


# Curso Superior de Análise e Desenvolvimento de Sistemas

*Disciplina: Algoritmo*  
2009 / 2


Prof. MsC Simone B. K. Aires  
sbkaminski@utfpr.edu.br



# Habilidades

- **Métodos para construção de algoritmos.**
  - **Representação de algoritmos.**
  - **Tipos primitivos de dados:** constantes e variáveis.
  - **Comandos de atribuição:** entrada e saída de dados.
  - **Programas e sub-programas**
    - Programação Estruturada e Modular
    - Procedimentos
    - Funções
- 

# Habilidades

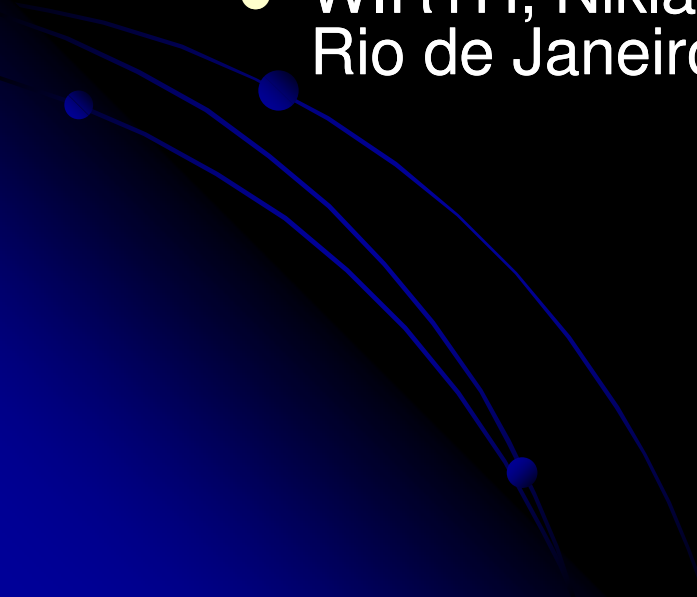
- **Operadores:** aritméticos, relacionais e lógicos.
  - **Estruturas controle de fluxo:**
    - Decisão:** simples, composta e múltipla escolha;
    - Repetição:** teste no início, teste no final, variável de controle;
  - **Estrutura de dados:**
    - Matrizes unidimensionais e bidimensionais (**vetores e matrizes**)
    - **Registros**
- 

# Referências Bibliográficas

## 2. Algoritmos:

- MANZANO, José Augusto N. G. , OLIVEIRA, Jayr Figueiredo. Estudo Dirigido. Coleção P.D. Editora Érica. 4<sup>a</sup> ed.
- LAGES, Guimarães. Introdução à ciência da computação. Livros Técnicos e Científicos. Editora S.A
- FORBELLONE, André Luiz Villar. BERSPACHER, Henri Frederico. Lógica de Programação - A construção de algoritmos e estruturas de dados. Makron Books, 1993.

# Referências Bibliográficas

- ZIVIANI, Nivio. Projeto de algoritmos: com implementações em Pascal e C. São Paulo: Pioneira Thomson Learning, 2002 267 p. ISBN 8522101744
  - MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. Estudo dirigido de algoritmos. 7. ed. São Paulo: Érica, 2002 220p.
  - WIRTH, Niklaus. Algoritmos e estruturas de dados. Rio de Janeiro: LTC, c1999. 255 p.
- 

# Sistema de Avaliação

- **1ª avaliação – 3,0**
  - Prova de lista de exercícios
  - Atividades práticas
- **2ª avaliação – 3,0**
  - Prova 1ª etapa
  - Atividades práticas
- **3ª avaliação – 3,0**
  - Prova 2ª etapa
  - Atividades práticas
- **Atividade extra – 1,0**
- **Re-avaliação - Substitutiva**

# 1. Abordagem Contextual

- Algoritmo não é a solução do problema, é pois, o caminho para a solução de um problema.
- Existem várias formas de representar um algoritmo.
- O aprendizado de algoritmos não se consegue a não ser através de muitos exercícios.
- Para a definição de um bom algoritmo é necessário desenvolver um raciocínio lógico.

# 1. Abordagem Contextual



# 1. Abordagem Contextual



PLAY

Por favor ajude o homem no barco mover - o lobo , a ovelha e a caixa de repolho ao outro lado do lago.  
**Note isso:** Os lobos comem ovelha & ovelha come repolho quando nenhum homem ao redor de.

NEXT

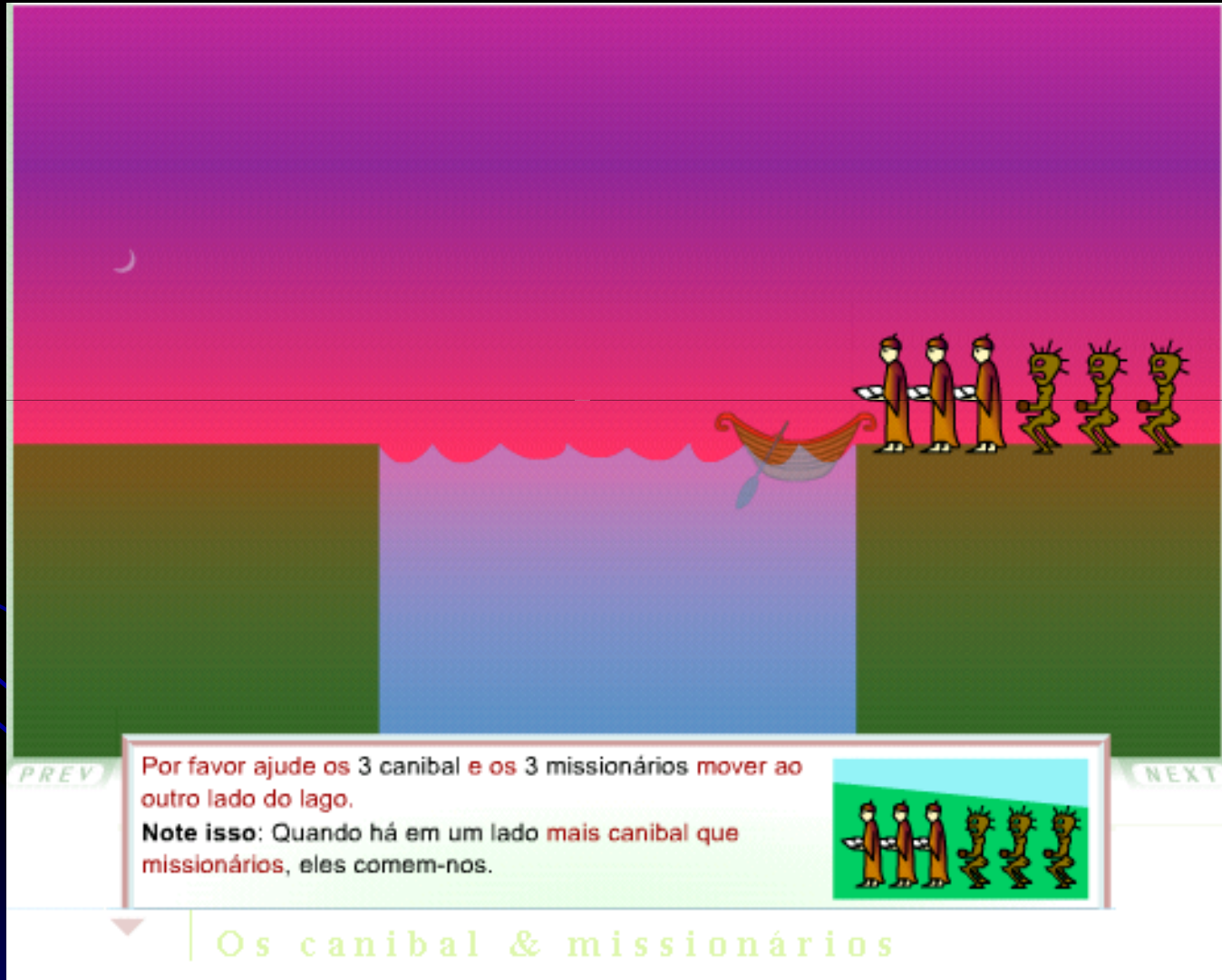
M

R

S


A ovelha de lobo & repolho

# 1. Abordagem Contextual



**PREV** Por favor ajude os 3 canibal e os 3 missionários mover ao outro lado do lago. **NEXT**

**Note isso:** Quando há em um lado mais canibal que missionários, eles comem-nos.



Os canibal & missionários

# 1. Abordagem Contextual



PLAY

PREV

Por favor ajude esta família cruzar ao outro lado da ponte.  
**Note isso: É noite**, então, deve ter uma lâmpada.  
Cada pessoa cruza a ponte numa velocidade diferente: 1 sec 3 sec 6 sec 8 sec & 12 sec.

A ponte pode segurar um máximo de 2 pessoas.  
Um par deve andar junto à razão da pessoa mais lenta.  
A lâmpada suficiente para 30 sec só!!!

NEXT

# 1. Abordagem Contextual (cont.)

## **Algoritmos não se aprendem:**

- *Copiando algoritmos*
- *Estudando algoritmos*

## **Algoritmos só se aprendem:**

- *Construindo algoritmos*
- *Testando algoritmos*

# 1.1. Conceito de algoritmo

- “Algoritmo é um conjunto finito de regras, bem definidas, para a solução de um problema em um tempo finito e com um número finito de passos.”

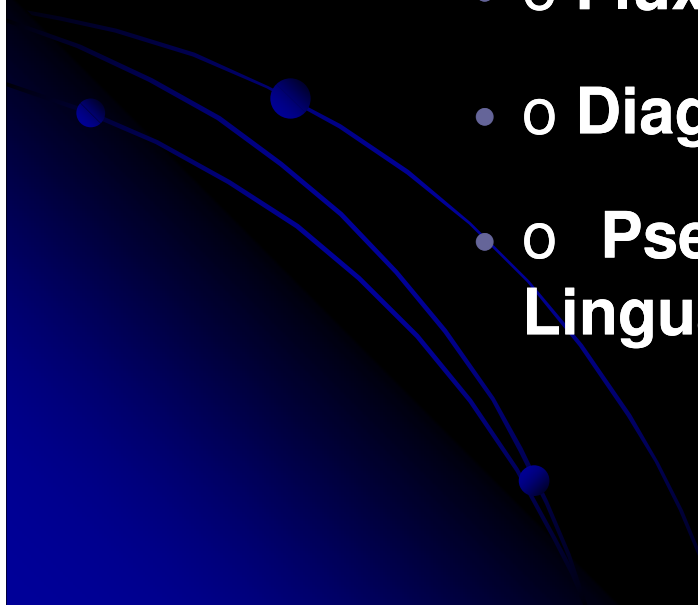


## 1.2. Para se ter um algoritmo, é necessário:

- Que se tenha um número finito de passos
- Que cada passo esteja precisamente definido, sem possíveis ambigüidades
- Que existam zero ou mais entradas tomadas de conjuntos bem definidos
- Que existam uma ou mais saídas
- Que exista uma condição de fim sempre atingida para quaisquer entradas e num tempo finito.



## 2. Formas de representação de um algoritmo

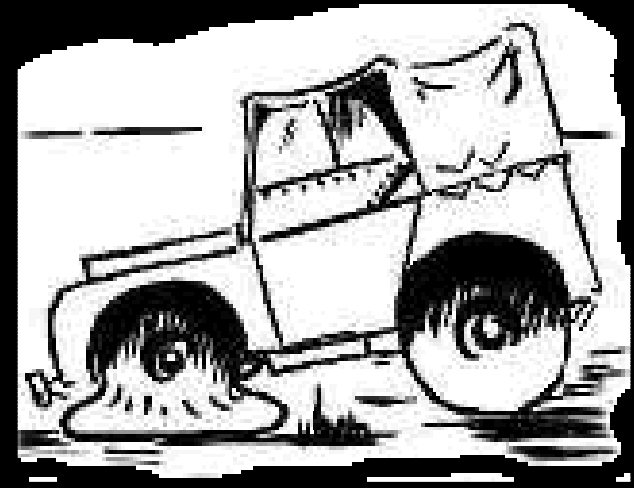
- Dentre as formas de representação de algoritmos mais conhecidas, sobressaltam:
    - a **Descrição Narrativa**
    - o **Fluxograma Convencional**
    - o **Diagrama de Chapin**
    - o **Pseudocódigo**, também conhecido como **Linguagem Estruturada** ou **Portugol**.
- 

## 2.1. Forma Narrativa

- Nesta forma de representação os algoritmos são expressos diretamente em **linguagem natural**. Como por exemplo:

### ↙ Troca de um pneu furado:

- Afrouxar ligeiramente as porcas
- Suspender o carro
- Retirar as porcas e o pneu
- Colocar o pneu reserva
- Apertar as porcas
- Abaixar o carro
- Dar o aperto final nas porcas



## 2.2. Fluxograma Convencional e Diagrama de Blocos

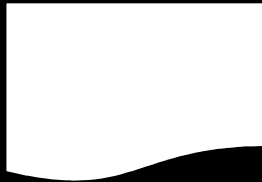
- É uma representação gráfica de algoritmos onde formas geométricas diferentes implicam ações (instruções, comandos) distintos. Tal propriedade facilita o entendimento das idéias contidas nos algoritmos.
- Símbolos utilizados no fluxograma:



= Início e final do fluxograma



= Operação de entrada de dados



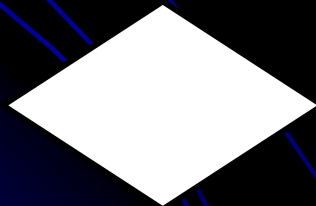
= Operação de saída de dados em impressora



= Operação de saída de dados em vídeo



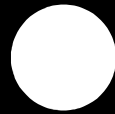
= Operações de atribuição e chamada ou retorno de subalgoritmo



= Decisão



= Seta do Fluxo de Dados



= Conector utilizado quando é preciso particionar o diagrama, colocando uma letra ou número no símbolo para identificar os pares da conexão



= Conector específico para indicar conexão do fluxo de execução em outra página



= Preparação - operações que utilizam uma chave que modifica a execução do programa

## 2.3. Diagrama de Chapin

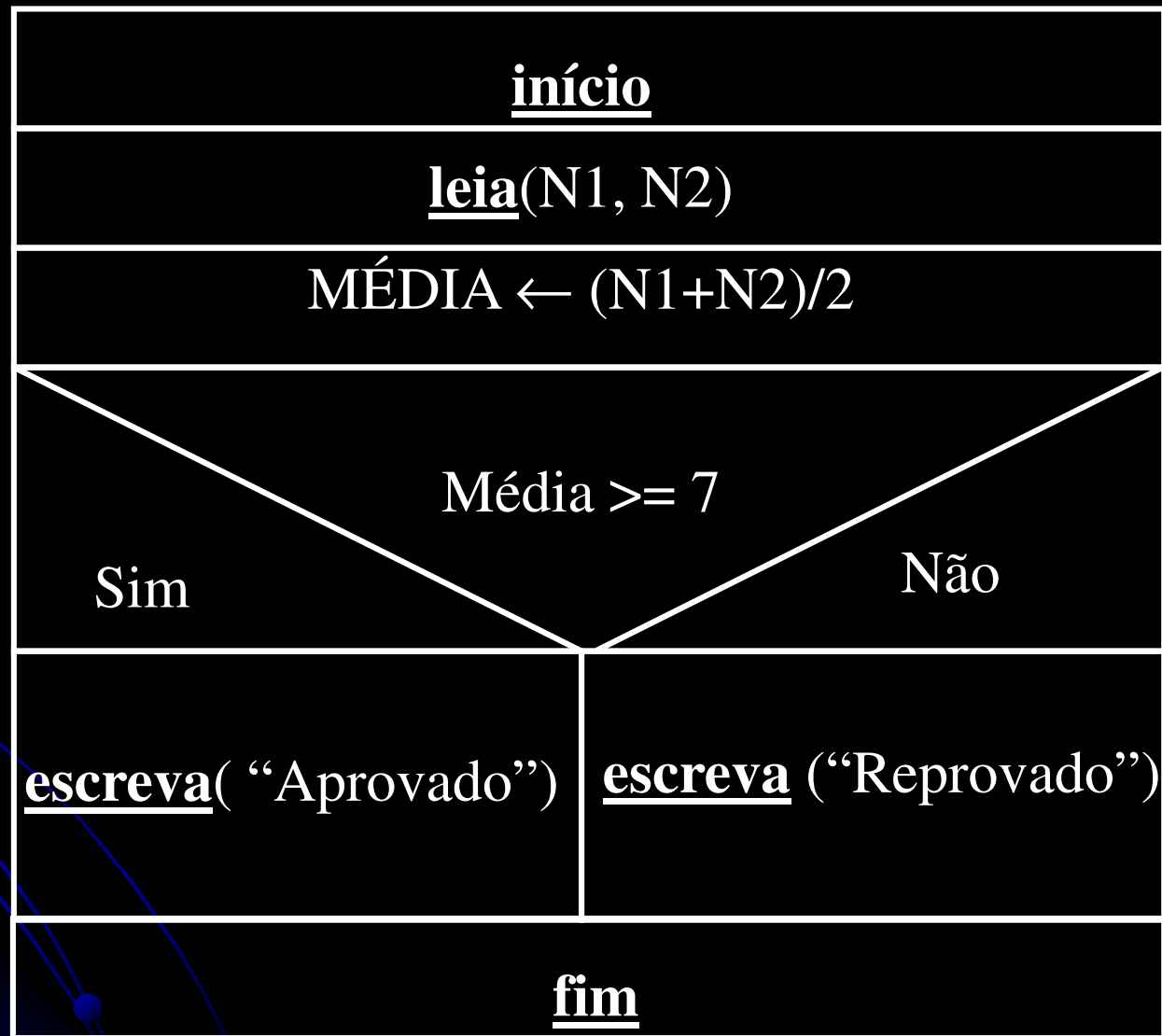
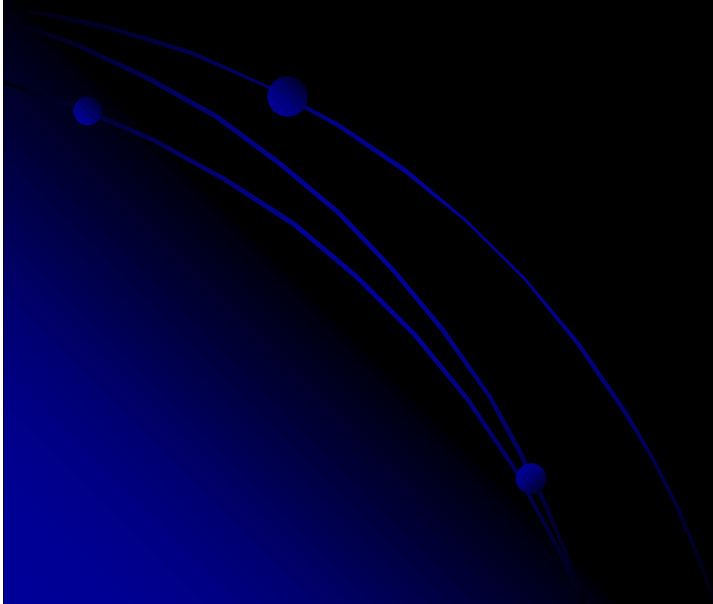


Diagrama de Chapin para o algoritmo do cálculo da média de um aluno

## 2.4. Pseudocódigo

- Conhecido como português estruturado ou portugol
- Rico em detalhes



## 2.4. Representação de um algoritmo na forma de pseudocódigo

```
algoritmo <nome_do_algoritmo>  
  <variáveis globais>  
  <subalgoritmos>  
{Programa Principal}  
  
  início  
    <declaração_de_variáveis>;  
    <corpo_do_algoritmo>;  
  
  fim;  
fim algoritmo.
```

algoritmo SOMA

início

real: NUM1, {Valor do primeiro número}  
NUM2, {Valor do segundo número}  
SOMA; {Valor da soma entre NUM1 e NUM2} \_\_\_\_\_

NUM1 ← 2;

NUM2 ← 4;

SOMA ← NUM1+NUM2;


escreva("O resultado foi:",SOMA);

fim;

fim algoritmo.

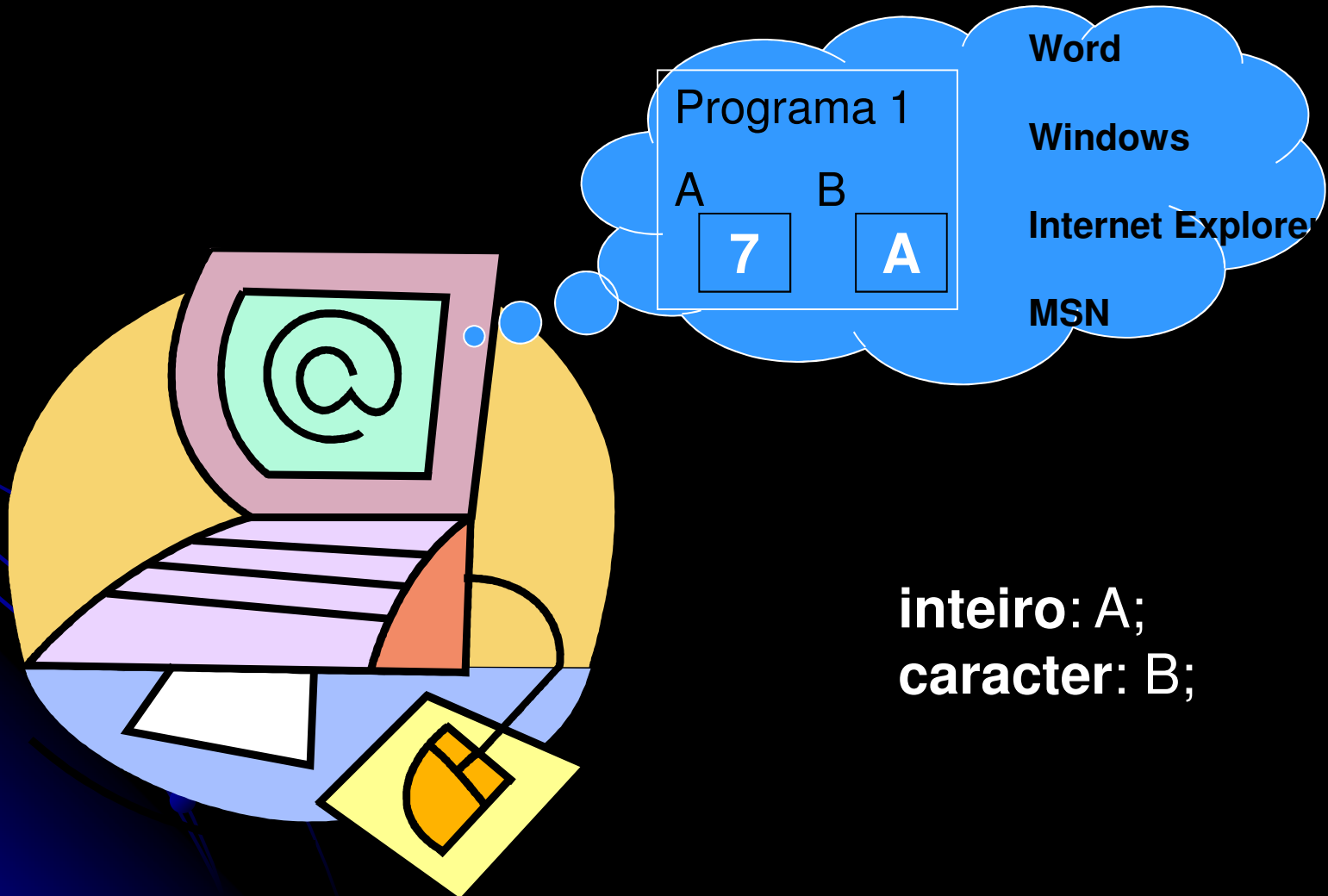
**Pseudocódigo para somar dois números**

# 3. Tipos de Dados

- Inteiros
  - Reais
  - Caracteres
  - Lógicos
- 

# 4. Constantes e Variáveis

## 4.1. Armazenamento de dados na memória



# 4. Constantes e Variáveis

## 4.2. Conceito e utilidades de variáveis:

- variável é tudo aquilo que é sujeito a variações, que é incerto, instável ou inconstante.
  - uma variável possui três atributos:
    - nome,
    - tipo de dado,
    - informação

## 4.3. Definição de Variáveis em Algoritmos

*<tipo\_da\_variável> : <nome\_da\_variável> ;*

**ou**

*<tipo\_das\_variáveis> : <lista\_de\_variáveis> ;*

# 4. Constantes e Variáveis

## 4.4. Conceito e Utilidade de Constantes

- tudo aquilo que é fixo ou estável.

## 4.5. Definição de Constantes em Algoritmos

const <nome\_da\_constante> = <valor>;

