

INSTITUTO FEDERAL
Maranhão

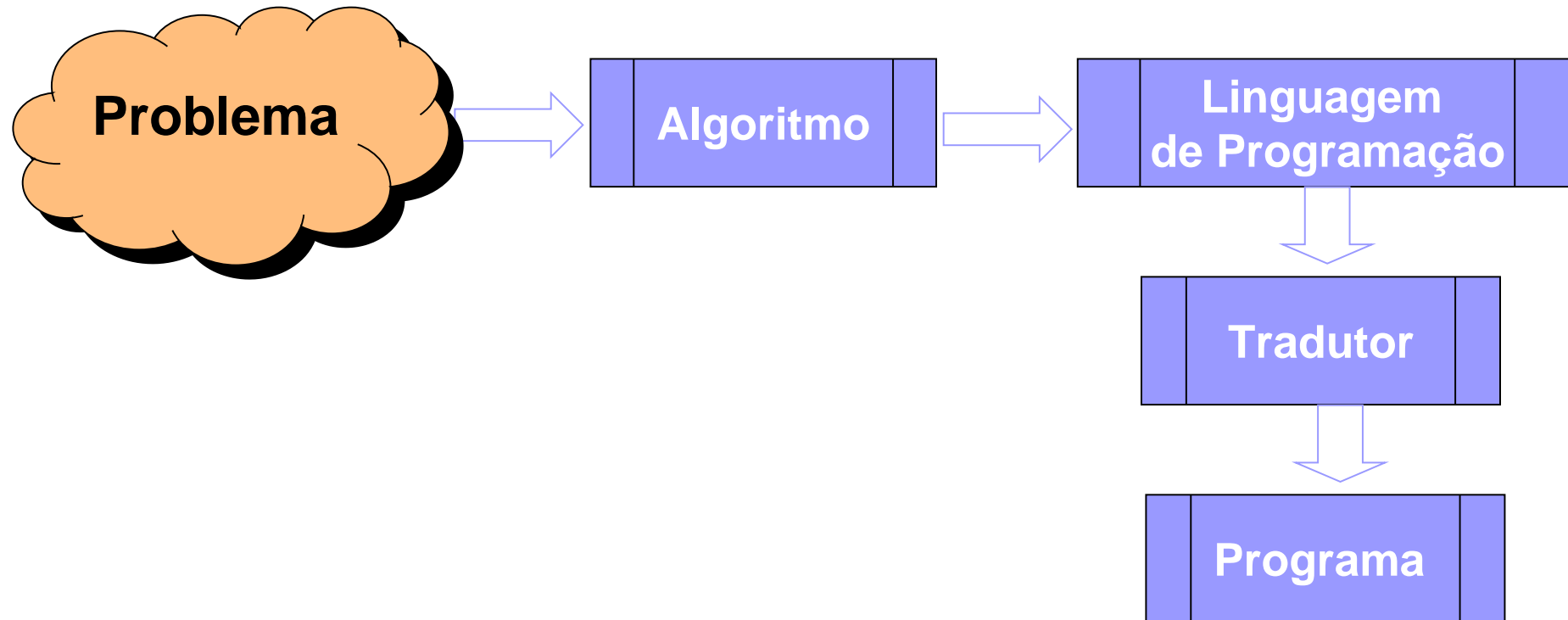
Introdução à Ciência da Computação

Curso: Sistemas de Informação
Profa. Evaldinolia Gilbertoni Moreira
Email: prof.evaldinolia@acad.ifma.edu.br

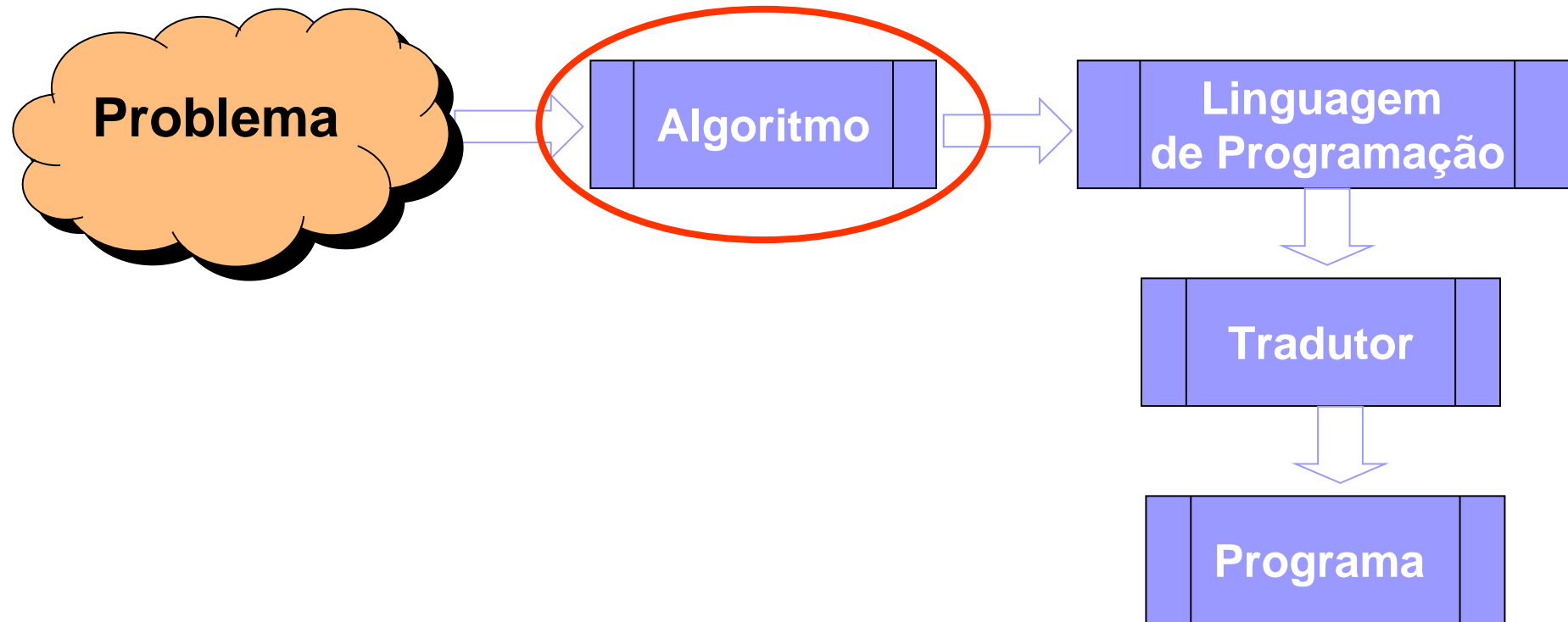
Introdução

- ❑ Ciclo da programação
- ❑ Introdução à Algoritmos

Ciclo da Programação



Ciclo da Programação



O que é algoritmo?

É a descrição de um conjunto de **ações** que, obedecidas, resultam numa sucessão finita de passos atingindo um objetivo.

Ação - é um acontecimento que a partir de um estado inicial, após um período de tempo finito, produz um estado final previsível e bem definido.

Objetivo: Algum processo de automação.

Automação é o processo em que uma tarefa deixa de ser desempenhada pelo homem e passa a ser realizada por máquinas, sejam estas, dispositivos mecânicos, eletrônicos (como computadores) ou de natureza mista.

Algoritmos

*“Ação ou acontecimento que, a partir de um **estado inicial**, após um **período de tempo** finito, produz um **estado final** previsível e bem-definido. Portanto, um algoritmo é a descrição de um conjunto de comandos que, obedecidos, resultam numa sucessão finita de ações.”*

(FARRER, 1999)

“Um algoritmo é uma lista de instruções que, quando executadas, transformam dados de entrada até a saída. As instruções são um conjunto finito de etapas que podem ser executadas, numa ordem precisa, por um mecanismo determinista. Quando estas etapas são efetivamente executadas, a execução deve terminar após um tempo finito.”

(HOLLOWAY, 2006)

Por que precisamos de algoritmos?

- **Conceber uma solução para um dado problema**, independente de uma linguagem computacional específica.
- É uma das maiores dificuldades, mas também um dos desafios mais atrativos para os principiantes em programação de computadores.

“A noção de algoritmo é básica para toda a programação de computadores”.

*[KNUTH - Professor da Universidade de Stanford,
autor da coleção “The art of computer programming”]*

Principais características dos algoritmos

- Todo algoritmo deve apresentar algumas características básicas:
 - Não dar margem à dupla interpretação (não ser ambíguo);
 - Ter capacidade de receber dado(s) de entrada;
 - Poder gerar informações de saída;
 - Ser efetivo, ou seja todas as etapas especificadas no algoritmo devem ser **alcançáveis em um tempo finito**;
 - Ter fim.

Linguagem Algorítmica

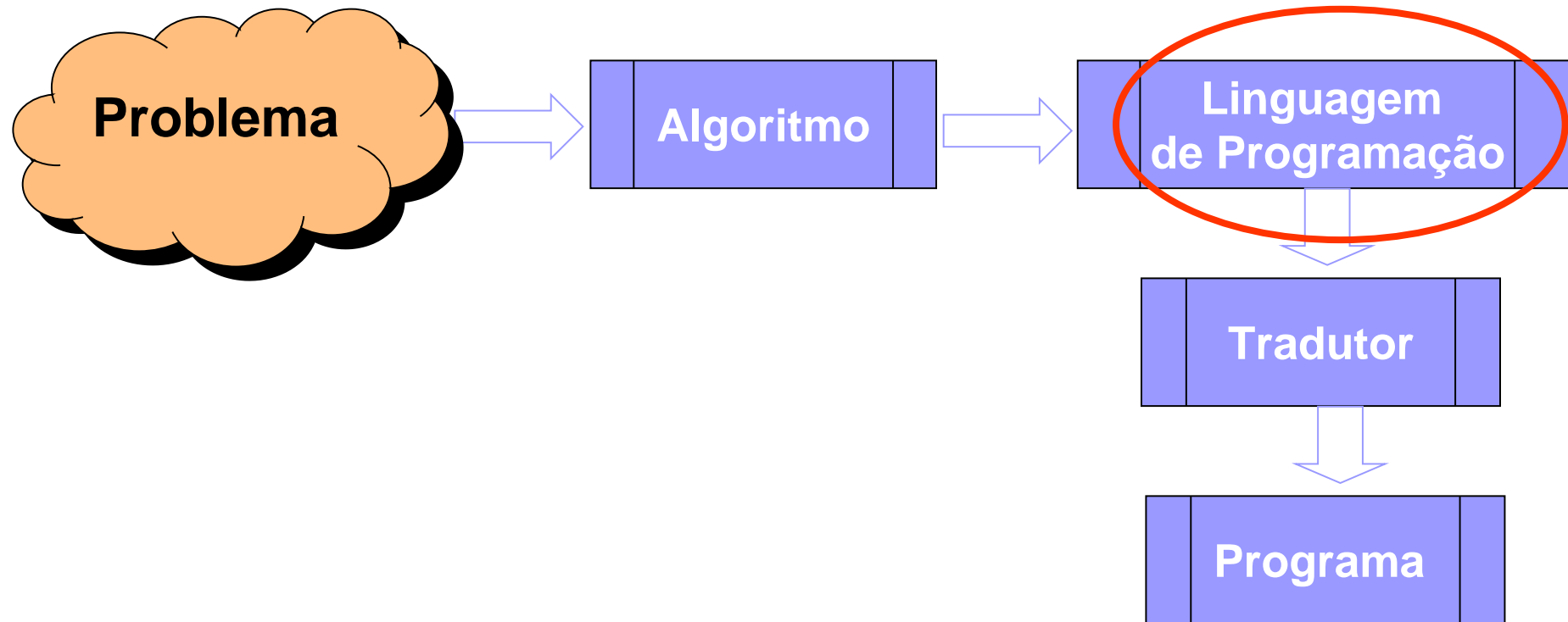
- **Vantagens:**

- Independência da solução.
- Trata-se de **solução lógica** apenas.
- Não interessa quem vai executar.
- Usa o português como linguagem de base;
- Pode-se definir quais dados serão usados e como utilizá-los;
- Passagem quase imediata do algoritmo para uma linguagem de programação qualquer.

- **Desvantagens:**

- Exige a definição de uma linguagem não real para trabalho;
- Não padronizado.

Ciclo da Programação



Linguagem de Programação

- É um vocabulário e um conjunto de regras sintáticas e semânticas usadas para escrever programas de computador.
- Cada linguagem possui um conjunto único de palavras-chave (palavras que ela reconhece) e uma sintaxe (regras) específica para organizar as instruções dos programas.
- Servi de um meio de comunicação entre computadores e humanos.

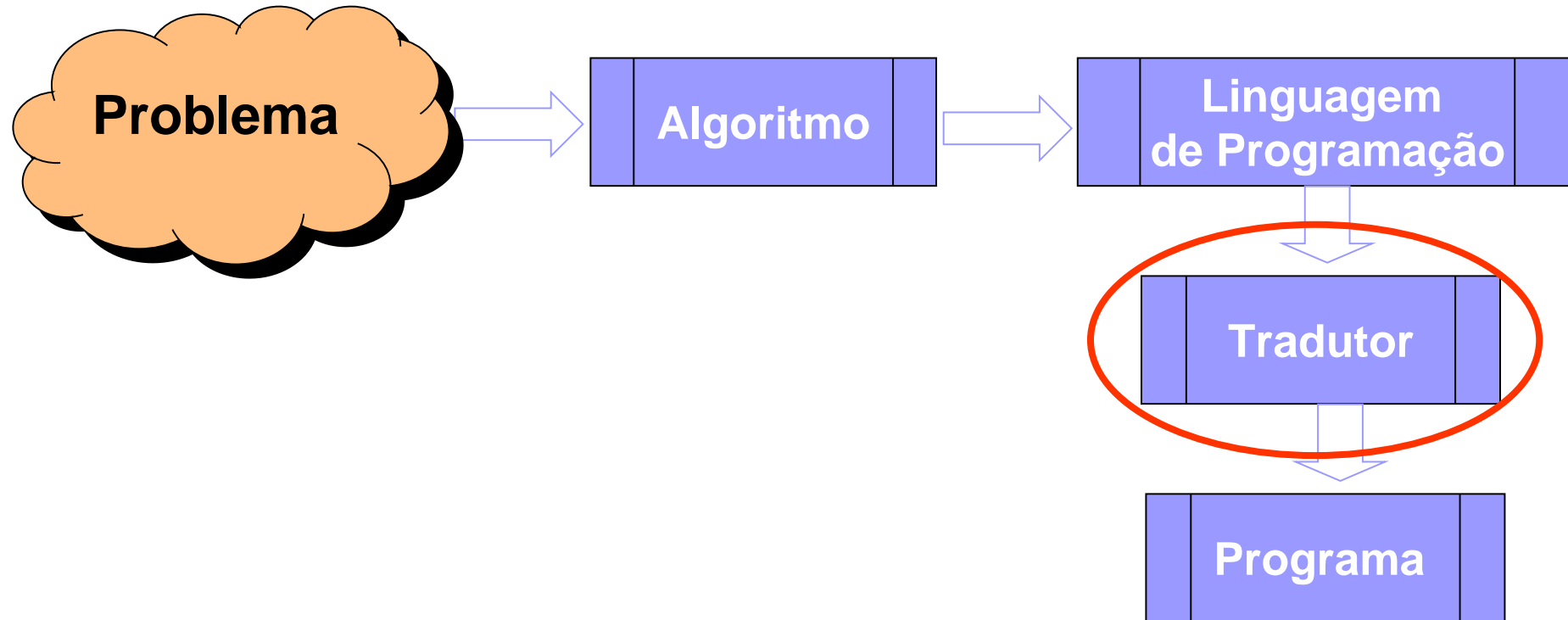
Linguagem de Programação

Linguagem de máquina	Linguagem de baixo nível	Linguagem de alto nível
0010 0001 1110	LOAD R1, val1	val2 = val1 + val2
0010 0010 1111	LOAD R2, val2	
0001 0001 0010	ADD, R1, R2	
0011 0001 1111	STORE R1, val2	

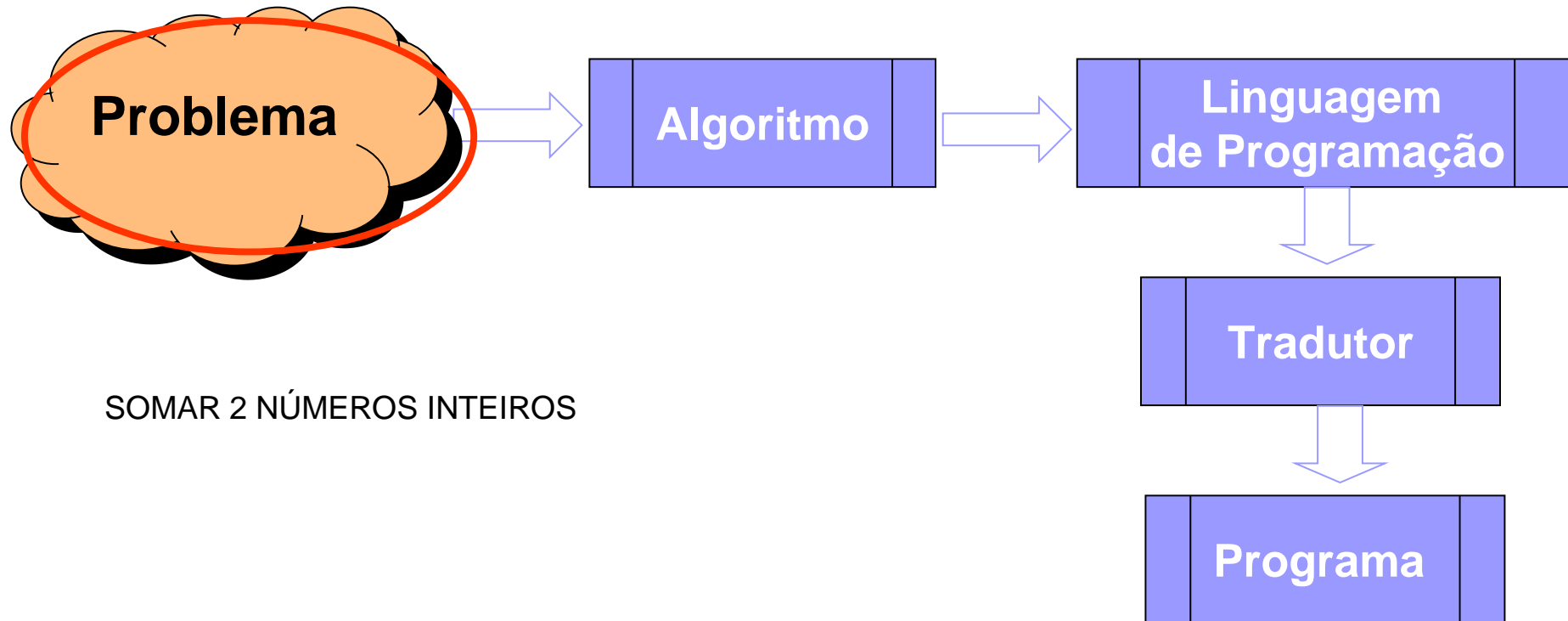
Para saber mais

Unidade II - Mathias, Ivo Mario; Algoritmos e programação I/ Ivo Mario Mathias. Ponta Grossa: UEPG/ NUTEAD, 2017.

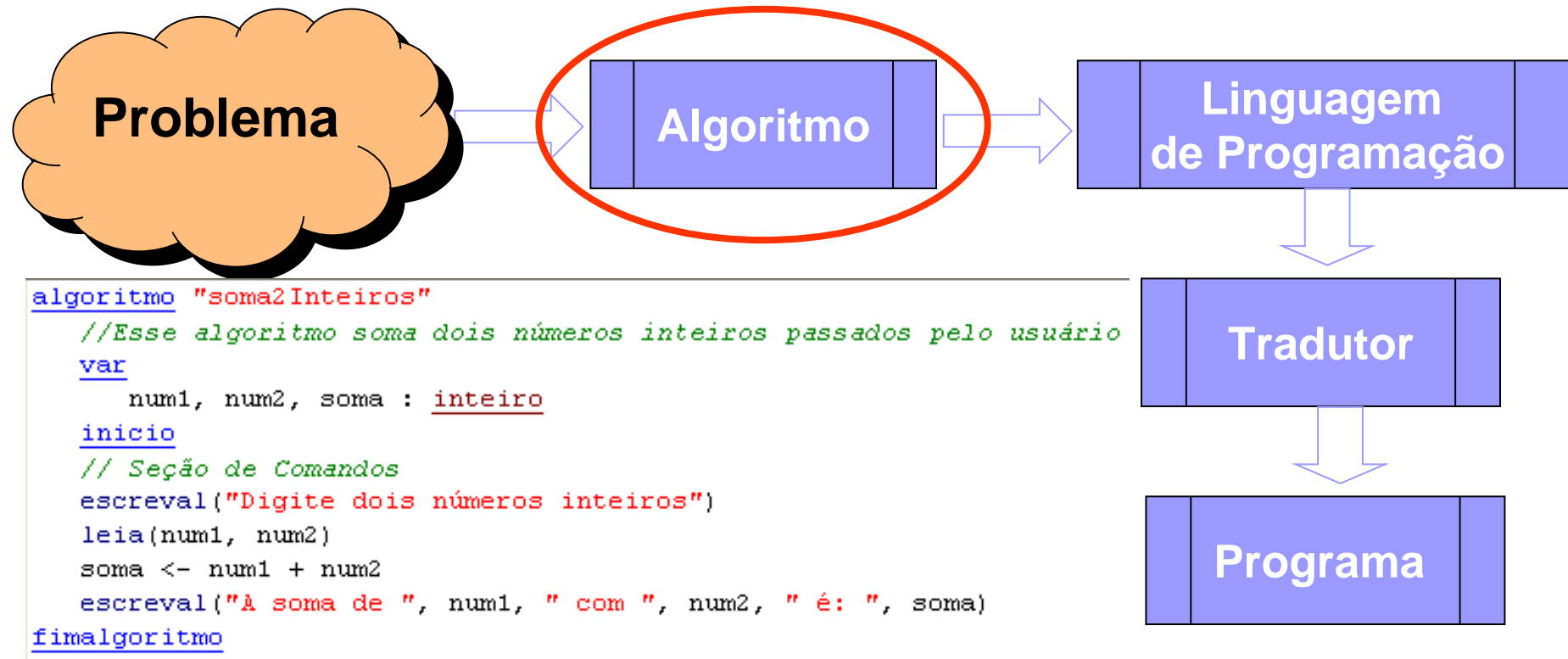
Ciclo da Programação



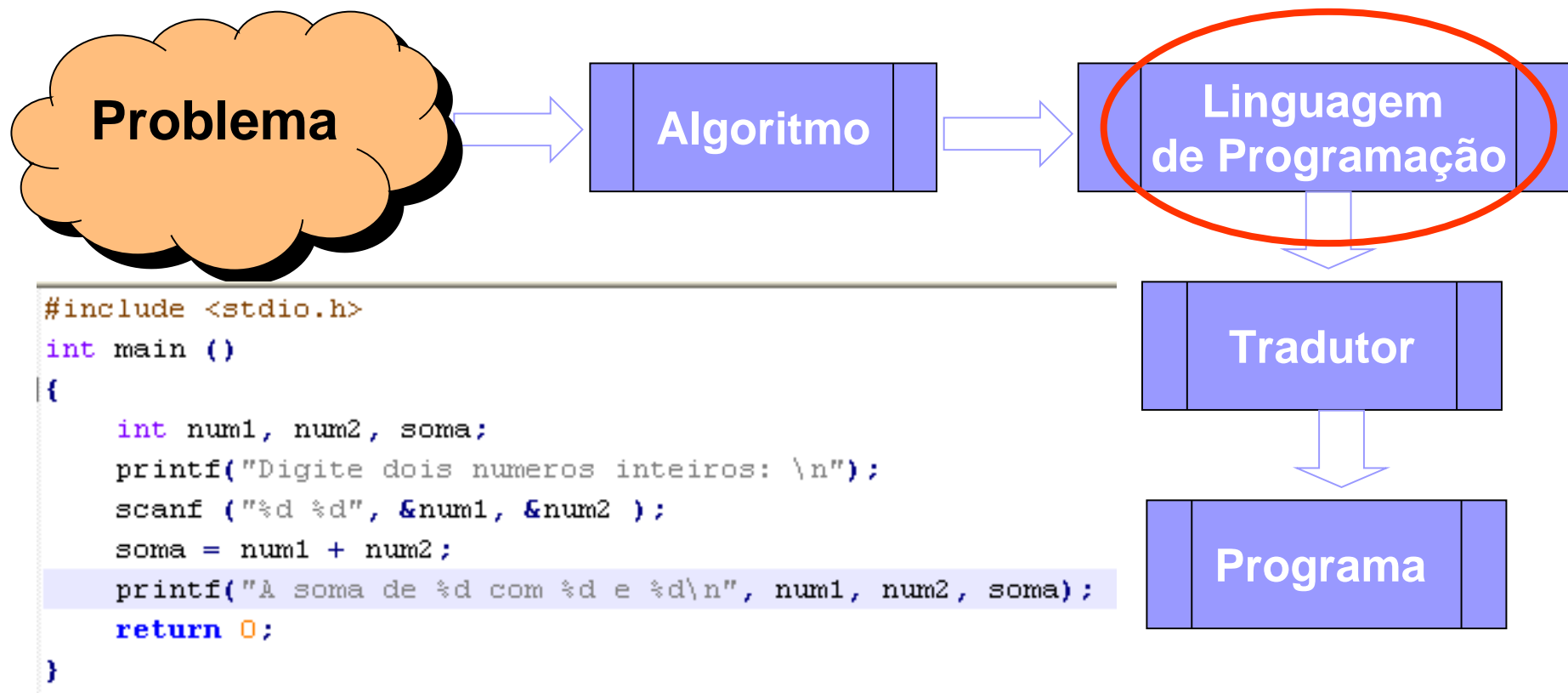
Ciclo da Programação



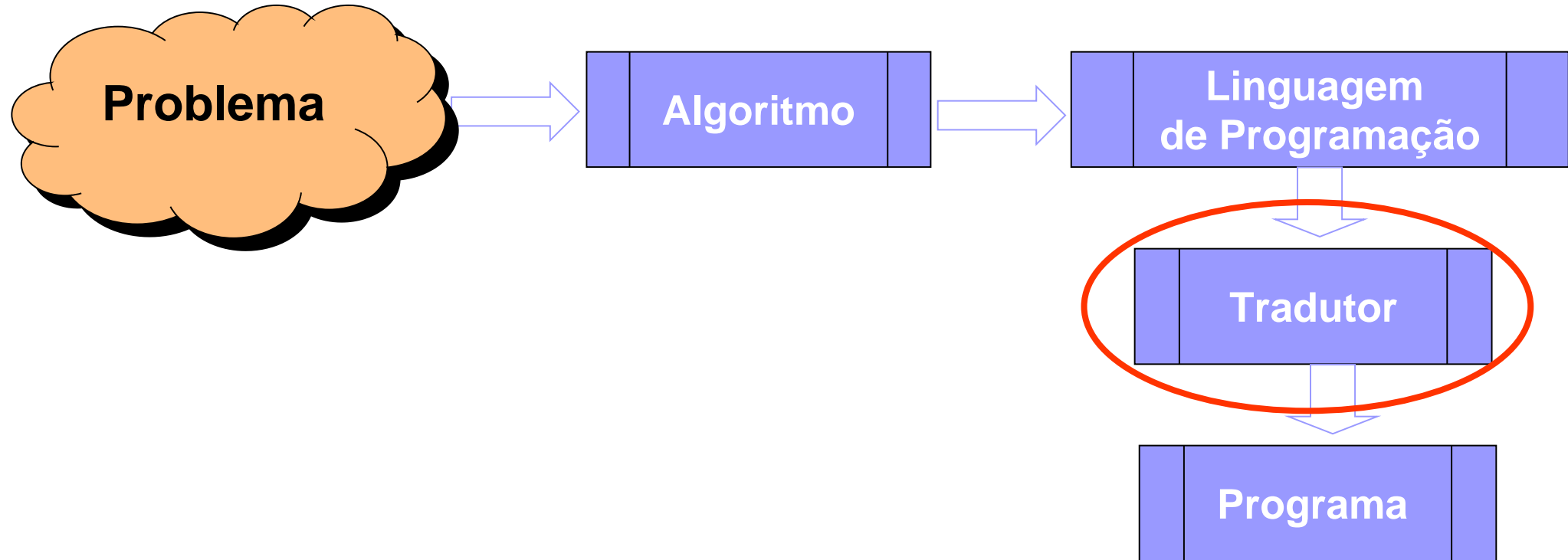
Ciclo da Programação



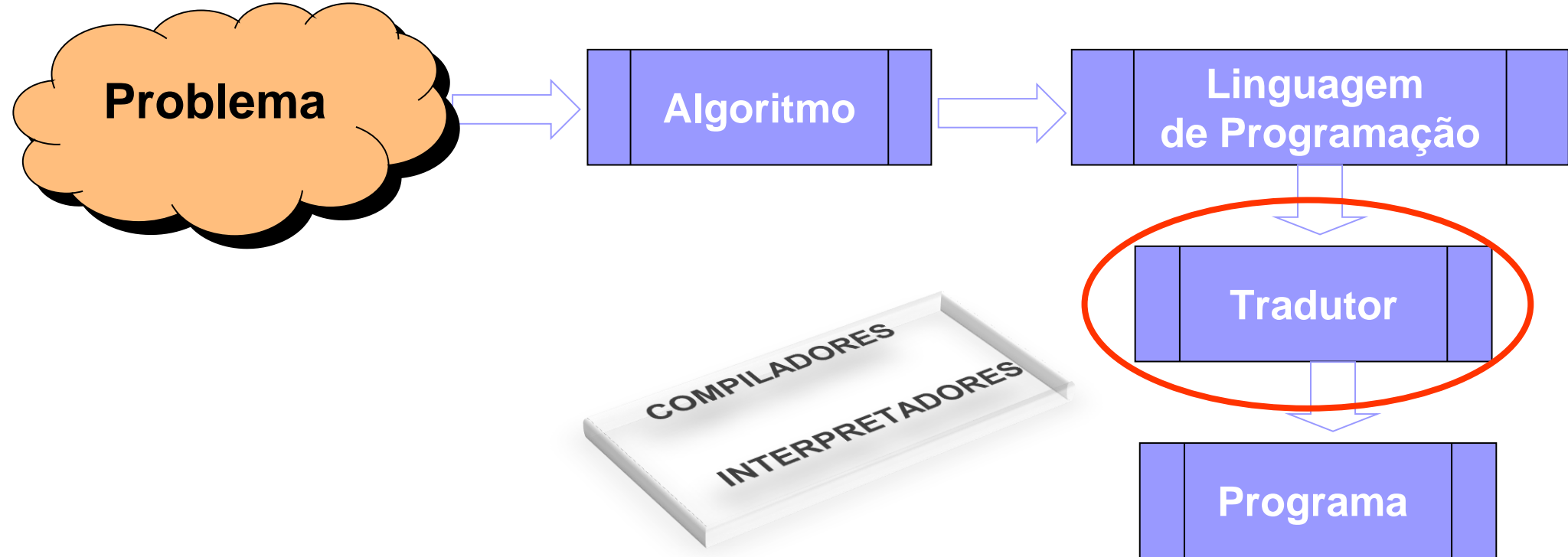
Ciclo da Programação



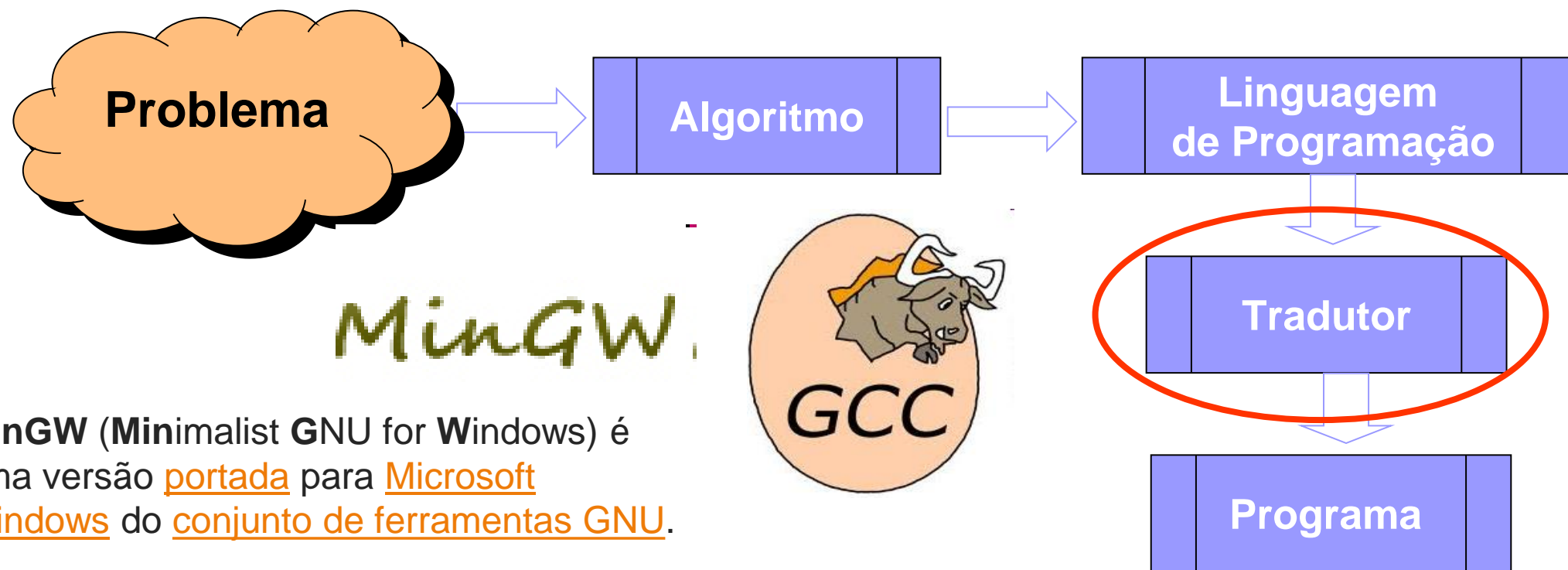
Ciclo da Programação



Ciclo da Programação



Ciclo da Programação

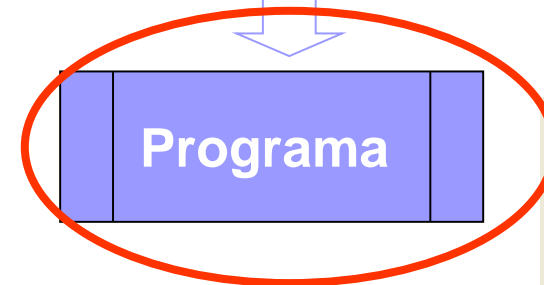
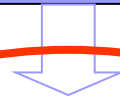
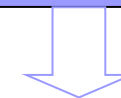
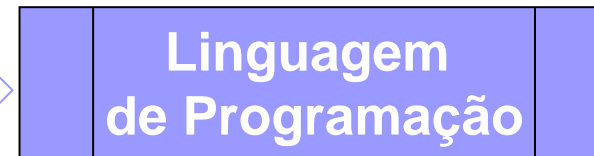
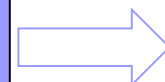
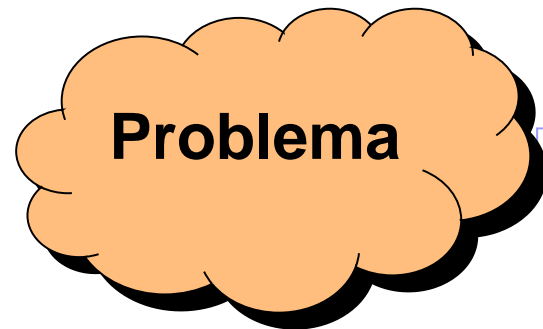


MinGW (Minimalist **GNU** for **Windows**) é uma versão portada para Microsoft Windows do conjunto de ferramentas GNU.

GNU Compiler Collection (chamado usualmente por **GCC**) é um conjunto de compiladores de linguagens de programação produzido pelo projecto GNU para construir um sistema operativo semelhante ao Unix livre.

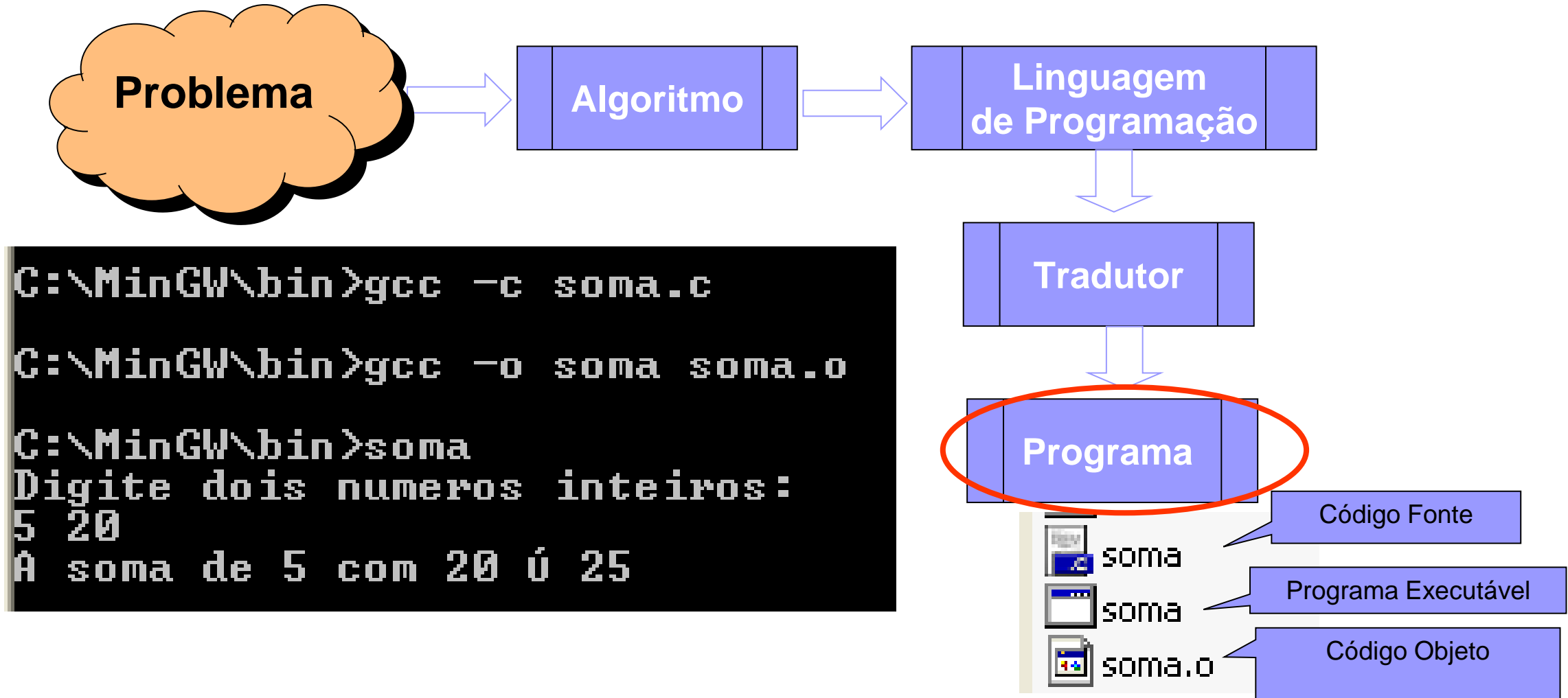
<https://www.techtudo.com.br/artigos/noticia/2012/12/saiba-o-que-e-gnu-e-qual-seu-papel-no-movimento-do-software-livre.html>

Ciclo da Programação



```
C:\MinGW\bin>gcc -c soma.c  
C:\MinGW\bin>gcc -o soma soma.o  
C:\MinGW\bin>soma  
Digite dois numeros inteiros:  
5 20  
A soma de 5 com 20 é 25
```

Ciclo da Programação



Programa

Sequência de instruções, escritas em uma *Linguagem de Programação*, a serem realizadas por um computador que visa atingir um objetivo (resolver um problema).

Etapas para a construção de um programa:

- **Análise:** compreensão do problema. Identificação da entrada, das operações de processamento e da saída.
- **Algoritmo:** descrição do problema e da sua solução.
- **Codificação:** tradução do algoritmo para uma linguagem de programação.

Método para a construção de Algoritmos

Segundo ASCENCIO e CAMPOS (2002), os passos para a construção de algoritmos são os seguintes:

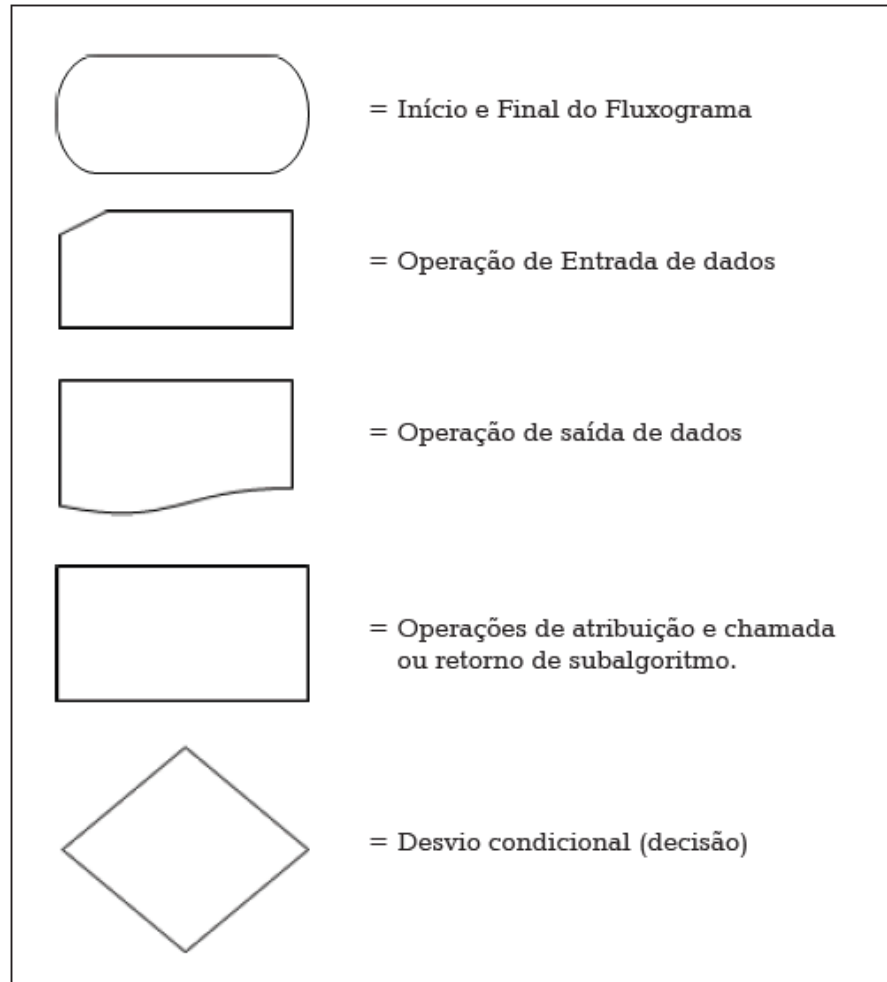
1. Ler atentamente o enunciado, destacando os pontos mais importantes;
2. Definir os dados de entrada;
3. Definir o processamento;
4. Definir os dados de saída;
5. Construir o algoritmo uma das técnicas descritas a seguir;
6. Testar o algoritmo realizando simulações.

Técnicas de Representação de Algoritmos

Há muitas formas, padronizadas e não padronizadas, para que seja feita a representação de um Algoritmo, escolhemos quatro para abordarmos em nosso curso:

- **Descrição Narrativa**
- **Fluxograma**
- **Linguagem Algorítmica (Pseudocódigo, Português Estruturado ou Portugol)**

Fluxograma - Símbolos

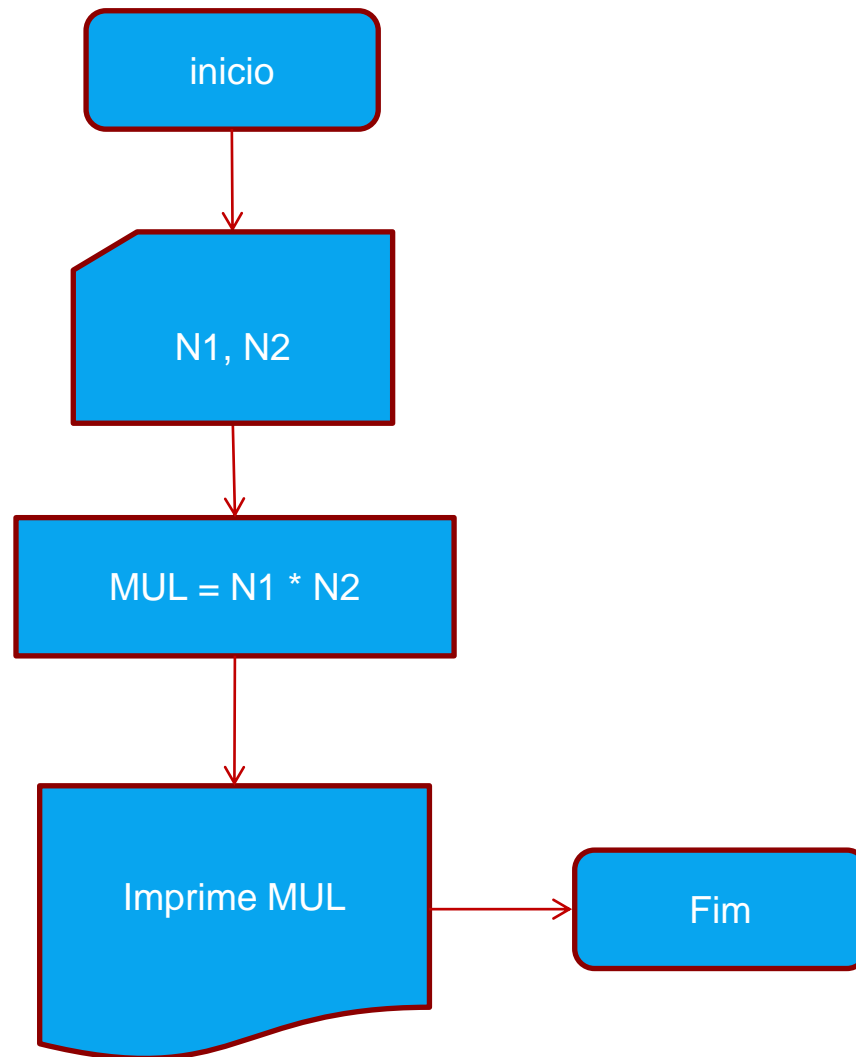


Exemplo 1- Descrição Narrativa

Crie um algoritmo para mostrar o resultado da multiplicação de dois números

- Passo 1: Receber os dois números que serão multiplicados
- Passo 2: Multiplicar os números
- Passo 3: Mostrar o resultado obtido na multiplicação

Exemplo 1 - Fluxograma



Exemplo 1 — PSEUDOCÓDIGO - LINGUAGEM ESTRUTURADA - PORTUGOL

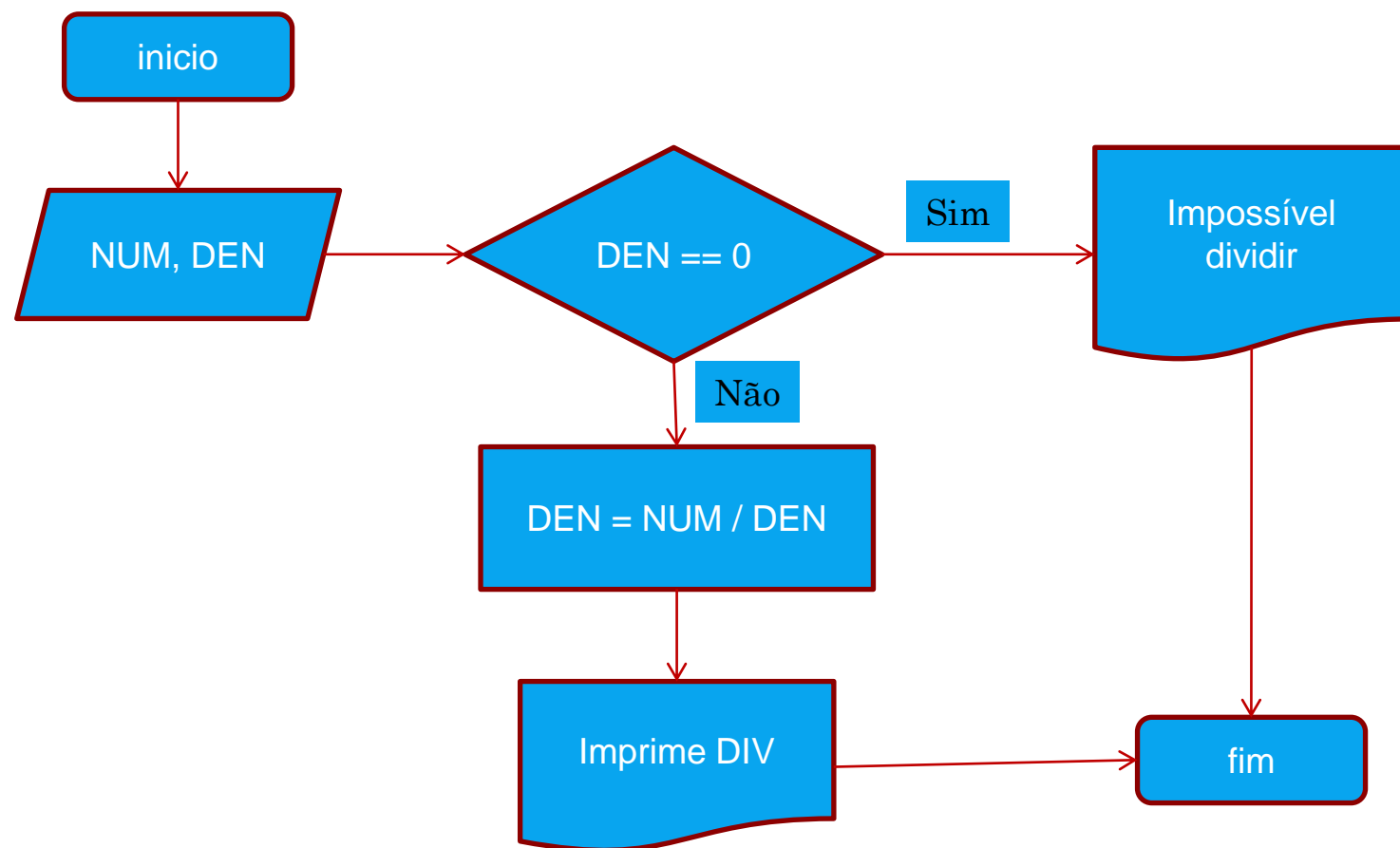
```
Algoritmo Multiplicação_dois_Numeros;  
Var  
N1, N2, MUL : real;  
Início  
Leia N1, N2;  
MUL := N1 * N2;  
Escreva MUL;  
Fim.
```

Exemplo 2

Faça um algoritmo que apresente como saída a divisão de dois números

- Passo 1: receber os dois números
- Passo 2: Se o denominador (segundo número) for igual a zero a divisão não poderá ser feita; caso contrário dividir os dois números e mostrar o resultado

Exemplo 2



Exemplo 1 — PSEUDOCÓDIGO - LINGUAGEM ESTRUTURADA - PORTUGOL

```
Algoritmo Divisão_dois_Numeros;  
Var  
NUM, DEN, DIV : real;  
Início  
Leia NUM, DEN;  
Se (DEN == 0) então  
    escreva ("Impossível dividir");  
Senão  
    DIV := NUM / DEN;  
    Escreva DIV;  
fim_se;  
Fim.
```

Esquema simplificado do funcionamento dos programas de computador



Entrada – dados que são enviados ao computador, por meio de seus dispositivos de entrada, tais como: teclado, mouse, tela do monitor de vídeo sensível ao toque, discos, pendrive, etc;

Processamento – a UCP reconhece os dados que foram transmitidos pelos dispositivos de entrada e efetua o processamento deles conforme as diretivas do programa;

Saída – após o processamento dos dados e também conforme as diretivas do programa, os dispositivos de saída são acionados, por exemplo: tela do monitor de vídeo, impressora, discos, pendrive, etc.

Controle dinâmico da execução de um algoritmo

Os três conceitos de estruturação de algoritmo descrevem o controle do fluxo dinâmico do algoritmo quando ele é executado, e todos os três devem ser expressos de alguma maneira na descrição estática do algoritmo. (HOOLOWAY, 2006)

- Sequência - especifica uma ordem linear de execução na qual uma tarefa particular é explicitamente indicada para suceder uma outra.
- Iteração - é a execução repetida de um grupo de instruções até que alguma condição seja satisfeita
- Seleção - é a execução seletiva de um grupo de instruções baseada em alguma condição.