

# Fatec

Indaiatuba

Dr. Archimedes Lammoglia

# Algoritmos e Lógica de Programação



# 1) Apresentação

- ✓ **Engenheiro Eletrônico**

- ✓ **Mestre em Ciências da Computação (IA)**

**Instituto Tecnológico de Toquio, TIT (TOKODAI), Japão** **Figura 1**

- ✓ **Doutor em Engenharia da Produção**

**Telemedicina: um projeto de sistema preventivo para a saúde do trabalhador**

- ✓ <http://lattes.cnpq.br/2227550014469250>

- ✓ [Sergio Gustavo Medina Pereira | LinkedIn](#)

Ementa: Princípios de sistemas computacionais, representação binária, memória e endereçamento, compiladores.

Tipos de dados básicos e representações gráficas dos principais comandos nas linguagens procedurais.

Conceitos básicos sobre algoritmos e métodos para sua construção. Tipos de dados e variáveis. Operadores lógicos.

Estruturas fundamentais de programas: sequencial, condicional e com repetição. Estilo de codificação, indentação, legibilidade, comentários.

Testes de mesa e unitários. Funções. Variáveis compostas homogêneas: vetores e matrizes. Conceitos de controle de versão e gestão de código fonte, Criação de repositórios locais e remotos, Envio (Commit) e resgate de versões, Checkin e Checkout..

Objetivos: Identificar as etapas necessárias para elaboração de um algoritmo e de um programa de computador definindo as diferenças entre eles. Identificar as principais estruturas para construção de algoritmos voltados para a programação de computadores estabelecendo relações entre problemas com estruturas semelhantes.

Construir algoritmos utilizando técnicas de programação estruturada e modular apresentando as características fundamentais da linguagem de programação. Utilizar aplicações ou plataformas para versionamento e controle da programação

# Formas de Avaliação

- Avaliação Formativa – Ao final de cada aula - Peso: 10%
  - Projeto – 20%
  - Avaliação Objetiva:
    - 1º Bimestre
    - 2º Bimestre
- Todos os tópicos tratados - Peso: 70%

As aulas acontecerão nos dias e horários planejados no Sistema Acadêmico (SIGA). Por ser um curso presencial, exige a presença em pelo menos 75% dos 20 encontros (80 aulas) planejadas.

# Conteúdo da Aula

- a) Princípios de sistemas computacionais,
- b) Memória
- c) Espaço de endereçamento
- d) Representação binária,


# Sistema computacional

**Sistemas Computacionais** são conjuntos complexos de hardware e software que trabalham juntos para processar informações e executar tarefas.

Eles abrangem desde dispositivos pessoais, como:

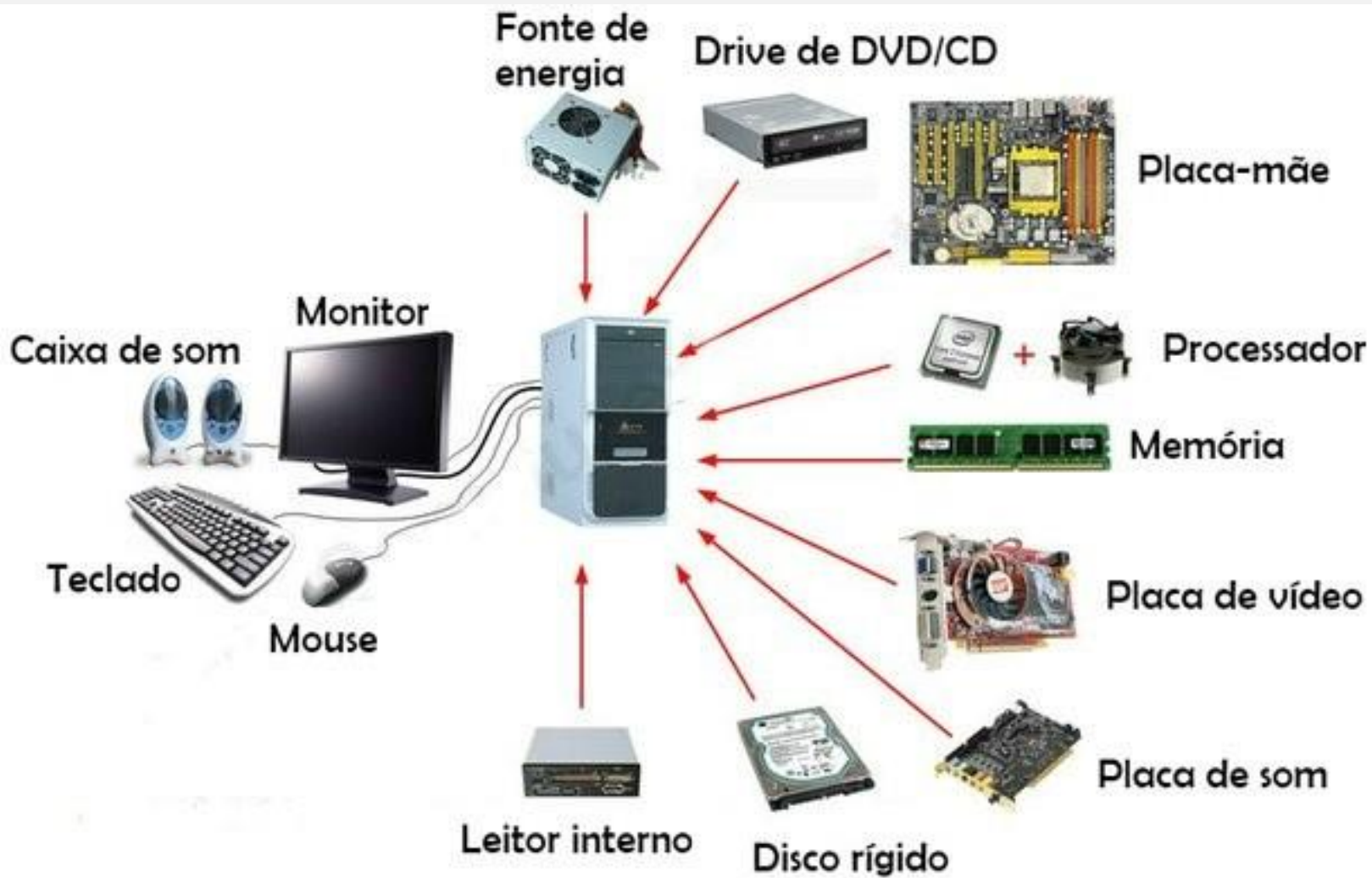
- ✓ Computadores
- ✓ laptop
- ✓ smartphones,
- ✓ Smart TV
- ✓ tablets,
- ✓ impressoras
- ✓ máquinas de lavar
- ✓ geladeiras
- ✓ micro-ondas.
- ✓ Ar-condicionado
- ✓ supercomputadores
- ✓ servidores em data centers.

# Componentes de um Sistema computacional

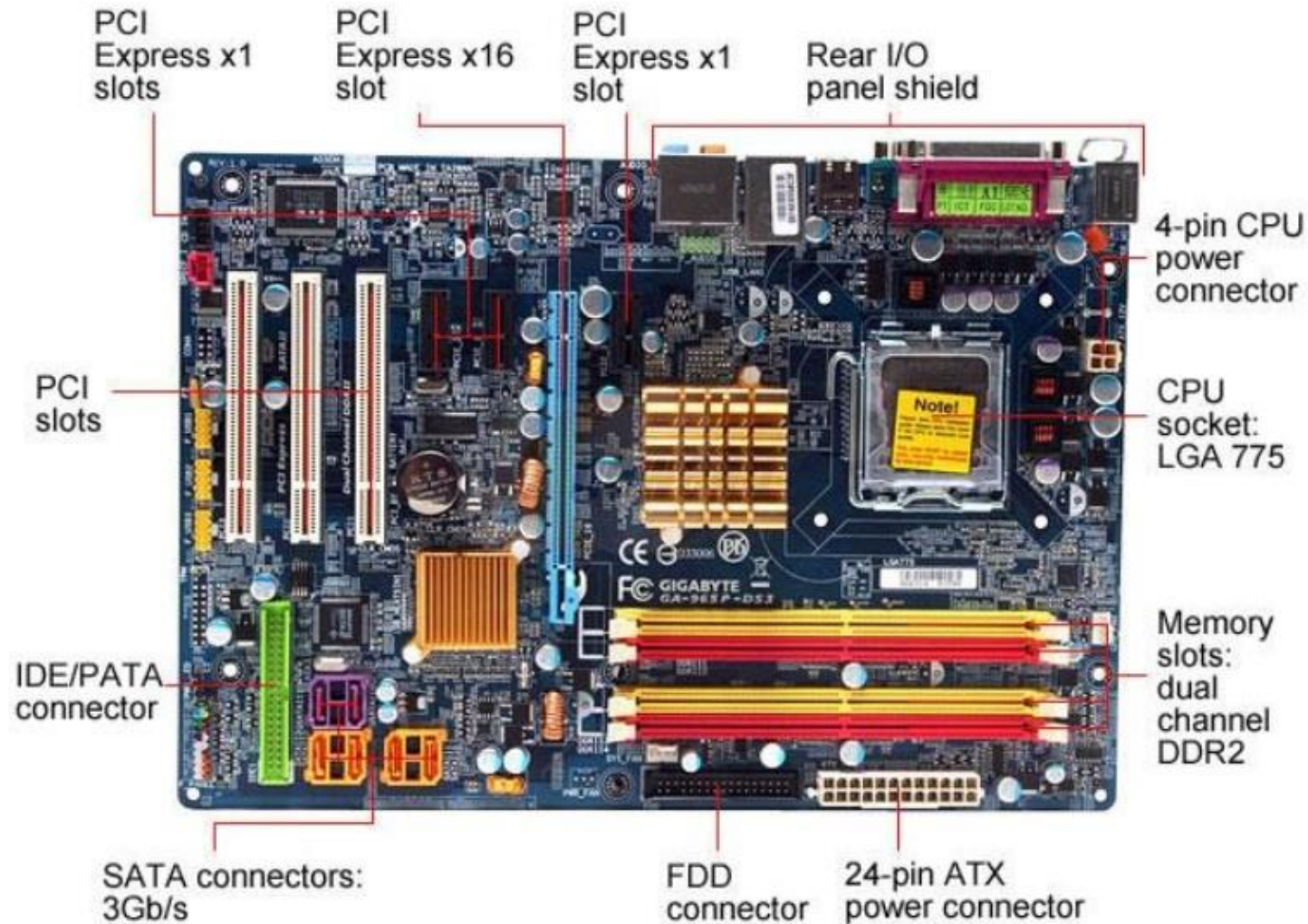
Tipo	Descrição
<b>Hardware</b>	<ul style="list-style-type: none"><li>✓ Refere-se aos componentes físicos, como processadores, memória, dispositivos de entrada/saída e armazenamento.</li><li>✓ Exemplo: CPU (Central Processing Unit),</li><li>✓ RAM (Random Access Memory), disco rígido.</li></ul> 
<b>Software</b>	<ul style="list-style-type: none"><li>✓ Inclui programas, sistemas operacionais e aplicativos.</li><li>✓ Exemplo: Windows, Linux, Microsoft Office.</li></ul>
<b>Firmware</b>	<ul style="list-style-type: none"><li>✓ Software incorporado em dispositivos específicos (por exemplo, BIOS em computadores).</li><li>✓ Exemplo: BIOS (Basic Input/Output System)</li></ul>
<b>Arquitetura de Computadores</b>	<ul style="list-style-type: none"><li>✓ Define a estrutura e organização dos componentes de hardware.</li><li>✓ Exemplo: Arquitetura von Neumann, que separa memória e processamento.</li></ul>



# Hardware



# Mather board









# Componentes de um Sistema computacional

Tipo	Descrição
<b>Organização de Computadores</b>	<ul style="list-style-type: none"><li>✓ Refere-se à maneira como os componentes de hardware são interconectados.</li><li>✓ Exemplo: barramento de dados, barramento de endereços.</li></ul>
<b>Representação de Dados</b>	<ul style="list-style-type: none"><li>✓ Como os computadores armazenam e manipulam informações.</li><li>✓ Exemplo: representação binária, complemento de 2.</li></ul>
<b>Instruções e Programas</b>	<ul style="list-style-type: none"><li>✓ Conjunto de instruções que um processador pode executar.</li><li>✓ Exemplo: ADD (adição), LOAD (carregar).</li></ul>
<b>Compiladores e Linguagens de Programação</b>	<ul style="list-style-type: none"><li>✓ Compiladores traduzem código-fonte em linguagem de alto nível para código de máquina.</li><li>✓ Exemplo: GCC (GNU Compiler Collection), Python.</li></ul>
<b>Exemplos</b>	<ul style="list-style-type: none"><li>✓ Desktops e laptops para uso pessoal.</li><li>✓ Servidores em data centers para hospedar sites e aplicativos.</li><li>✓ Supercomputadores para simulações científicas intensivas.</li></ul>

# Memória

A **memória de um computador** é um dispositivo que armazena dados e instruções digitalmente.

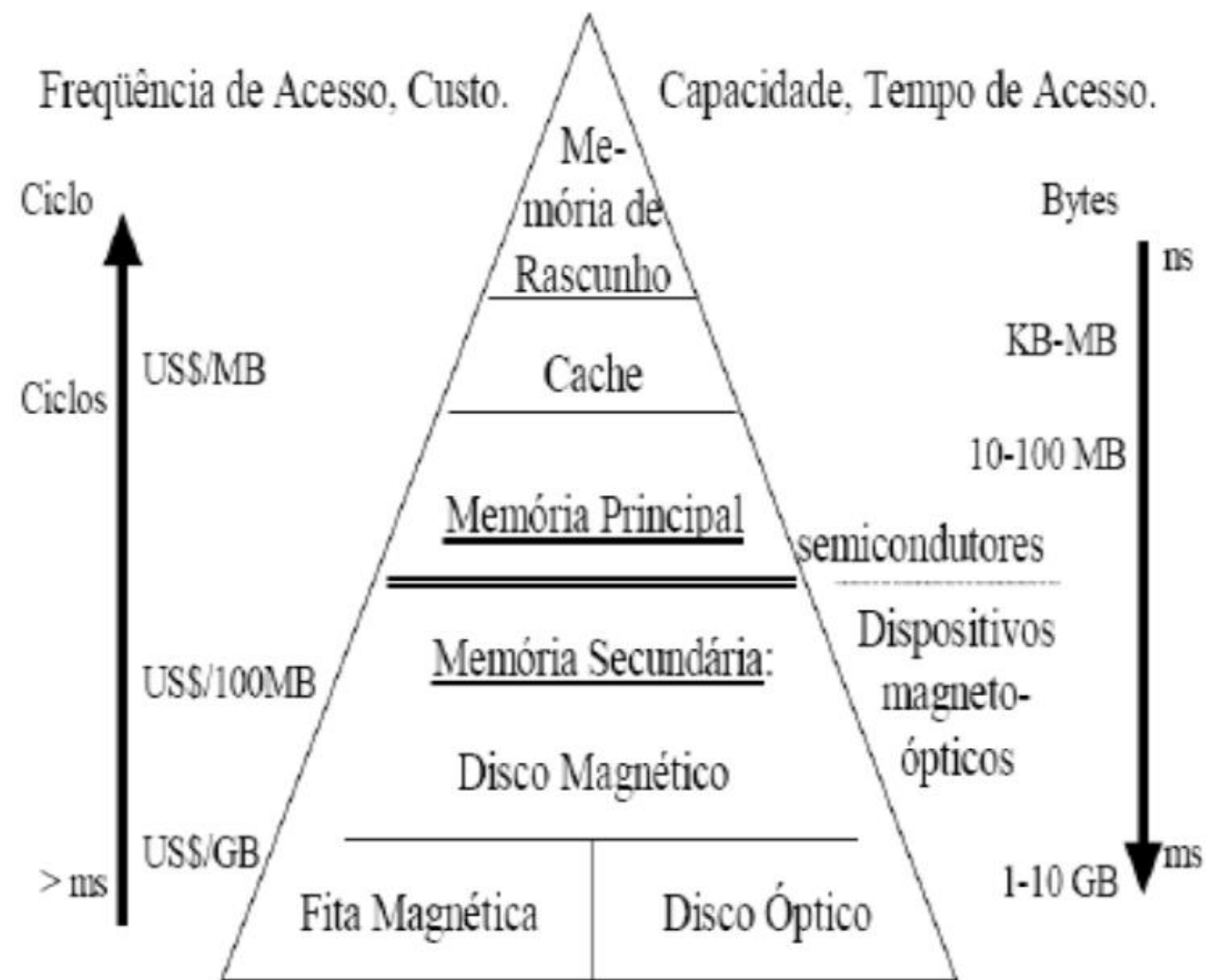
Ela pode ser temporária (volátil) ou permanente (não volátil).

Tipos de Memória	Descrição
<b>Memória RAM</b> 	<ul style="list-style-type: none"><li>✓ Random Access Memory:</li><li>✓ Volátil e de acesso rápido.</li><li>✓ Armazena dados temporariamente enquanto o computador está ligado.</li><li>✓ Exemplo: Quando você abre um aplicativo, os dados são carregados na RAM.</li></ul>
<b>Memória ROM</b> 	<ul style="list-style-type: none"><li>✓ Read-Only Memory</li><li>✓ Não volátil e de acesso somente leitura.</li><li>✓ Contém instruções permanentes, como o código de inicialização do computador.</li><li>✓ Exemplo: BIOS (Basic Input/Output System) em um PC</li></ul>

# Memória

Tipos de Memória	Descrição
<b>Memória Cache</b>	<ul style="list-style-type: none"><li>✓ Volátil e de alta velocidade.</li><li>✓ Armazena dados frequentemente acessados para acelerar o processamento.</li><li>✓ Exemplo: Cache L1, L2 e L3 em processadores</li></ul>
<b>Memória Virtual</b>	<ul style="list-style-type: none"><li>✓ Combina RAM e espaço em disco para expandir o espaço de endereçamento.</li><li>✓ Permite que programas usem mais memória do que a RAM física disponível.</li><li>✓ Exemplo: Arquivos de paginação no disco rígido.</li></ul>

# Relação entre Dispositivos de Armazenamento



# Características de comuns (Memória)

<b>Unidade de Armazenamento</b>	<ul style="list-style-type: none"><li>✓ O bit é a menor unidade de armazenamento.</li><li>✓ Bits formam bytes, kilobytes(KB), megabytes(MB) e gigabytes(GB).</li></ul>
<b>Capacidade de Armazenamento</b>	<ul style="list-style-type: none"><li>✓ Medida em KB, MB ou GB.</li><li>✓ Exemplo: Um disco rígido de 1 TB tem capacidade para 1 terabyte (1.000 GB) de dados.</li></ul>
<b>Tempo de Acesso</b>	<ul style="list-style-type: none"><li>✓ O tempo desde o endereçamento até a leitura/escrita.</li><li>✓ Memórias mais rápidas têm menor tempo de acesso.</li></ul>
<b>Tipo de Acesso</b>	<ul style="list-style-type: none"><li>✓ Aleatório: Acesso direto a qualquer posição de memória (RAM).</li><li>✓ Serial: Acesso sequencial (disco rígido).</li></ul>
<b>Estabilidade</b>	<ul style="list-style-type: none"><li>✓ Volatilidade (RAM) vs. não volatilidade (ROM).</li></ul>
<b>Funcionalidade</b>	<ul style="list-style-type: none"><li>✓ Armazenamento temporário (RAM) vs. armazenamento permanente (ROM).</li></ul>



# Características de comuns (Memória)

## Exemplo Didático:

- ✓ Imagine a RAM como uma mesa de trabalho temporária onde você coloca os livros (dados) que está usando no momento.
- ✓ A ROM é como uma estante de livros com instruções permanentes, como um manual de usuário.

Em resumo, a memória de um computador é essencial para o funcionamento do sistema, permitindo o armazenamento e acesso eficiente de dados e programas

## Definição:

- O espaço de endereçamento é uma faixa de endereços discretos que correspondem a locais específicos na memória.
- Cada endereço pode apontar para um registrador físico, uma célula de memória, um dispositivo periférico ou outra entidade lógica.

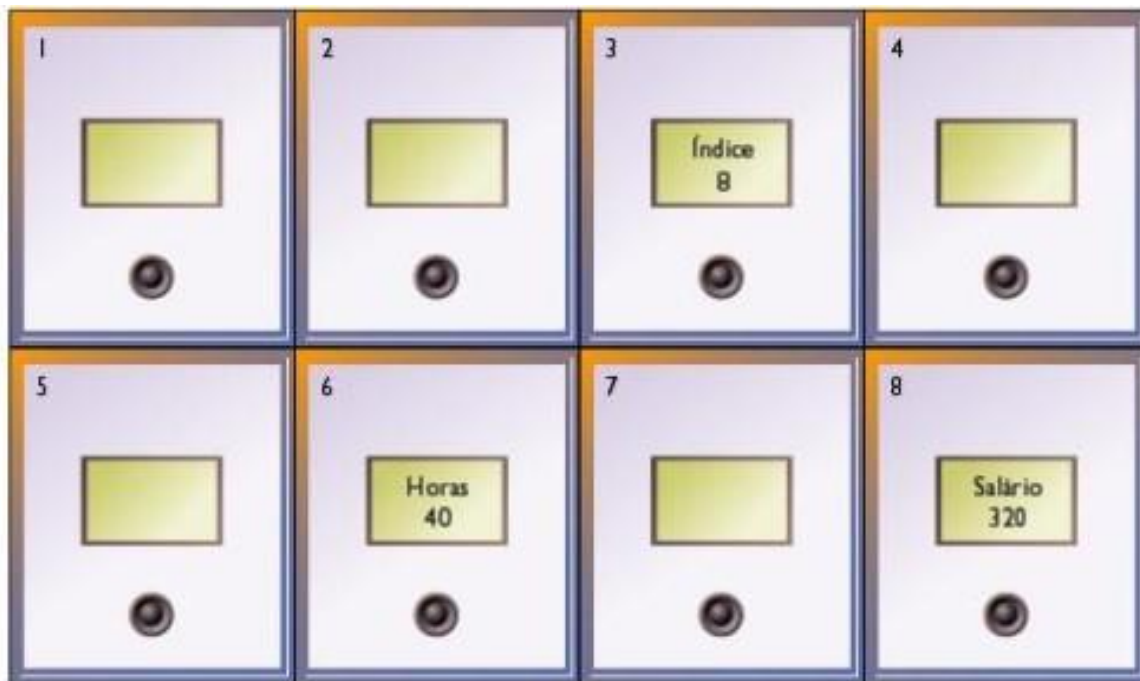
## Analogia Residencial:

- Imagine que o espaço de endereçamento é como um bairro com várias casas.
- Cada endereço é como o número de uma casa, indicando onde os dados estão armazenados.

## Exemplos de Espaços de Endereçamento:

- ✓ **Endereço IP:** Identifica um dispositivo em uma rede.
- ✓ **Memória Principal (RAM):** Cada endereço aponta para uma célula de memória.
- ✓ **Memória Virtual:** Expande o espaço de endereçamento além da memória física.
- ✓ **Espaço de Portas de E/S:** Endereça dispositivos de entrada/saída.
- ✓ **Esquema CHS dos HDs:** Define setores em discos rígidos.

# Tipos de endereçamento de memória



**Endereço**

A013545D

A013545E

A013545F

A0135460

A0135461

A0135462

A0135463

**Conteúdo**

0	1	0	0	1	1	0	1
0	1	1	0	1	0	1	1
0	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0
0	1	0	1	1	1	0	1
1	0	1	1	1	0	1	1
1	0	1	0	0	1	0	1

**Locação**

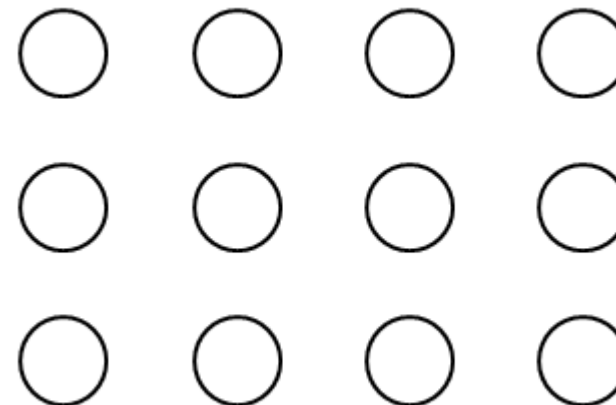
## Exemplos Específicos no Núcleo Linux:

- ✓ **Espaço de Endereçamento Virtual do Núcleo:** Acessado pelo próprio núcleo.
- ✓ **Espaço de Endereçamento Virtual do Usuário:** Acessado pelo núcleo para copiar dados entre usuário e núcleo.
- ✓ **Memória E/S:** Acessada por funções como `readb()`, `writel()`, `memcpy_toio()`.

Em resumo, o espaço de endereçamento é fundamental para a comunicação entre hardware, software e dispositivos, permitindo que o computador acesse e manipule dados de forma eficiente.

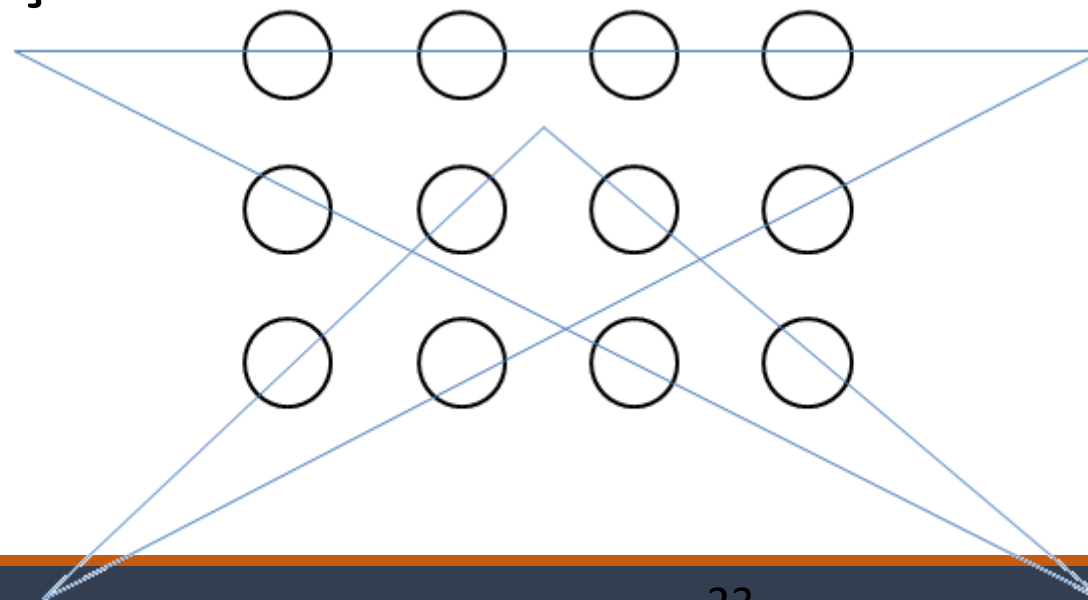
# Dinâmica 1 – Lógica (Pensar fora da caixa)

1. Passar a caneta em todas as bolinhas
2. Usar 5 linhas retas
3. Não pode tirar a caneta do papel \*
4. Terminar onde começou



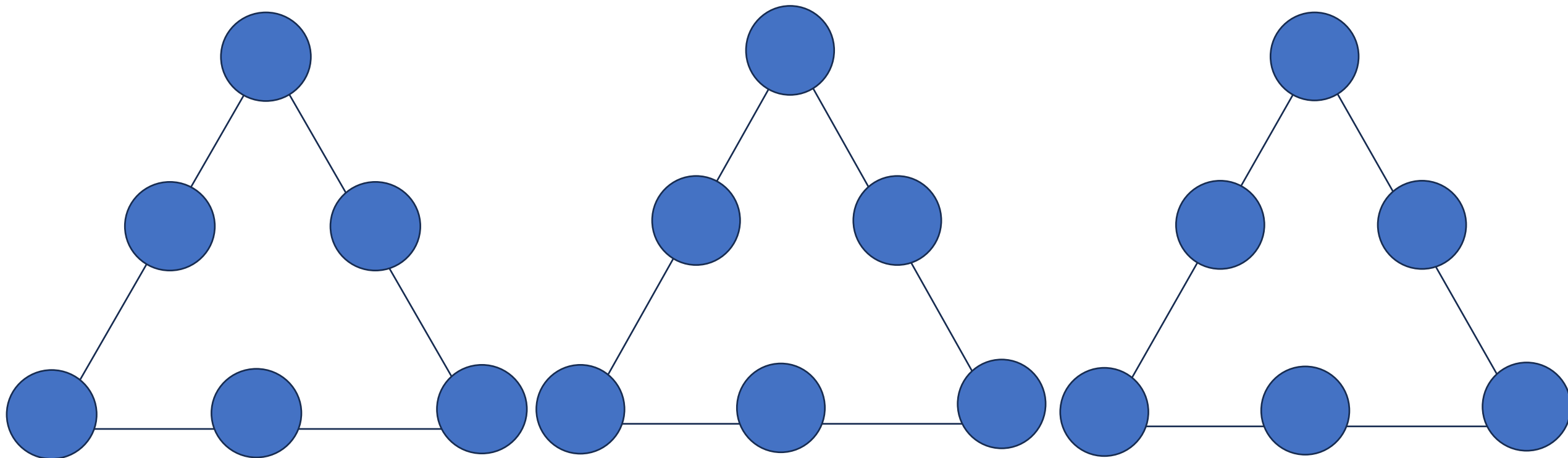
# Solução: Dinâmica 1 - Lógica

1. Passar a caneta em todas as bolinhas
2. Usar 5 linhas retas
3. Não pode tirar a caneta do papel \*
4. Terminar onde começou



## Desafio do triângulo

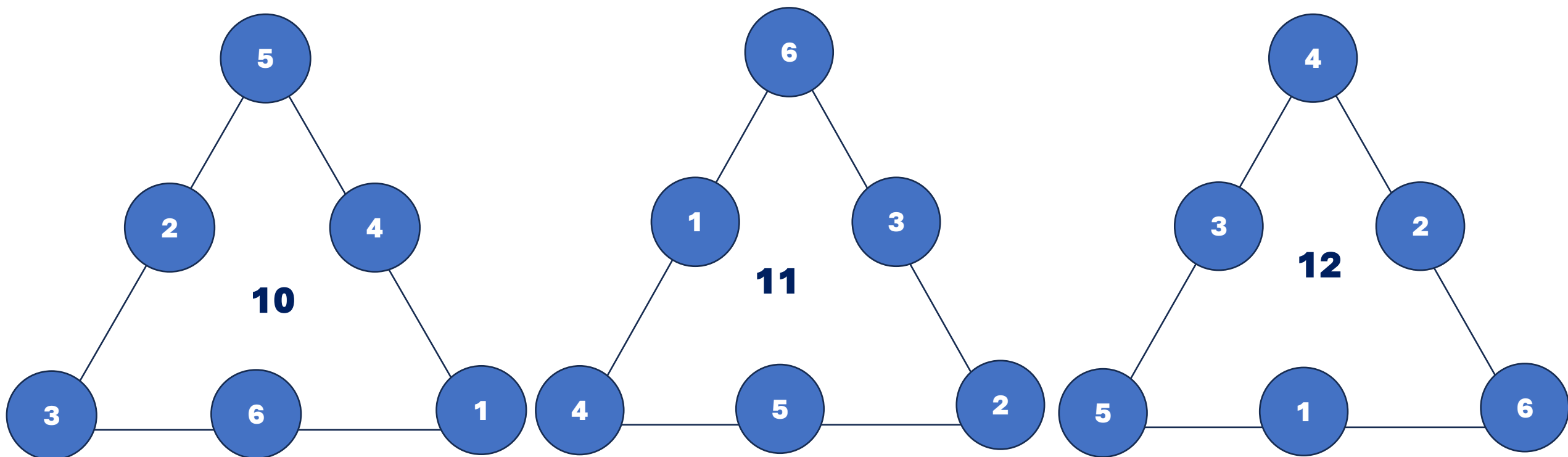
1. Preencher os círculos com números de 1 a 6
2. A soma de cada lado deve ser 10, 11 e 12





## Desafio do triângulo

1. Preencher os círculos com números de 1 a 6
2. A soma de cada lado deve ser 10, 11 e 12



## Puzzle - Travessia de Rio

Um fazendeiro quer levar de uma margem de um rio para a outra uma raposa uma galinha e um saco de milho.

No barco só pode levar um de cada vez. como deve fazer para que a raposa não coma a galinha nem esta coma o milho, que é o que acontece.

Se por exemplo, ele tentar levar 1º o milho ( fica a raposa e a galinha, logo a raposa come a galinha.

Mas também não podem ficar juntas na outra margem. enquanto ele vem buscar o milho, como é evidente)



## Puzzle - Travessia de Rio

### Algoritmo

- 1) O fazendeiro leva(barco) 1ro a galinha para a margem direita(galinha fica)
- 2) O fazendeiro retorna sozinho(barco) para a margem esquerda
- 3) O fazendeiro leva(barco) a raposa para a margem direita(raposa fica)
- 4) O fazendeiro retorna (barco) com a galinha para a margem esquerda
- 5) O fazendeiro deixa a galinha e leva(barco) o milho para a margem direita(a raposa fica com o milho)
- 6) O fazendeiro retorna sozinho (barco) com a galinha para a margem esquerda (raposa, milho e a galinha)



# ALGORITMO



➤ O que é?

Sequência ordenada de passos que devem ser seguidos para realização de uma tarefa. (BERG e FIGUEIRÓ, 1998)

Norte para solucionar problemas computacionais.

## ➤ O que é?

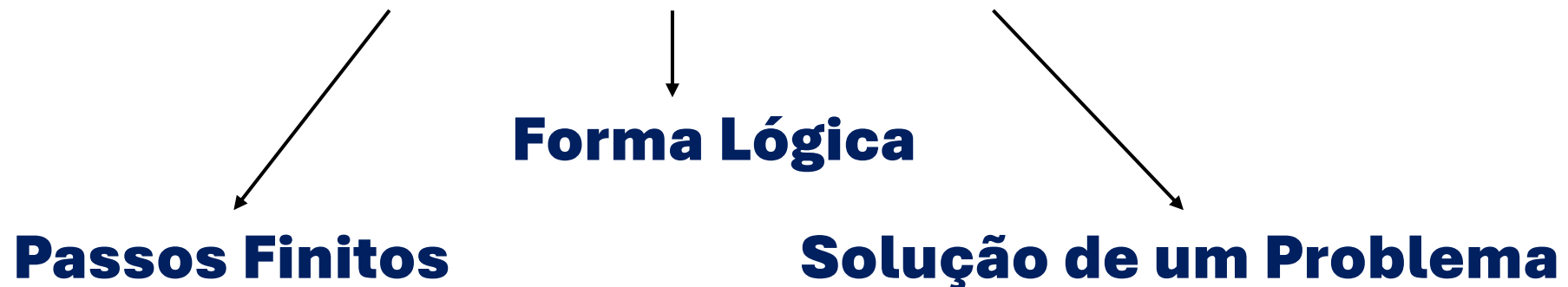
"Algoritmo é a descrição de uma seqüência de passos que deve ser seguida para a realização de uma tarefa." (ASCÊNCIO, 2003)

"Algoritmo é uma seqüência de passos que visam atingir um objetivo bem definido." (FORBELLONE, 2005)

"Algoritmo são regras formais para a obtenção de um resultado ou da solução de um problema, englobando fórmulas de expressões aritméticas." (MANZANO, 2000)

➤ O que é?

Algoritmo é uma sequência de passos **finitos** de forma lógica para solucionar um determinado problema/tarefa.



## ➤ Exemplos

- ✓ Chupar uma bala.
- ✓ Trocar uma lâmpada.
- ✓ Atravessar a rua.
- ✓ Preparar um bolo.

Situações do cotidiano  
que requer passos  
lógicos para serem  
realizadas.

Rotina.



## ➤ Exemplos

### ✓ Chupar uma bala.

- Os passos um após o outro;
- Ser suficientes;
- Podem existir mais de um algoritmo para o mesmo resultado.

1. Pegar a bala
2. Retirar o papel
3. Jogar o papel no lixo
4. Colocar a bala na boca
5. Chupar a bala

## ➤ Exemplos

Algoritmo: Chupar Bala

1. Pegar a bala
2. Retirar o papel
3. Jogar o papel no lixo
4. Colocar a bala na boca
5. Chupar a bala

- ✓ Passos Finitos
- ✓ Forma lógica
- ✓ Solução do problema

## ➤ Exemplos

### **Sacar dinheiro no banco 24 horas**

Passo 1 - Ir até o banco 24 horas.

Passo 2 - Colocar o cartão.

Passo 3 - Digitar a senha.

Passo 4 - Solicitar a quantia desejada.

Passo 5 - Se o saldo for maior ou igual à quantia desejada, sacar; caso contrário, mostrar mensagem de impossibilidade de saque.

Passo 6 - Retirar o cartão.

Passo 7 - Sair do banco 24 horas.

## ➤ Exemplos

### **Trocar uma lâmpada**

Passo 1 - Pegar uma escada.

Passo 2 - Posicionar a escada embaixo da lâmpada.

Passo 3 - Buscar uma lâmpada nova.

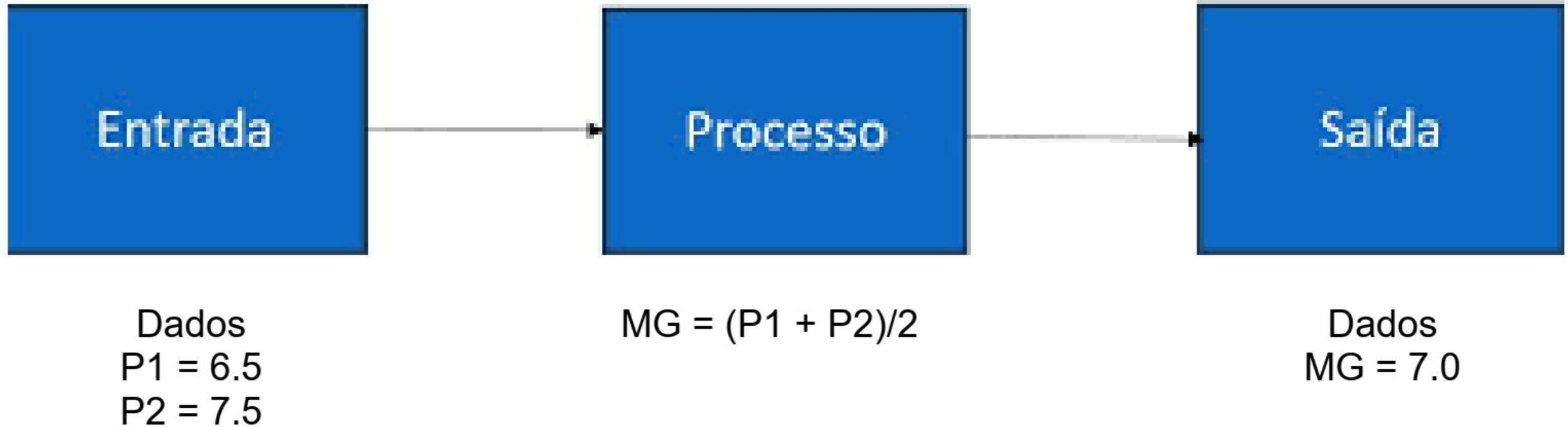
Passo 4 - Subir na escada.

Passo 5 - Retirar a lâmpada velha.

Passo 6 - Colocar a lâmpada nova.

➤ Responda a essas questões:

- ✓ Qual o problema/tarefa a ser resolvido?
- ✓ Qual os passos necessários?
- ✓ Esses passos seguem uma forma lógica?
- ✓ Seguindo esses passos conseguimos a solução?



## ➤ Métodos para a Construção de Algoritmos

- Ler atentamente o enunciado, destacando os pontos mais importantes.
- Definir os dados de entrada, ou seja, quais dados serão fornecidos.
- Definir o processamento, ou seja, quais cálculos serão efetuados e quais as restrições para esses cálculos. O processamento é responsável pela transformação dos dados de entrada em dados de saída.
- Definir os dados de saída, ou seja, quais dados serão gerados depois do processamento.
- Construir o algoritmo.
- Testar o algoritmo realizando simulações.

"Lógica é a arte de bem pensar; é a ciência das formas do pensamento."  
(FORBELLONE, 2005)

Estão associados à Lógica a coerência e a racionalidade, a correção do pensamento e a ordem do pensamento.



- 1) Todo mamífero é um animal.  
Todo cavalo é um mamífero.  
Portanto, todo cavalo é um animal.
  
  - 2) A gaveta está fechada.  
A caneta está dentro da gaveta.  
Precisamos primeiro abrir a gaveta para depois pegar a caneta.
- A lógica é fundamental na elaboração de algoritmos.

Ação e o efeito de programar.

Programação em Informática.

- ✓ Lógica de programação.
- ✓ Linguagem de programação

**Programar é configurar o computador para que ele siga os passos de um algoritmo!**

## Técnicas de Programação

É a técnica de desenvolver **algoritmos** (sequências lógicas) para atingir determinados objetivos dentro de certas regras baseadas na Lógica

- A Lógica de Programação pode ser representada com a utilização de linguagens de programação.
- Os algoritmos são utilizados para representar mais fielmente o raciocínio da Lógica de Programação.

# Definição de Programa

"Algoritmo codificado em uma linguagem de programação."

- O algoritmo é algo abstrato escrito em linguagem que o computador não entende.
- Daí, a necessidade de codificá-lo, isto é, escrevê-lo em uma linguagem que o computador entenda.

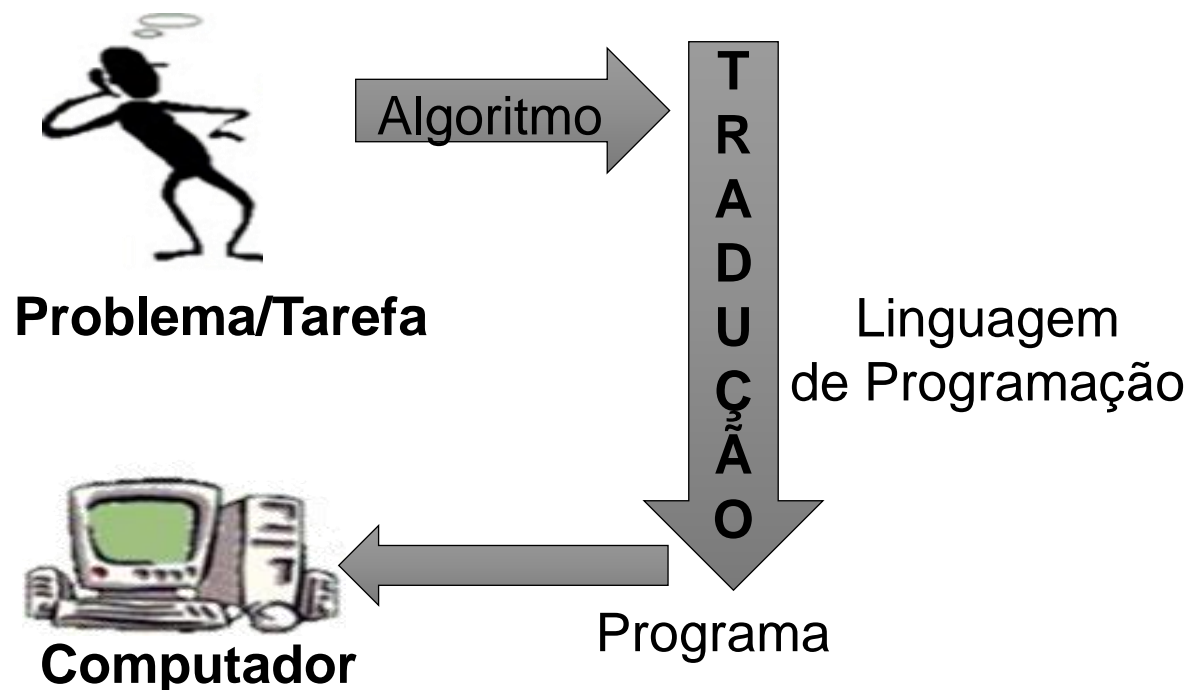
Existem várias linguagens de programação:

- ✓ C/C++
- ✓ PHP
- ✓ JAVA
- ✓ C#
- ✓ PYTHON

Algoritmo  $\neq$  Programa de computador

- ✓ São os passos necessários para se realizar uma tarefa.
- ✓ Podem ser compilados em qualquer linguagem de programação.

**O algoritmo não é a solução do problema, mas o caminho que leva a solução!**



Algoritmos não se aprendem:

- Copiando algoritmos
- Estudando algoritmos

Algoritmos só se aprendem:

- Construindo algoritmos
- Testando algoritmos

- a) Linguagem natural
- b) Diagrama de Blocos – (Fluxograma)
- c) Pseudocódigo



- ✓ Representados diretamente na linguagem natural, como por exemplo o algoritmo para chupar uma bala:

### Algoritmo: Chupar Bala

1. Pegar a bala
2. Retirar o papel
3. Jogar o papel no lixo
4. Colocar a bala na boca
5. Chupar a bala

- **Tipos de Representação**

- **Descrição Narrativa**

- Utiliza uma linguagem natural e descreve os passos a serem seguidos para a resolução do problema.

- **Vantagem**: Não é necessário aprender nenhum conceito novo.

- **Desvantagem**: Problemas de interpretação e dificuldade de transcrição para código.










- **Tipos de Representação (cont.)**

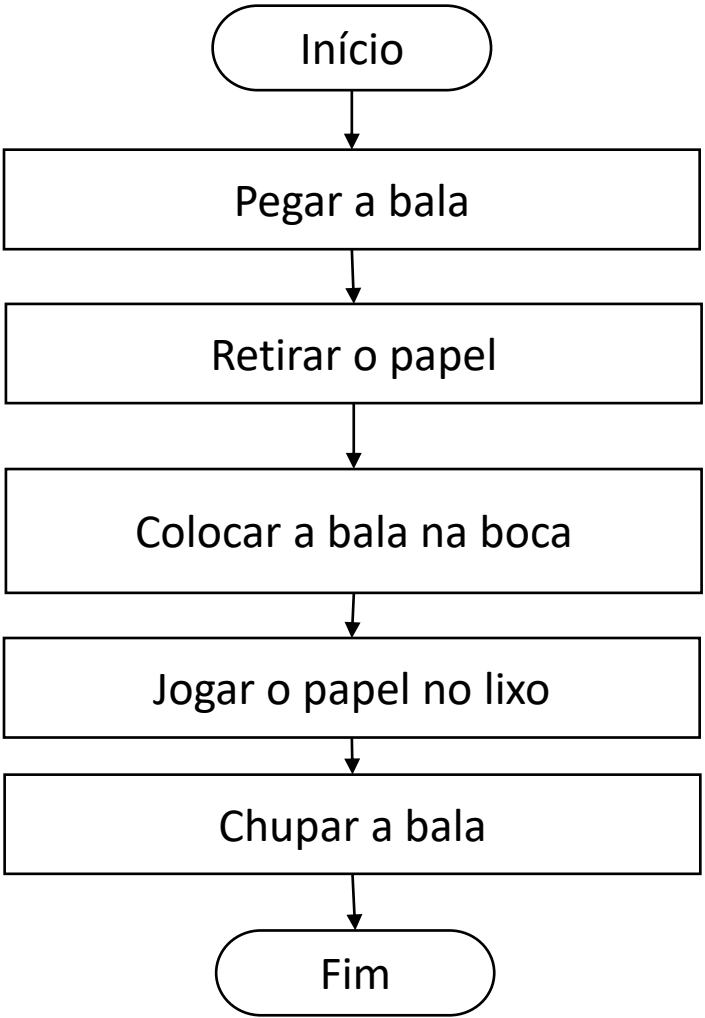
- **Fluxograma**

- Utiliza símbolos gráficos pré-definidos para descrever os passos a serem seguidos para a resolução do problema.
    - **Vantagem:** Fácil entendimento e visualização.
    - **Desvantagem:** Necessidade de aprender nova simbologia e dificuldade de transcrição para código.

## b) Fluxograma

✓ Representação gráfica:

Símbolo	Significado	Descrição
	Terminal	Representa o início ou o fim de um fluxo lógico. Em alguns casos definem as sub-rotinas.
	Entrada manual	Determina a entrada manual dos dados, geralmente através de um teclado.
	Processamento	Representa a execução de ações de processamento.
	Exibição	Mostra o resultado de uma ação, geralmente através da tela de um computador.
	Decisão	Representa os desvios condicionais nas operações de tomada de decisão e laços condicionais para repetição de alguns trechos do programa.
	Preparação	Representa a execução de um laço incondicional que permite a modificação de instruções do laço.
	Processo Predefinido	Define um grupo de operações relacionadas a uma sub-rotina.
	Conector	Representa pontos de conexões entre trechos de programas, que podem ser apontados para outras partes do diagrama de bloco.
	Linha	Representa os vínculos existentes entre os símbolos de um diagrama de blocos.





**TERMINAL** = Início e final do fluxograma



**ENTRADA MANUAL** = Entrada de Dados



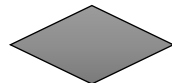
**PROCESSAMENTO** = Operações, cálculos e atribuição de valores



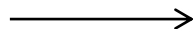
**RELATÓRIO** = Saída de dados



**SAÍDA/EXIBIÇÃO** = Saída de dados em vídeo



**DECISAO** = Decisão ou desvio



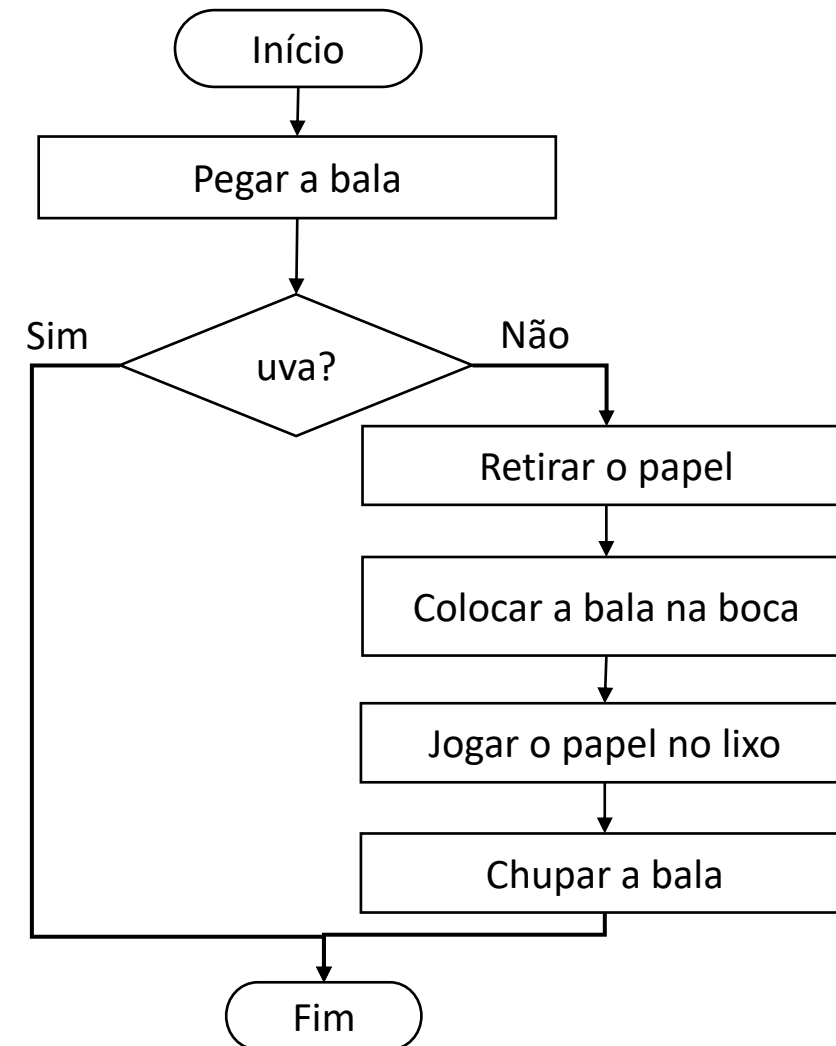
**FLUXO** = Fluxo de Dados



**CONECTOR/DESVIO** = Conector

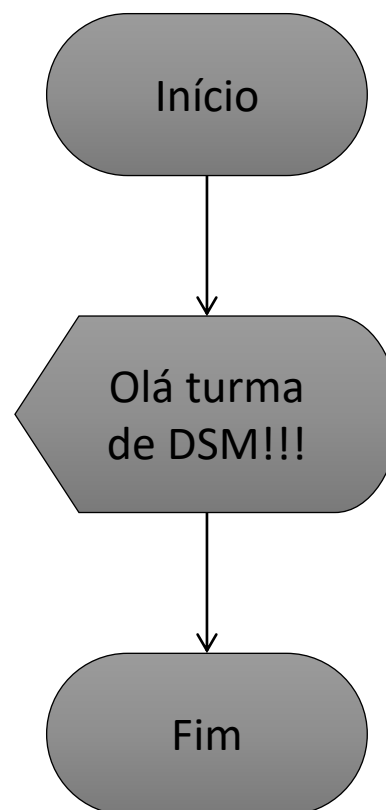
✓ Condições podem existir:

Considere que você não gosta de bala de uva.



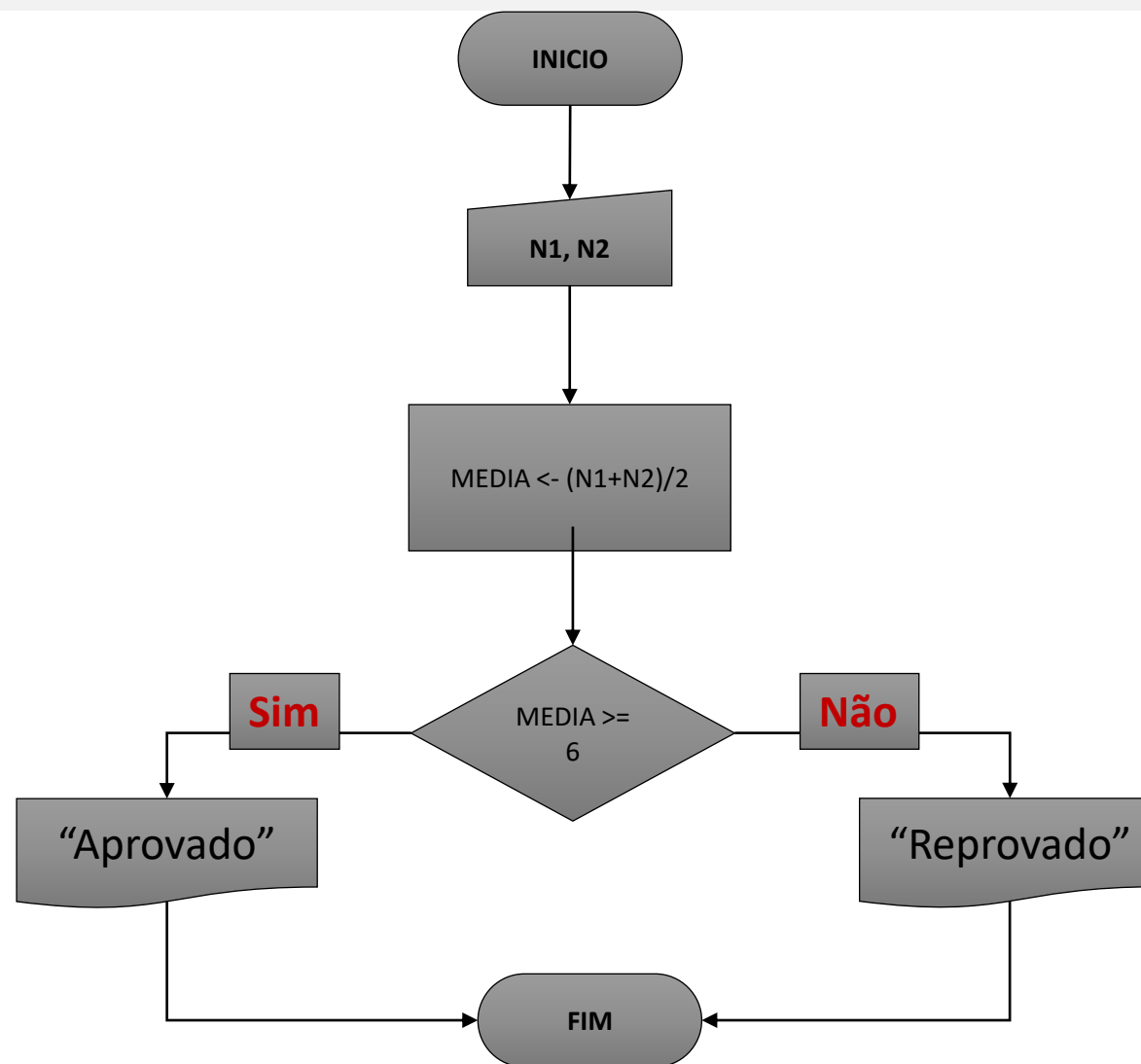
## Fluxograma - Exemplo 01

- Imprime a frase "Olá turma de DSM!" na tela do computador.



# Fluxograma - Exemplo 01

- Cálculo da média



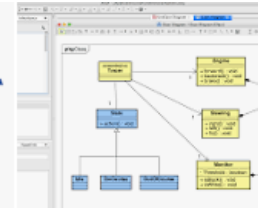




**Visual Paradigm**  
Online



**astah**



Draw.io

# Fluxograma - Ferramentas



**Dia Diagram Editor** para Window

✓ Grátis ✓ Em Português V 0.97.2

3.5 ★★★★★ (491) ✓ Status de segurança

- **Tipos de Representação (cont.)**
  - **Portugol**
    - Utiliza regras pré-definidas e descreve os passos a serem seguidos para a resolução do problema.
    - **Vantagem**: Fácil transcrição para qualquer linguagem de programação e pouca necessidade de novo aprendizado.
    - **Desvantagem**: Ainda que pouco, é necessário algum aprendizado.

- **Portugol - Sintaxe Básica**

- Bloco Principal

algoritmo "semnome"  
*// Seção de Declarações*  
var

inicio  
*// Seção de Comandos*  
fimalgoritmo

✓ Mais próximo da programação real.

```
Algoritmo <nome_do_algoritmo>;  
<declaração_de_variáveis>;  
Início  
<corpo do algoritmo>  
Fim
```

- **Portugol - Sintaxe Básica (cont.)**

- Declaração de Variáveis

- var

- X : numerico

- Y, Z : literal

- Comandos de Atribuição

- X  $\leftarrow$  4

- X  $\leftarrow$  X + 2

- Y  $\leftarrow$  "aula"

- **Portugol - Sintaxe Básica (cont.)**
  - Comandos de Entrada  
leia(X)  
leia(Y)
  - Comandos de Saída  
escreva(X)  
escreva(Y)
- **Ferramenta: VisuAlg**  
(<https://visualg3.com.br/baixar-o-visualg3-0/>)

Tarefa: Calcular a média de um aluno por meio de duas notas.

### Linguagem natural

Algoritmo: Calcular Média

1. Pegar a primeira nota
2. Pegar a segunda nota
3. Somar a primeira nota com a segunda nota.
4. Dividir o resultado por dois

**Entrada:** Nota 1 e Nota 2

**Processamento:** Média =  $(\text{Nota 1} + \text{Nota 2}) / 2$

**Saída:** Média

Os valores que não conhecemos (e que representamos pelo nome Nota1, Nota2 e Media) são denominados **variáveis**.

## Pseudocódigo

### Algoritmo Calcular\_Media

**Var** Nota1, Nota2, Media: **real**;

### Início

**Leia**(Nota1, Nota2);

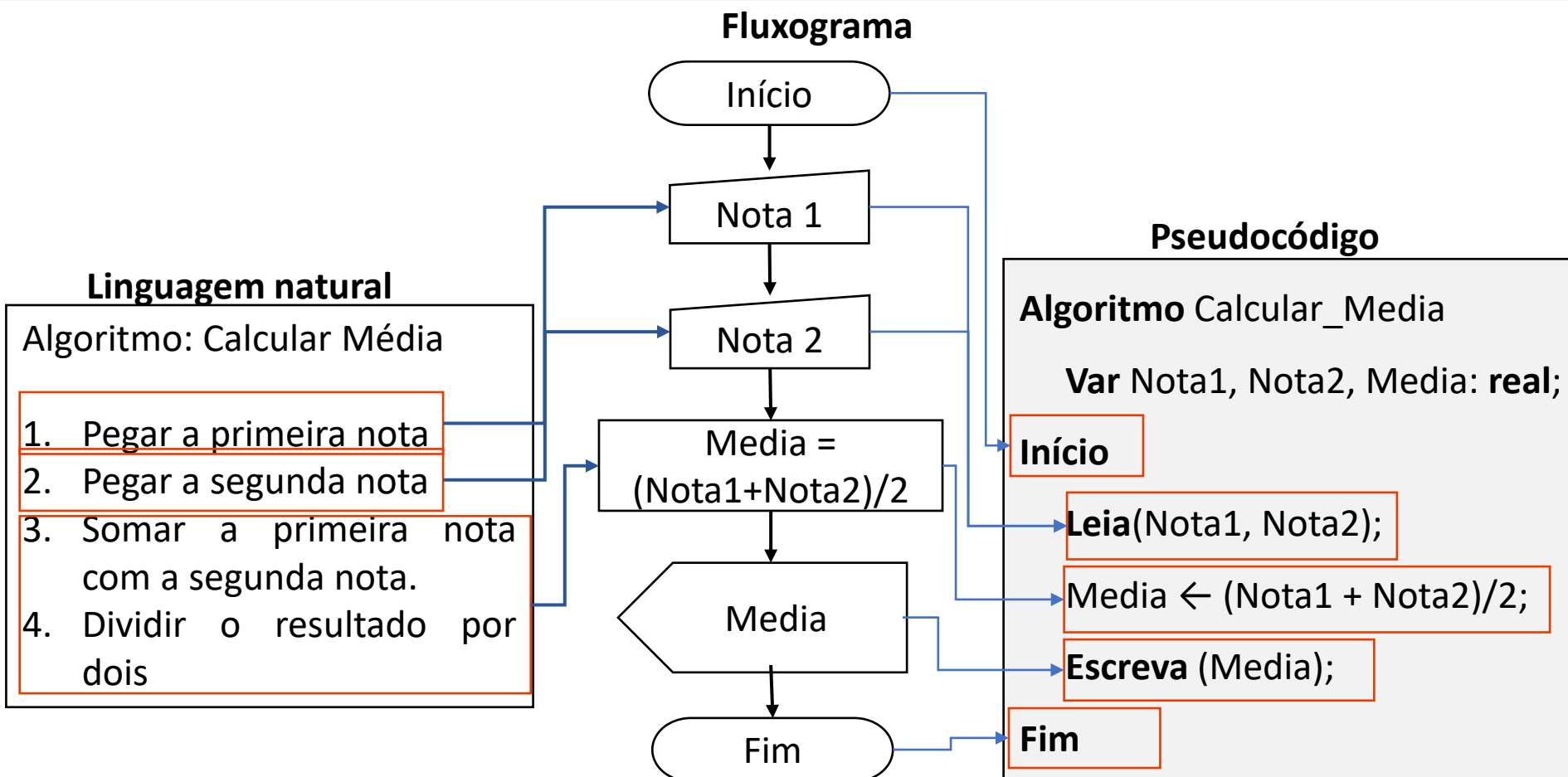
Media  $\leftarrow$  (Nota1 + Nota2)/2;

**Escreva** (Media);

### Fim

```
1 Algoritmo "mediaSimples"
2
3 Var
4 // Seção de Declarações das variáveis
5 valor1, valor2, media:real
6
7 Inicio
8 // Seção de Comandos, procedimento, funções, operadores, etc...
9
10 escreva("Digite o valor 1: ")
11 leia(valor1)
12 escreva("Digite o valor 2:")
13 leia(valor2)
14 media<-(valor1+valor2)/2
15 escreva("A media dos valores é:",media)
16
17 Fimalgoritmo
```





01 – Utilizando os três tipos de representação de algoritmos (linguagem natural, fluxograma e pseudocódigo), faça:

- a) Um algoritmo que receba a idade de uma pessoa e verifique se ela é “Maior” ou “Menor”. Caso seja maior retorne mensagem informado “Acesso permitido”, caso contrário, “Acesso negado”.
- b) Um algoritmo que receba 3 número e calcule a média aritmética desses números.

## Referências Bibliográficas

- ASCENCIO, A. F. G., CAMPOS, E. A. V. Fundamentos da Programação de Computadores: algoritmos, Pascal e C/C++ e Java. 3ª ed. São Paulo: Prentice Hall, 2012.
- FORBELLONE, L. V., EBERSPACHER, H. F. **Lógica de Programação: a construção de algoritmos e estruturas de dados.** 3ª ed.- São Paulo: Prentice Hall, 2005.
- MANZANO, J. A. N. G, OLIVEIRA, J. F. **Algoritmos: lógica para desenvolvimento de programação de computadores.** São Paulo: Érica, 2000.