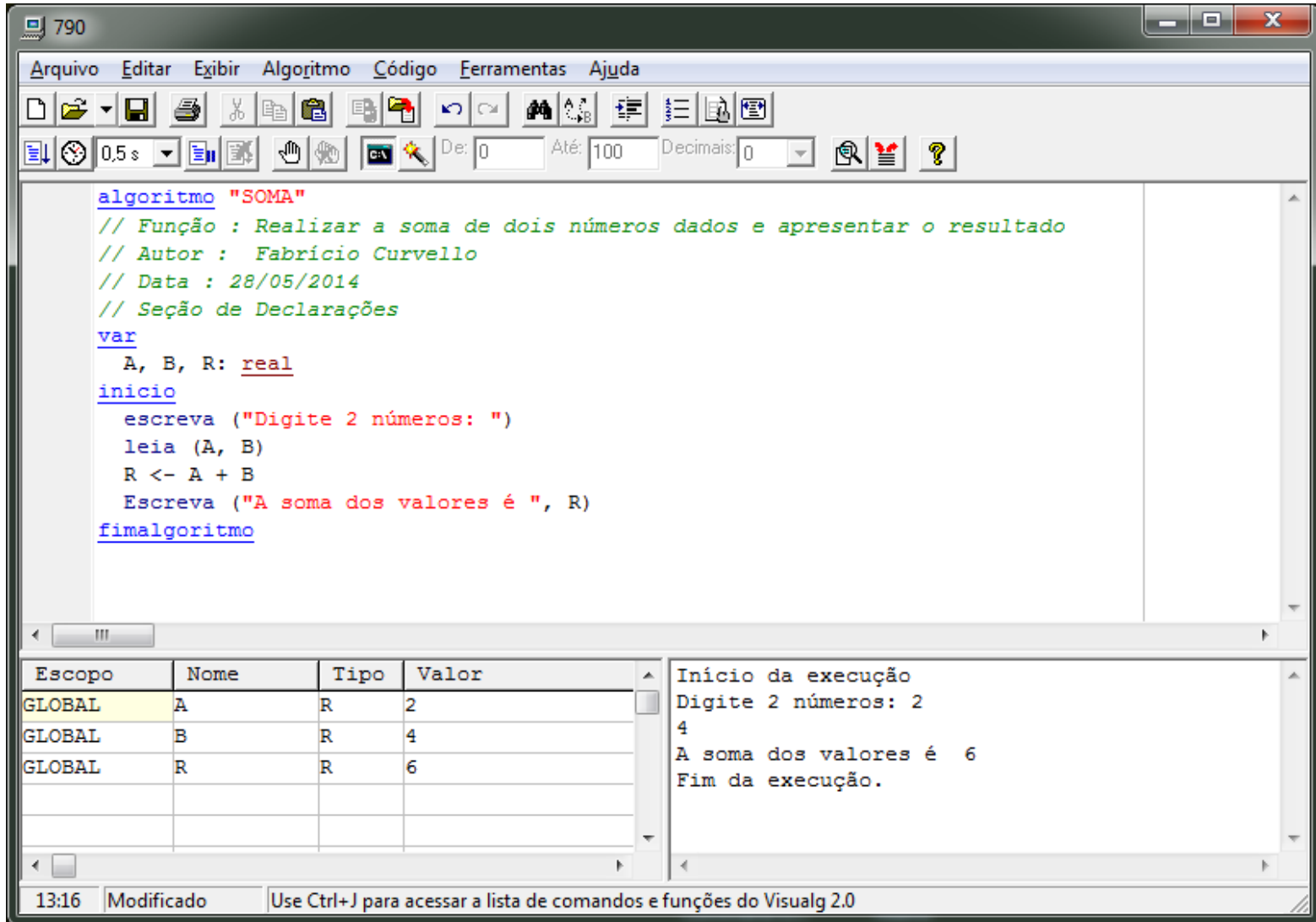


```
algoritmo "SOMA"
var
    A, B, R: real
inicio
    Escreva ("Digite 2 números:")
    Leia (A, B)
    R <- A + B
    Escreva ("A soma é ", R)
finalgoritmo
```



Pseudocódigo (Cont.)

Veja abaixo um exemplo da tela do aplicativo **VisuAlg**.



The screenshot shows the VisuAlg application window. The title bar indicates the window is titled "790". The menu bar includes "Arquivo", "Editar", "Exibir", "Algoritmo", "Código", "Ferramentas", and "Ajuda". The toolbar contains various icons for file operations, editing, and execution. Below the toolbar, there are input fields for "De: 0", "Até: 100", and "Decimais: 0".

The main editor area contains the following pseudocode:

```
algoritmo "SOMA"  
  // Função : Realizar a soma de dois números dados e apresentar o resultado  
  // Autor : Fabrício Curvello  
  // Data : 28/05/2014  
  // Seção de Declarações  
  var  
    A, B, R: real  
  inicio  
    escreva ("Digite 2 números: ")  
    leia (A, B)  
    R <- A + B  
    Escreva ("A soma dos valores é ", R)  
  fimalgoritmo
```

At the bottom of the window, there is a table with 4 columns: "Escopo", "Nome", "Tipo", and "Valor". The table contains the following data:

Escopo	Nome	Tipo	Valor
GLOBAL	A	R	2
GLOBAL	B	R	4
GLOBAL	R	R	6

To the right of the table, there is a text area showing the execution log:

```
Início da execução  
Digite 2 números: 2  
4  
A soma dos valores é 6  
Fim da execução.
```

The status bar at the bottom shows the time "13:16", the word "Modificado", and the text "Use Ctrl+J para acessar a lista de comandos e funções do Visualg 2.0".



Código

Este curso vai trabalhar diretamente a estrutura de algoritmo dentro de uma linguagem de programação profissional.

Desta forma, todos os conceitos de algoritmo serão trabalhados, e ao mesmo tempo, uma linguagem de programação servirá de base para o desenvolvimento dos códigos.

Vamos utilizar para estudos de códigos e conhecimentos de algoritmos a linguagem de programação JAVA.

Será necessário preparar todo o ambiente de desenvolvimento em JAVA. Todos os detalhes de como preparar este ambiente estão no documento *LP - Extra 01 - Preparação de ambiente JAVA.pdf*



Dúvidas?

