

Bacharelado em Ciência e Tecnologia



Universidade Federal do ABC

BCM0505

Processamento da Informação

Introdução a programação de computadores
Introduction to computer programming

Prof. Rogério Neves

Rogério.neves@ufabc.edu.br

1º Quadrimestre de 2016

Hardware Versus Software

Hardware

Physical part
Support for input/output
and processing of data



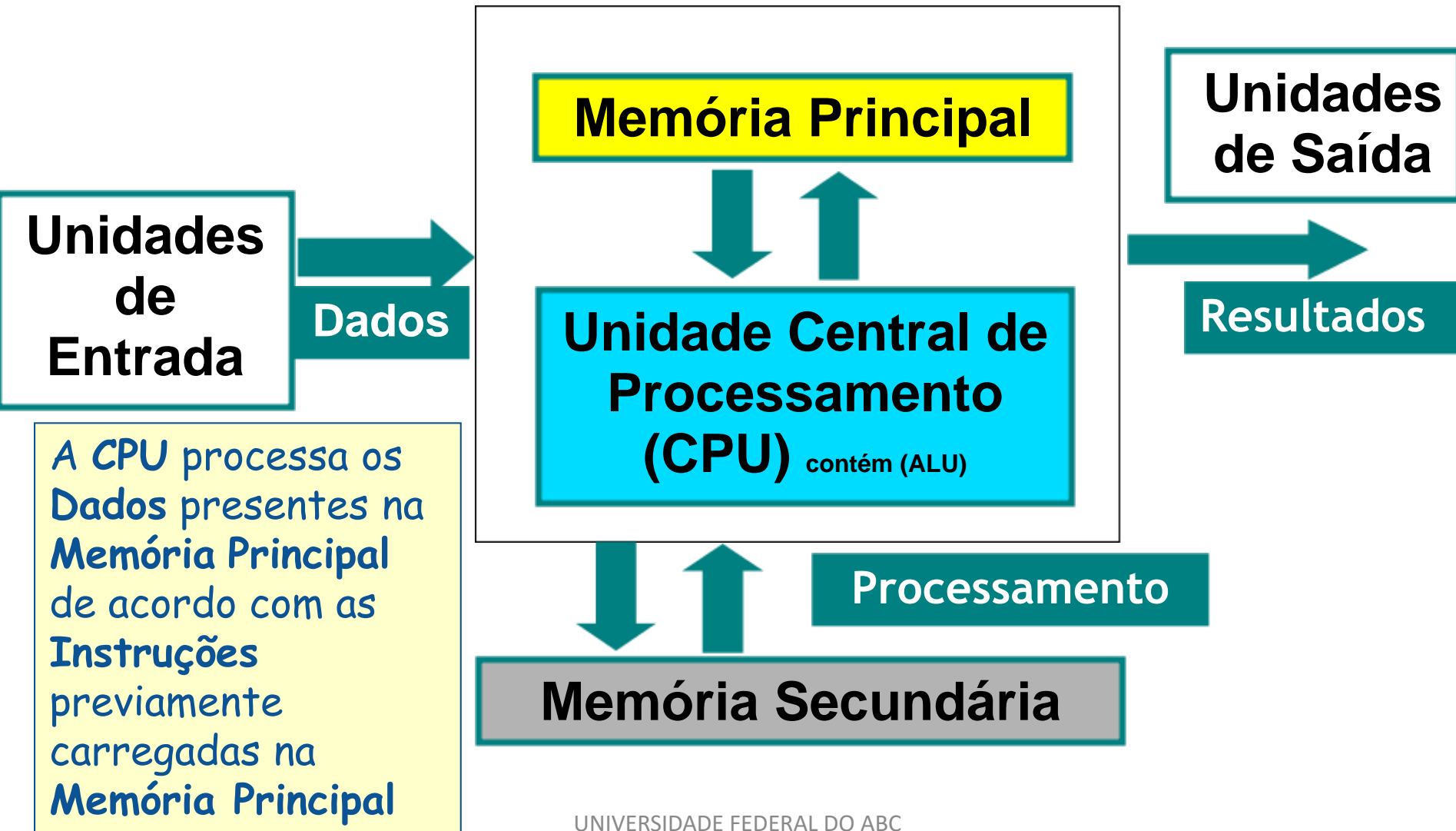
Software

Everything else
electronic/magnetic information:
programs
data

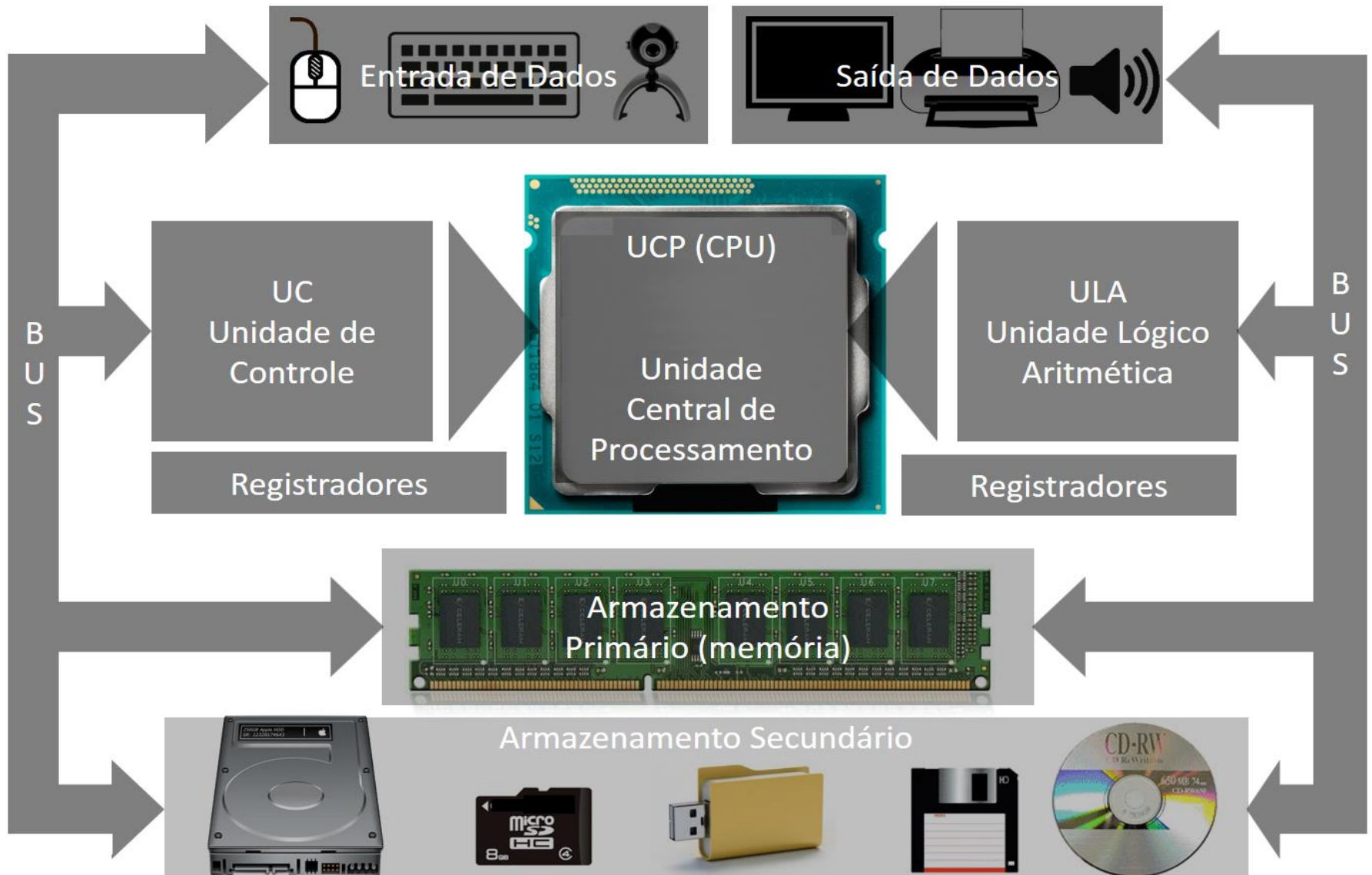


Arquitetura Von Neumann (1936)

A Memória Principal armazena tanto Instruções como Dados



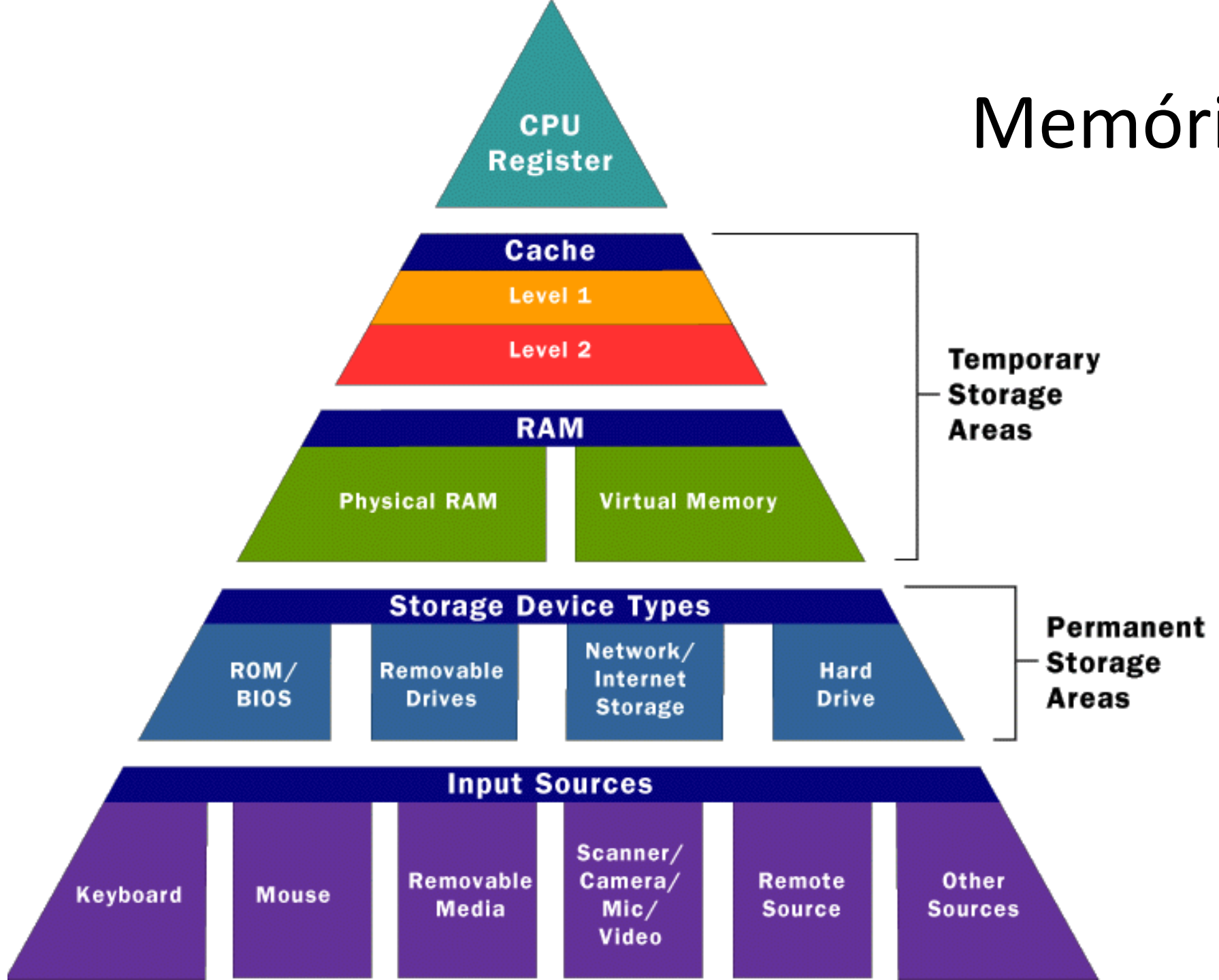
Von Neumann Architecture



Von Newmann Devices



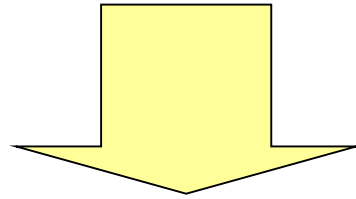
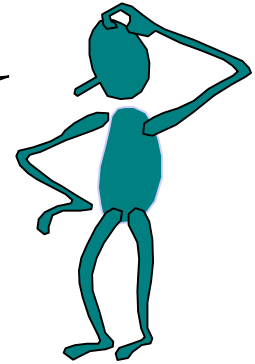
Memória



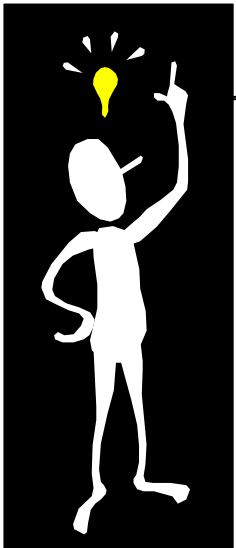
Kinds of Software

- ***OS (Operational System)*** - sistema operacional
 - Windows, Linux, iOS, Android, etc.
- ***Applications***
 - **Word processors** - Editores de texto: Word, WordPerfect, etc
 - **Spreadsheets** - Planilhas eletrônicas: Excel, Lotus 123, etc
 - **Presentations**: PowerPoint, etc
 - **Drawing** - Desenho: PaintBrush, CorelDraw, etc
- **Specific:**
 - Folha de Pagamento, Sistemas de Vendas, Sistema Acadêmico, etc.
- **Data:**
 - MP3, DOC, JPG, PNG, AVI, PPT, etc.
- **SOFTWARES** ↔ **PROGRAMAS DE COMPUTADOR**

How *softwares* are made?



Programming languages



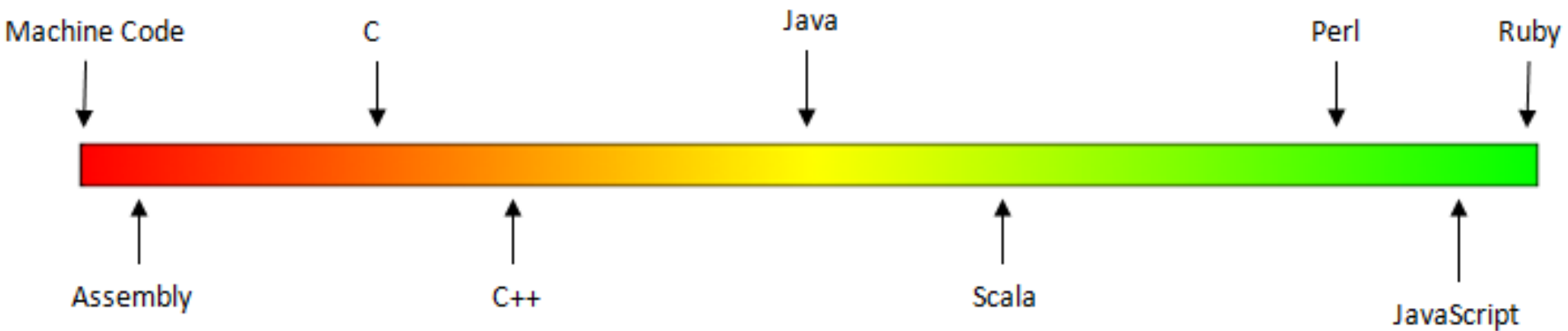
A programming language can be defined as:

- limited **set** of **symbols and commands**, used to write programs;
- standardized method to give instructions to a computer;
- Man-machine Interface, allowing men to express what they want the machine to do.

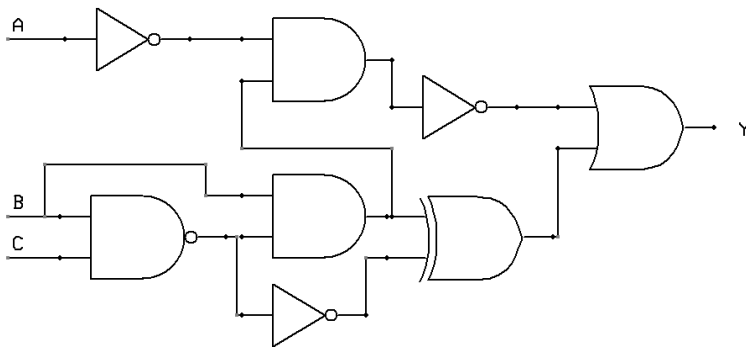
Uma linguagem de programação pode ser definida como:

- **conjunto** limitado **de símbolos e comandos**, utilizados para escrever programas;
- método padronizado para expressar instruções para um computador;
- por meio dela se estabelece uma comunicação Homem-computador, fazendo com que ele compreenda e execute o que o programador quer.

High, mid and low-level languages



A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



Linguagens de Programação



- Uma linguagem de programação pode ser de **baixo nível** (linguagem de máquina)

Linguagem Assembler

```
B0 FF MOV AL, 0FFh  
A2 00 20 MOV [2000h], AL
```

- Linguagens de **alto nível** apresentam uma **"sintaxe"** mais próxima da linguagem natural (humana)
- Usam **palavras reservadas** extraídas do vocabulário corrente (`int`, `public`, `if`, `else`, `while`, ...)
- O que faz o programa abaixo (em linguagem java) ?

```
class HelloWorld {  
    public static void main (String[] args) {  
        System.out.println ("Hello world!");  
    }  
}
```

High level Languages

Linguagens de Alto Nível

1957	FORTRAN	1975	Pascal	1986	CLP(R)
1958	ALGOL	1975	Scheme	1986	Eiffel
1960	LISP	1977	OPS5	1988	CLOS
1960	COBOL	1978	CSP	1988	Mathematica
1962	APL	1978	FP	1988	Oberon
1962	SIMULA	1980	dBase II	1990	Haskell
1964	BASIC	1983	Smalltalk	1995	Delphi
1964	PL/1	1983	Ada	1995	Java
1966	ISWIM	1983	Parlog		
1970	Prolog	1984	Standard ML		
1972	C	1986	C++		

Linguagens de Programação Compiladas

- Os programas escritos em linguagens de alto nível são **convertidos** para a linguagem de máquina através de um programa compilador, ou de um interpretador



- Trataremos de “**Compiladores**” e “**Interpretadores**” mais adiante”

Programming:

Programando:

Where to start?

Por onde começar?

Programming logic!

Lógica de programação

Aprendendo a Programar Computadores

- O principal componente de um programa de computador é a sua **“lógica”**, ou seja, o que o “computador deve fazer” para resolver um problema.
- Assim, é mais fácil aprender a programar (organizar a lógica) numa linguagem de **“alto nível”**
- **Qual é a linguagem de mais alto nível para um ser humano que fala naturalmente o Português?**

Aprendendo a Programar Computadores

- Parece mais simples dar os primeiros passos na arte de programar computadores, numa linguagem mais próxima da “linguagem natural”
- **Assim, o caminho que parece mais tranquilo é através de ALGORITMOS.**

Algorithms

DEFINITION

Algorithm: A **finite** sequence of instructions, **ordered** in a **logic** fashion in order to **solve** a particular problem or task



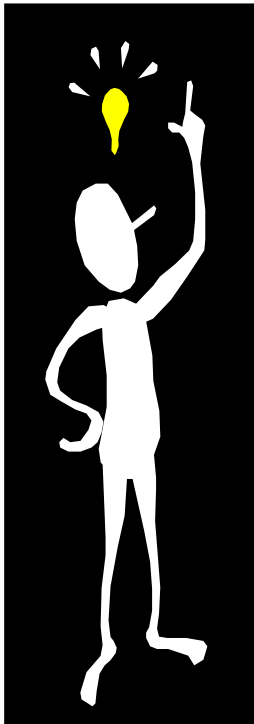
Sequence of instructions

Walkthrough

Recipe

ALGORITMO - DEFINIÇÃO

Algoritmo: Uma seqüência **finita** de instruções, **ordenada** de forma **lógica** para a **resolução** de uma determinada tarefa ou problema



Sequencia de instruções

Guia tecnico

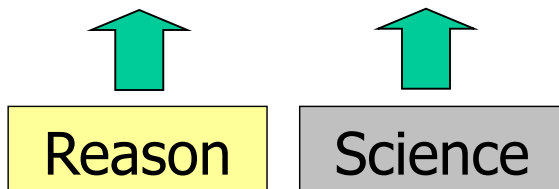
Receita

LÓGICA - LOGIC

- Senso comum – common sense:

Obviously

Logica ⇒ Logos + Ica (latin)



- Estuda o raciocínio/pensamento humano;
- Ciência dos argumentos.
 - Argumento é uma seqüência de enunciados, na qual um dos enunciados é a conclusão, derivado a partir dos outros enunciados **premissas**.

SILOGISMO - ARISTÓTELES

Todas as baleias são mamíferos. (premissa 1)

Alguns animais são baleias. (premissa 2)

Logo, alguns animais são mamíferos. (conclusão)

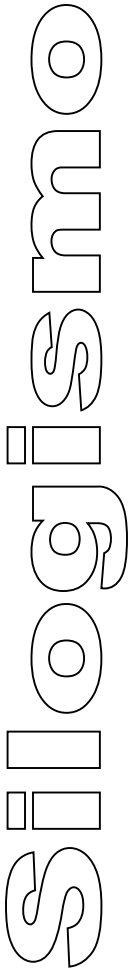
Raciocínio certo, baseado em premissas corretas

Todos os peixes vivem na água. (premissa1)

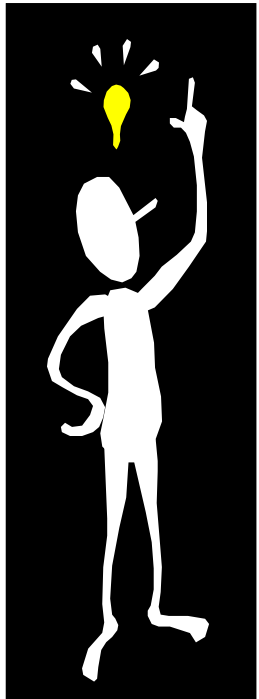
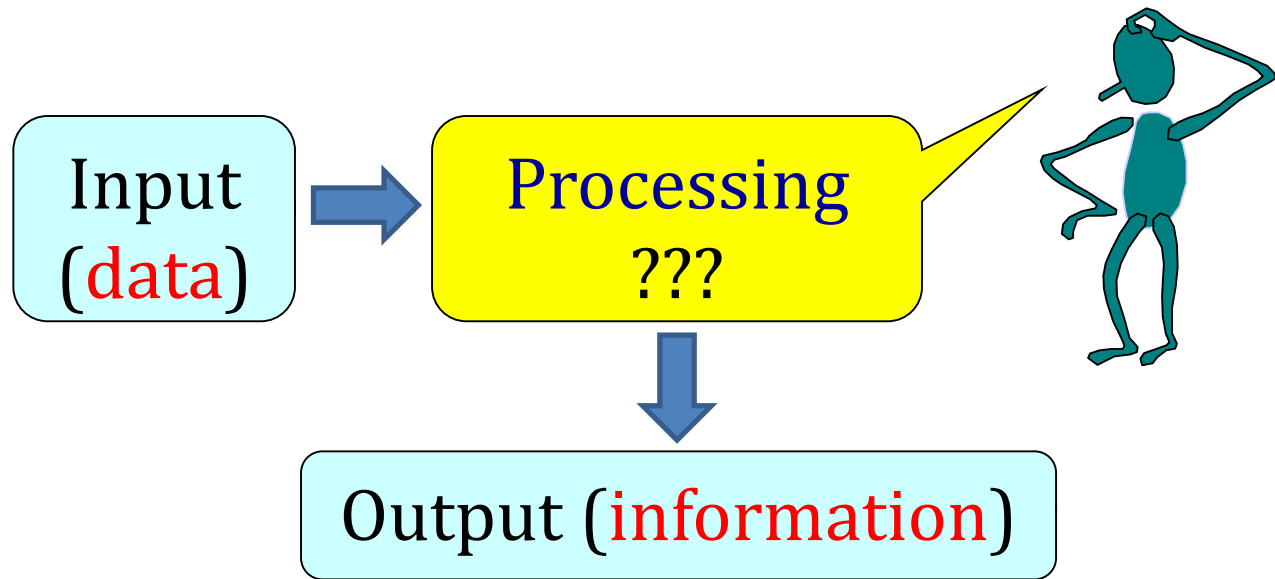
Golfinhos são peixes. (premissa 2)

∴ Golfinhos vivem na água. (conclusão)

Tanto a forma de argumento quando a conclusão são verdadeiras, apesar de esta última ter sido deduzida a partir de uma premissa falsa ("golfinhos são peixes").

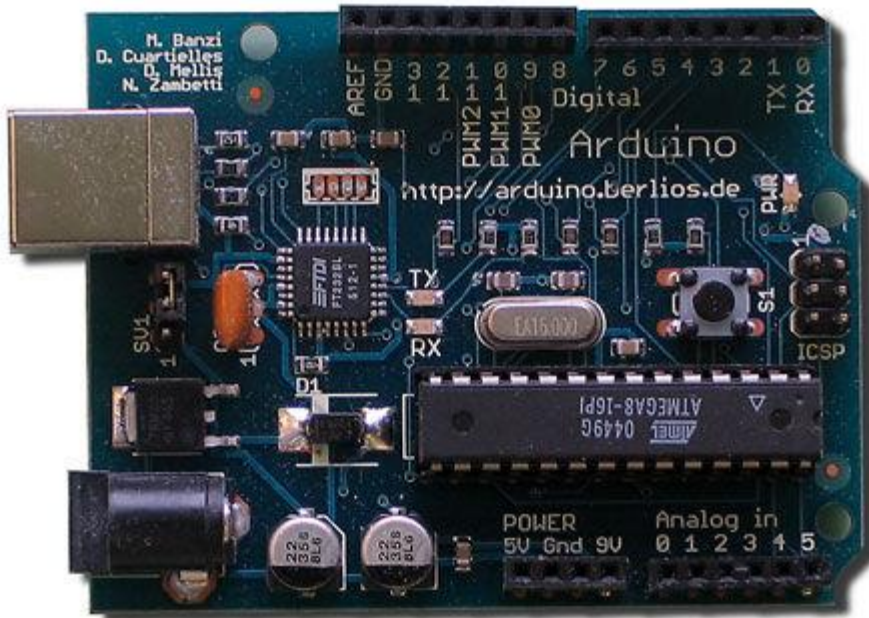


THINKING LIKE A PROGRAMMER



- Lógica de programação
- raciocínio que precisamos desenvolver
- para resolver um problema
- **encadeando pensamentos**
- para atingir determinado objetivo

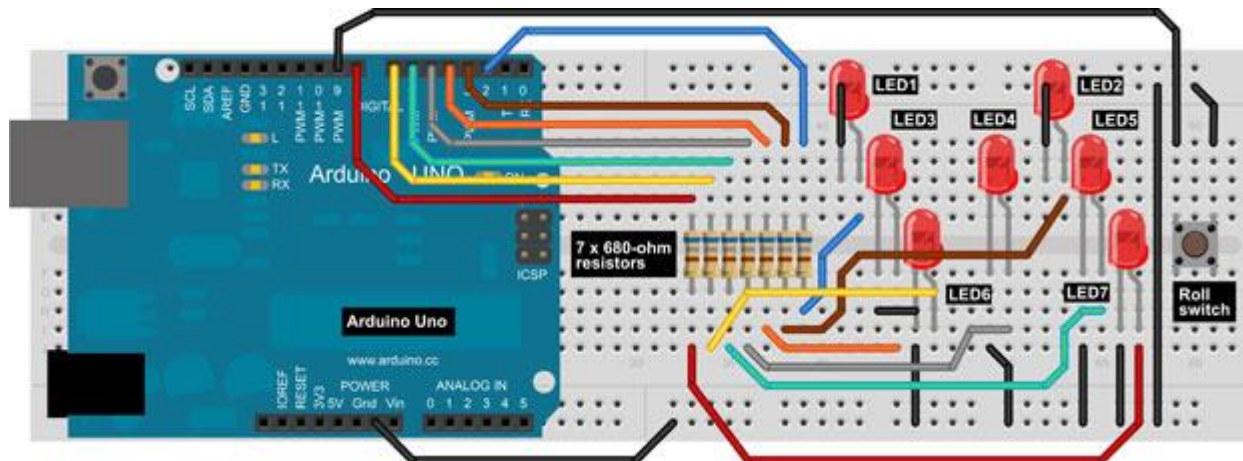
PROBLEM SOLVING



```
// configuration
const byte    LED_PIN    = 12;
unsigned long DELAY_TIME = 200;
const int     ITERATIONS = 5;

void setup ()
{
  Serial.begin (115200);
  pinMode (LED_PIN, OUTPUT);
} // end of setup

void loop ()
{
  for (int i = 0; i < ITERATIONS; i++)
  {
    digitalWrite (LED_PIN, HIGH);
    delay (DELAY_TIME);
    digitalWrite (LED_PIN, LOW);
    delay (DELAY_TIME);
  } // end of for
} // end of loop
```



EXAMPLES

- **SOLVED PROBLEMS:**

- <http://professor.ufabc.edu.br/~rogerio.neves/info/>

ALGORITHM

Representation
of a **solution to a given problem**
Through a **sequence of steps**

- Examples:
 - ❖ Receipts
 - ❖ Technical Manuals
 - ❖ Assembly Guides
 - ❖ Programs

PROBLEM

Chocolate cake

1st : What is the desired output?



OUTPUT - SAÍDA



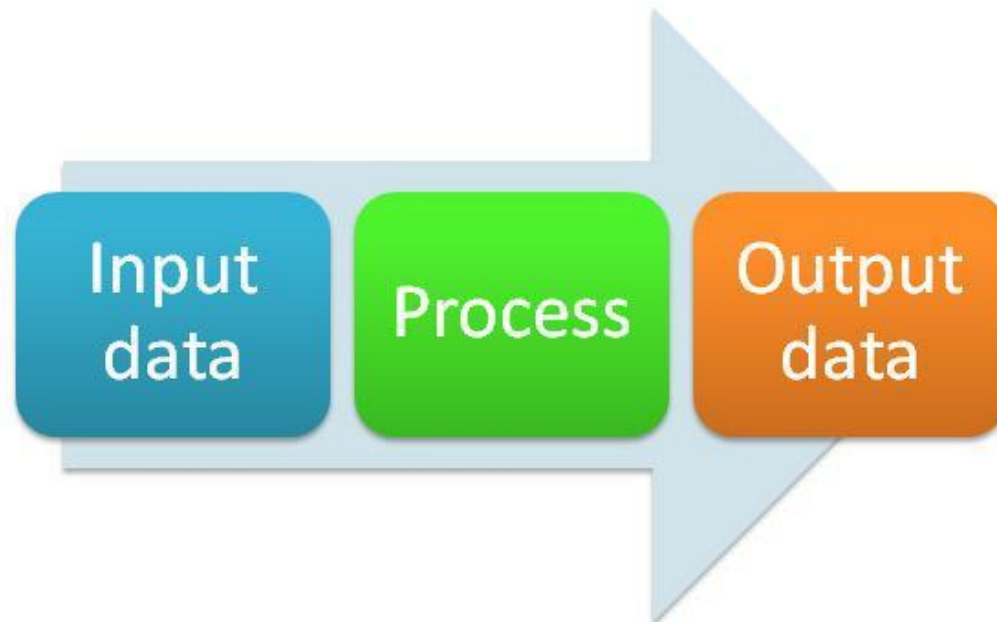
2nd : What is the input?

Input
(ingredients)
(insumos)
What do I need?



3rd : Processing

**What are the steps to go from
(input) to (output)**



ALGORITMO - EXEMPLO

Dados de Entrada

Ingredientes:

- ❖ 5 ovos
- ❖ 250 gramas de margarina cremosa
- ❖ 2 xícaras (chá) de açúcar
- ❖ 1 xícara (chá) de farinha de trigo
- ❖ 1 xícara (chá) de chocolate
- ❖ 200 gramas de côco ralado
- ❖ 1 copo de leite
- ❖ 1 colher (sopa) de fermento

ALGORITMO - EXEMPLO

Processamento

Modo de Preparo:

- ❖ **Bata** a margarina, as gemas e o açúcar até ficar cremoso
- ❖ **Junte** o leite, o côco e a farinha e continue batendo
- ❖ **Acrescente** o fermento e, por último, as claras em neve
- ❖ **Unte** uma forma com manteiga e leve ao forno para assar

* Repare o verbo no imperativo ou infinitivo, caracterizando um comando

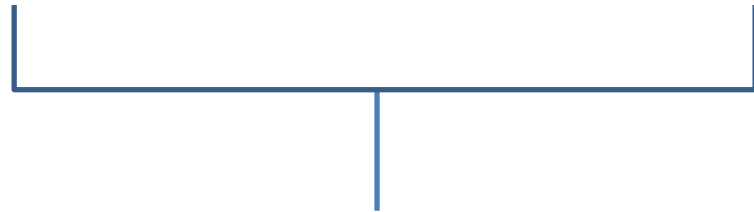
COMMANDS

DO



command

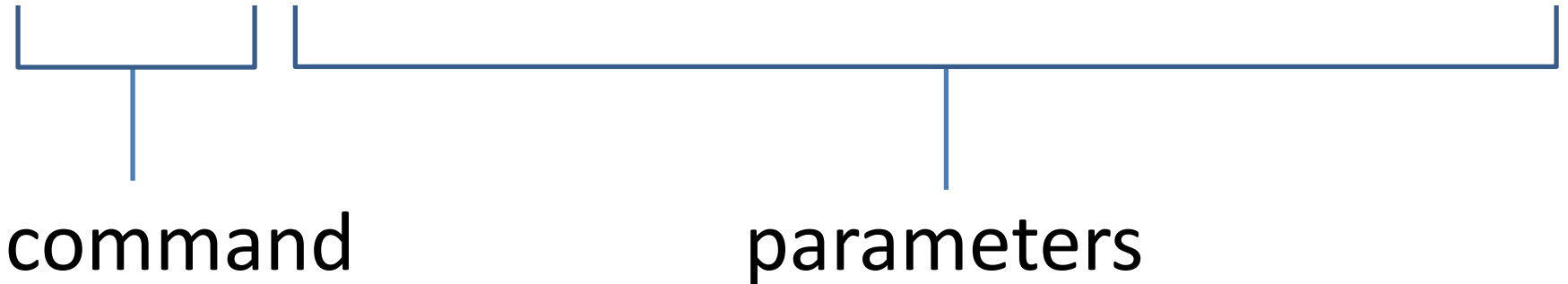
SOMETHING



parameters

COMMANDS

DO something with something else



Um algoritmo para realizar a troca de uma lâmpada

- **Pegue** uma escada
- **Posicione** a escada embaixo da lâmpada
- **Busque** uma lâmpada nova
- **Suba** na escada
- **Retire** a lâmpada velha
- **Coloque** a lâmpada nova

E se a lâmpada não estiver queimada?

1. Acionar o interruptor
2. Se a lâmpada não acender, então
 - ◆ Pegue uma escada
 - ◆ Posicione a escada embaixo da lâmpada
 - ◆ Busque uma lâmpada nova
 - ◆ Suba na escada
 - ◆ Retire a lâmpada queimada
 - ◆ Coloque a lâmpada nova

- Each person thinks differently
 - SO, there are several ways to solve a problem
 - SO, several different algorithms
 - All are correct, if they lead to the objective
-



Individual

Creativity

Experience



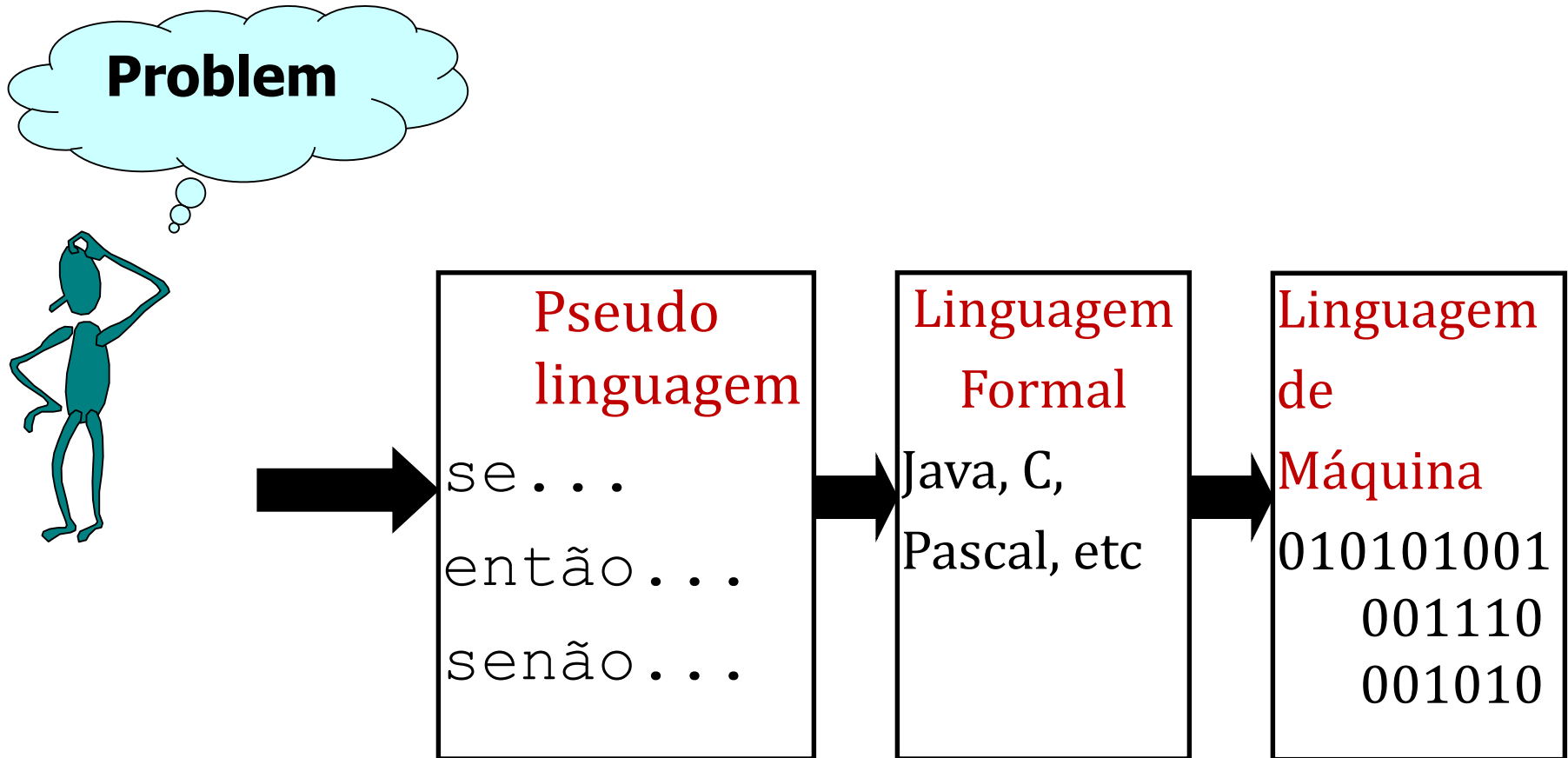
EFFICIENT CODING

Efficiency:

- ❖ Quick implementation
- ❖ Faster execution
- ❖ Less code
- ❖ Less effort
- ❖ Objectivity



PSEUDOCODE - PSEUDOLINGUAGEM



PSEUDOLINGUAGEM

Quando da definição de uma linguagem, precisamos definir:

- ❖ **Sua sintaxe**: como escrever os comandos e seus componentes
- ❖ **Sua semântica**: o significado de cada comando e conceito
- A **sintaxe** e a **semântica** de uma linguagem (ou pseudolingagem) devem considerar vários tipos de “ações”, como por exemplo:
 - ❖ Declarações de Variáveis
 - ❖ Operadores
 - ❖ Comandos de entrada e de saída
 - ❖ Comandos de seleção, repetição, etc.

BLOCO DE EXECUÇÃO

- Bloco de execução é um conjunto de ações que possui uma função bem definida
- O início de um bloco é marcado pela palavra **início**
- O fim de um bloco é marcado pela palavra **fim**

```
início
```

```
<declaração de variáveis>
```

```
<comandos>
```

```
fim
```

BLOCO DE EXECUÇÃO

O próprio algoritmo é um bloco de execução. A sintaxe da definição do bloco de um algoritmo é:

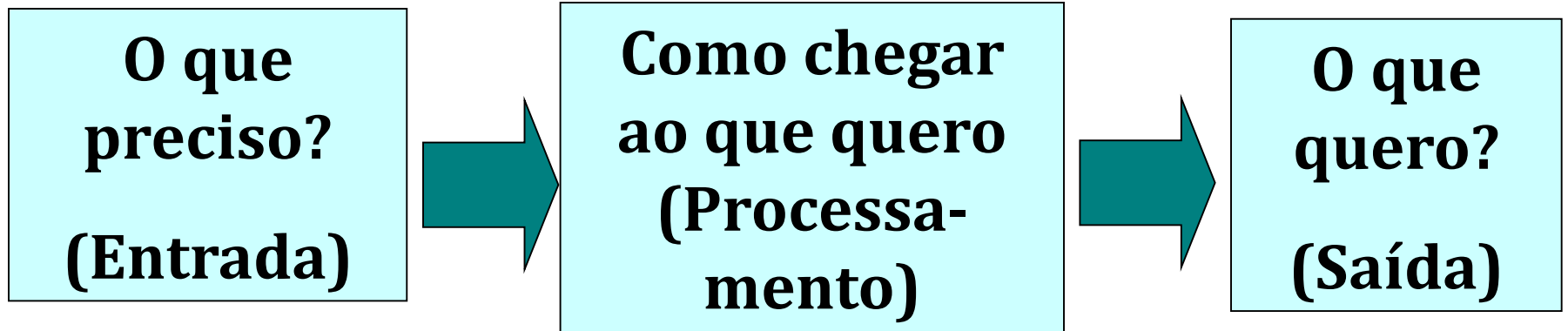
```
Algoritmo <NomeDoAlgoritmo>  
início  
    <declaração de variáveis>  
    <comandos>  
fimalgoritmo.
```

ALGORITMO – PASSOS PARA CONSTRUÇÃO

- Identificar o problema (objetivo) mediante leitura atenta de seu enunciado (**entendimento**)
- Identificar no enunciado os dados a serem fornecidos para o algoritmo (**entrada de dados**)
- Identificar os resultados que devem ser gerados na solução (**saída de dados**)
- Determinar o que deve ser feito para transformar as entradas nas saídas desejadas (**processamento**)



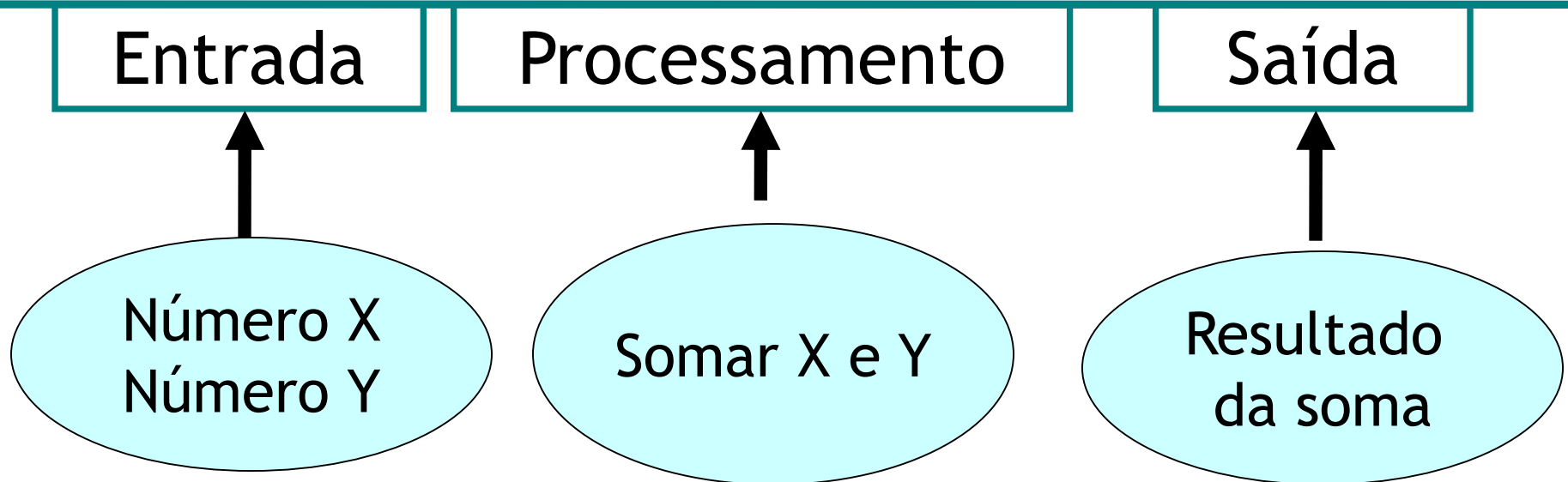
ALGORITMO – PASSOS PARA CONSTRUÇÃO



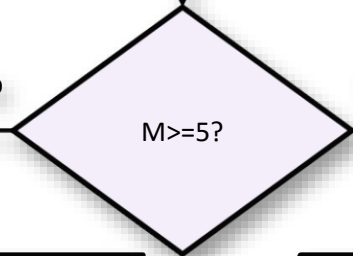
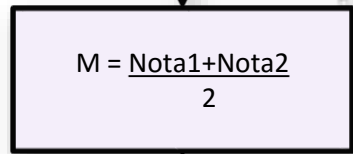
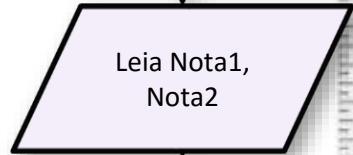
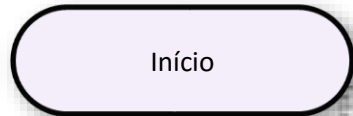
ALGORITMO - EXEMPLO

Escreva um algoritmo para **somar dois números** quaisquer

1. Leia o número X
2. Leia o número Y
3. Some X e Y
4. Mostre o resultado da soma

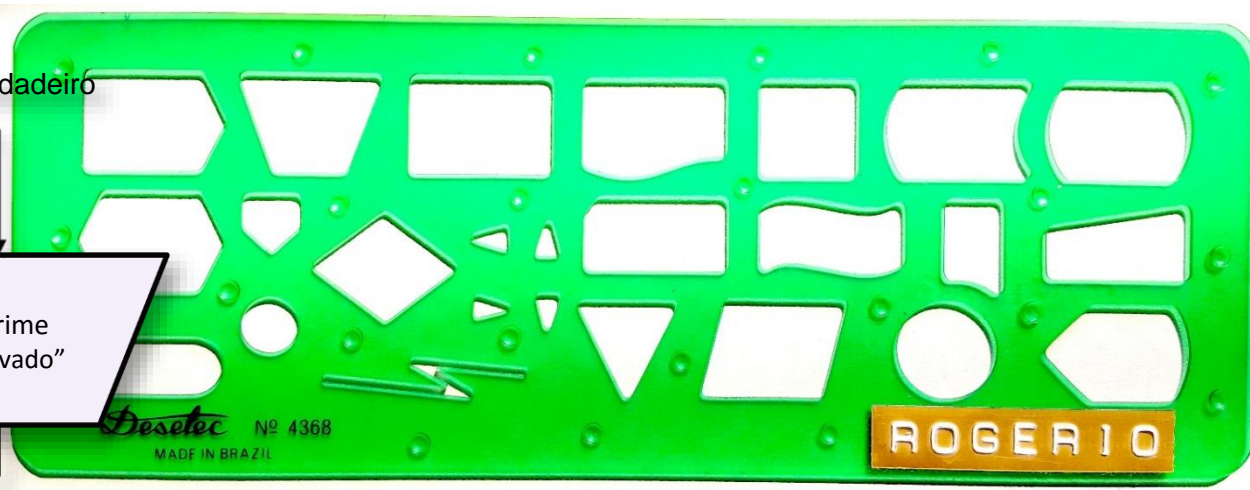
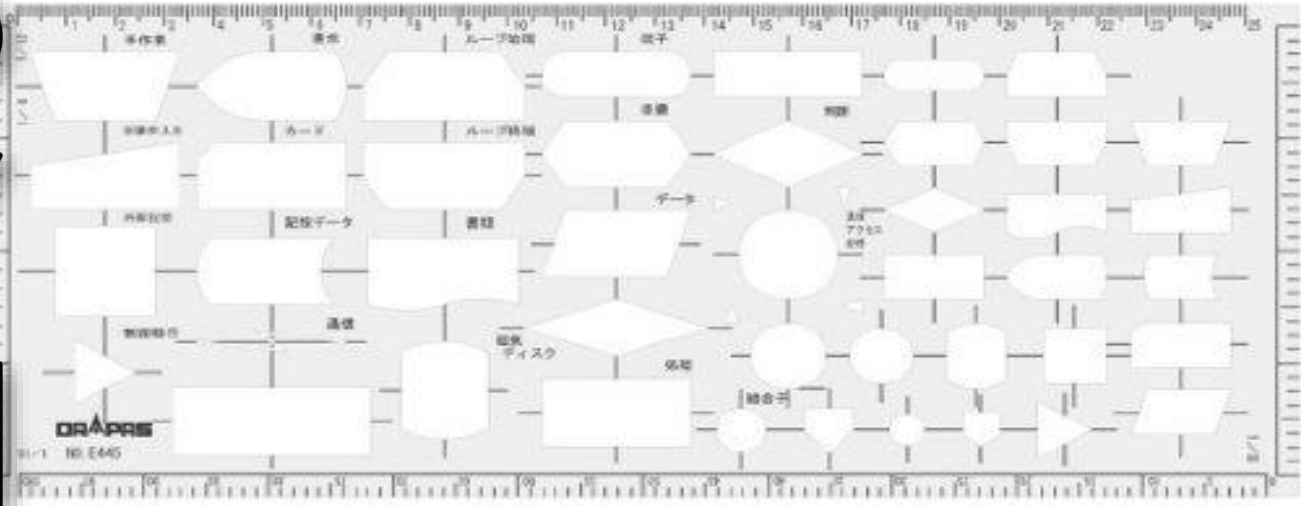
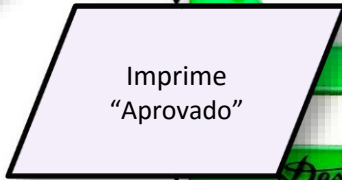
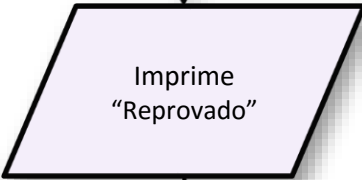


FLOWCHARTS - FLUXOGRAMAS



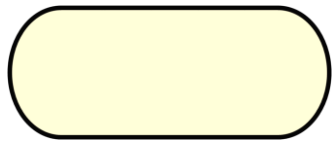
Falso

Verdadeiro

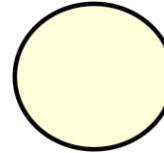


FLOWCHARTS - FLUXOGRAMAS

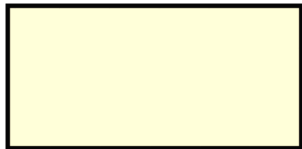
Tabela 1: Alguns símbolos de fluxogramas



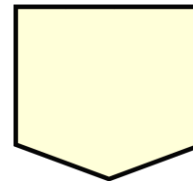
Terminador
Indica início/fim do programa



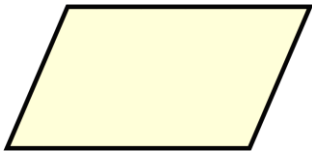
Conector (referência)
Conecta dois pontos separados na página



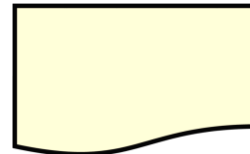
Processo/Operação



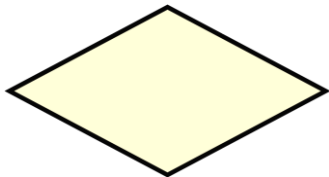
Referência externa
Conecta pontos em páginas distintas



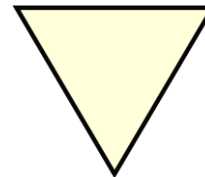
Entrada e saída de dados



Impressão



Tomada de decisão:
Define um desvio condicional de fluxo



Arquivo

LEITURAS SUGERIDAS

FORBELLONE, A. L. V.; EBERSPACHER, H. F., Lógica de Programação – A Construção de Algoritmos e Estruturas de Dados, Prentice Hall, 20??

- **CAPÍTULO 1** – Introdução a Lógica de Programação

Exercícios de Lógica

Exercício

Um lobo, uma cabra e uma couve têm de atravessar um rio num barco que transporta um de cada vez, incluindo o barqueiro. Como é que o barqueiro os levará para o outro lado de forma que a cabra não coma a couve e o lobo não coma a cabra?

Exercício

Eram dois homens que iam por um caminho. Um levava 8 litros de vinho em um barril e o outro levava 8 litros de vinho em dois garrafões, um de cinco litros e um de três. Beberam o vinho do barril de 8 litros e agora querem dividir os 8 litros de vinho restante igualmente, entretanto, eles só podem usar como medida, os próprios garrafões e o barril: 8, 5 e 3 litros. Como fazer?

Exercício

Eram dois homens que iam por um caminho. Um levava 8 litros de vinho em um barril e o outro levava 8 litros de vinho em dois garrafões, um de cinco litros e um de três. Beberam o vinho do barril de 8 litros e agora querem dividir os 8 litros de vinho restante igualmente, entretanto, eles só podem usar como medida, os próprios garrafões e o barril: 8, 5 e 3 litros. Como fazer?

1. Encher a vasilha de 3 litros.
2. Passar os 3 litros para a vasilha de 5 litros.
3. Encher outra vez a vasilha de 3 litros.
4. Encher a vasilha de 5 litros com a de 3, sendo que sobrar 1 na de 3.
5. Esvaziar a de 5 no barril.
6. Passar o litro da vasilha de 3 na de 5.
7. Encher a de 3 e esvaziar na de 5, que como já tinha 1, terá $1+3 = 4$.
8. Deixar o barril com 4 litros para o amigo.

Exercícios de “Algoritmos”

Exercícios

Descreva a seqüência de passos necessária para:

- Fritar um ovo
- Trocar um pneu furado
- Colocar um carro em movimento
- Jogar o jogo da velha

Algoritmos

- ❑ Conceitue Algoritmo (com suas palavras)
- ❑ Quais as “três” partes de um algoritmo?
- ❑ Como são compostas as **“frases”** (instruções) de um algoritmo?

Algoritmo - Exemplo

Fritar um ovo:

1. Pegue a frigideira, ovo, óleo e sal
2. Coloque o óleo na frigideira
3. Acenda o fogo
4. Coloque a frigideira no fogo
5. Espere o óleo esquentar
6. Quebre o ovo
7. Despeje o ovo no óleo quente
8. Coloque o sal
9. Retire quando estiver pronto
10. Desligue o fogo

Exercícios

Escreva um algoritmo para fazer uma ligação telefônica.

5 minutos

Algoritmos Computacionais

Algoritmo - Exemplo

Calcular a média aritmética a partir de 3 notas.

Identificando a solução:
ENTRADA, PROCESSAMENTO, SAÍDA!

- Qual a **SAÍDA** de Dados?
[que resultado o algoritmo dever produzir?]
– **média**
- Qual a **ENTRADA** de Dados?
[quais os insumos necessários para produzir a saída?]
– **3 notas** (nota1, nota2, nota3)

Algoritmo - Exemplo

Calcular a média aritmética a partir de 3 notas.

- Qual o **PROCESSAMENTO** necessário?
[para transformar as 3 notas na média?]
 - **Somar as 3 notas** (soma = nota1 + nota2 + nota3)
 - **Dividir a soma por 3** (media = soma / 3)
- O passo seguinte é organizar as “instruções” numa **sequência coerente**:
Entrada, Processamento e Saída

Escrevendo Algoritmos

- ❑ Quais as “instruções” necessárias para fazer:
 - ❑ **ENTRADA DE DADOS ?**
 - ❑ **PROCESSAMENTO?**
 - ❑ **SAÍDA DE DADOS ?**

- ❑ Para começar vamos aprender as instruções em na linguagem “PORTUGOL”
 - ❑ **PORTUGOL = “Português” + “Algoritmo”**

- ❑ Começemos pelo mais simples:
 - ❑ **SAÍDA DE DADOS**

SAÍDA DE DADOS

Saída de Dados

- EXIBE UMA MENSAGEM OU RESULTADO PARA O USUÁRIO NA TELA DO COMPUTADOR.
- A instrução para saída de dados em “PORTUGOL” é:
escreva (< mensagem>, <identificador>)

EXEMPLO:

- **escreva**(“Alô Mundo“)
- **escreva**(“A soma é igual a“, soma)

ENTRADA DE DADOS

Entrada de Dados

- INTERAGE COM O “**TECLADO**” DO COMPUTADOR OBTENDO INFORMAÇÕES QUE O USUÁRIO DIGITA
- A instrução para entrada de dados em “PORTUGOL” é:
leia (<identificador>)

EXEMPLO:

- **leia** (nota1)
- ATENÇÃO: Qualquer “dado” digitado no teclado precisa ser **armazenado na memória do computador!**
- Para tal, faz-se necessário a **RESERVA DE MEMÓRIA** através de **DECLARAÇÃO DE VARIÁVEIS**

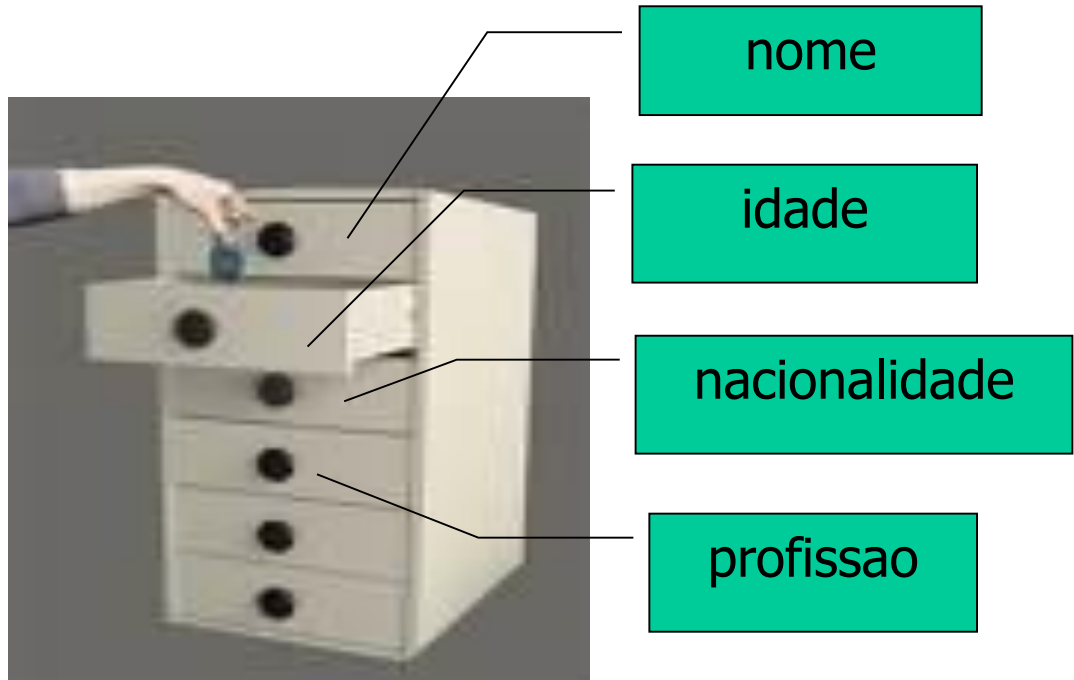
DECLARAÇÃO DE VARIÁVEIS

(reserva de memória)

PSEUDOLINGUAGEM - VARIÁVEIS

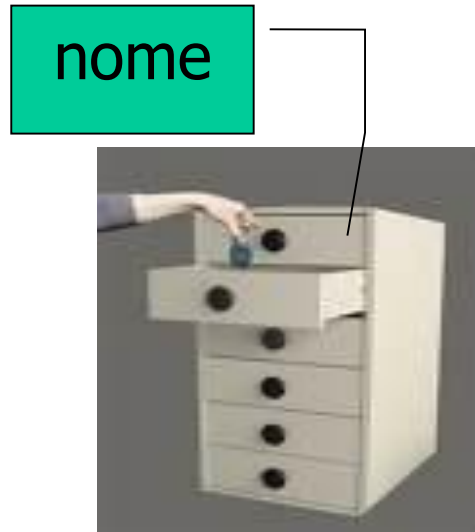
As linguagens de programação permitem que os usuário **atribuam nomes (identificadores)** para as posições de memória da máquina

Armário ↔
Memória do
computador



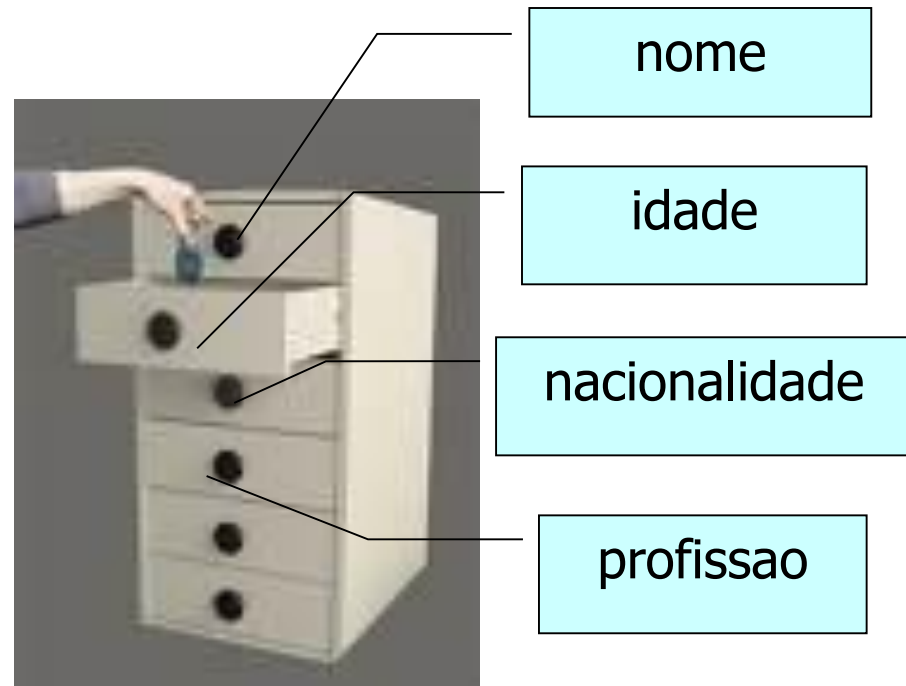
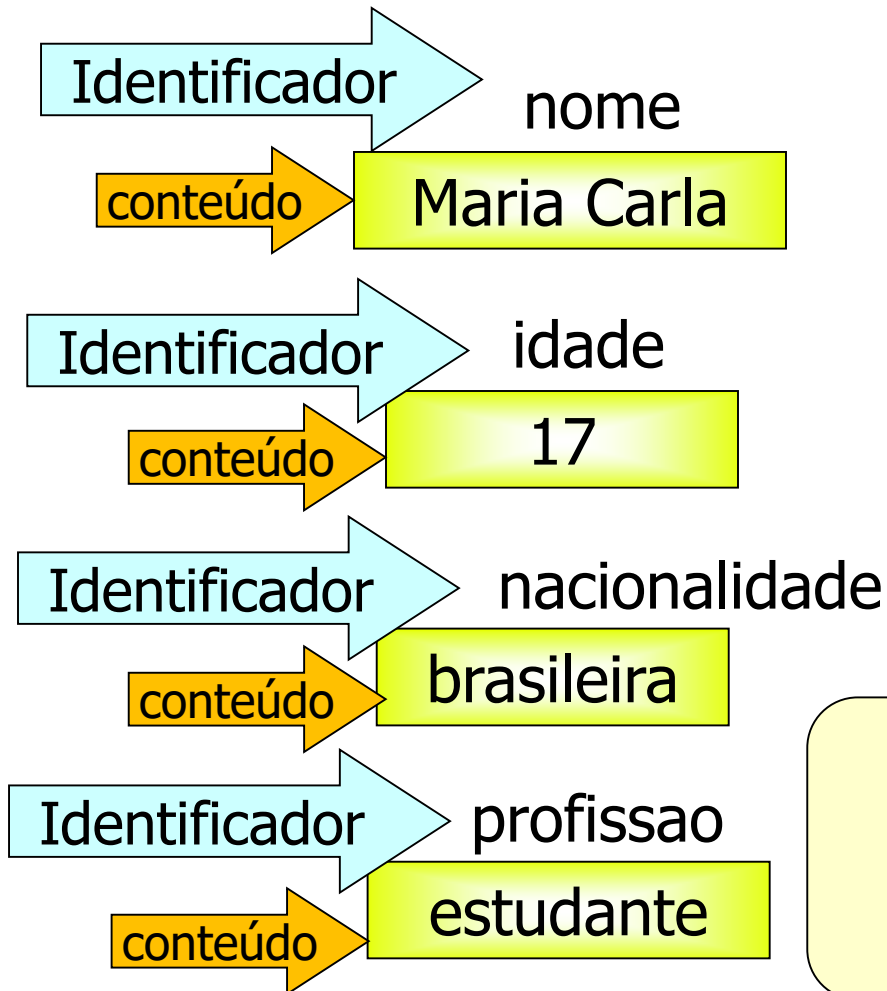
PSEUDOLINGUAGEM - VARIÁVEIS

Uma **variável** é um endereço da memória RAM, representada por um **identificador** (nome da variável), criado pelo usuário, cujo **conteúdo pode se alterar** no decorrer da execução do programa



- Uma variável é composta por dois elementos:
 - ❖ **Identificador**: nome dado pelo programador à variável
 - ❖ **Conteúdo**: valor atual da variável

PSEUDOLINGUAGEM - VARIÁVEIS



Uma variável assume apenas UM valor por vez

Declarando Variáveis

- ❑ Ao declarar uma variável é necessário informar ao computador que tipo de dado aquela “gaveta” pode armazenar
- ❑ Os tipos de dados básicos são:
 - Inteiro
 - Real
 - caracter

Forma Geral:

- <Identificador>: <tipo_de_dado>

Exemplos:

- Idade: **inteiro**
- nota1, nota2, nota3: **real**

PROCESSAMENTO

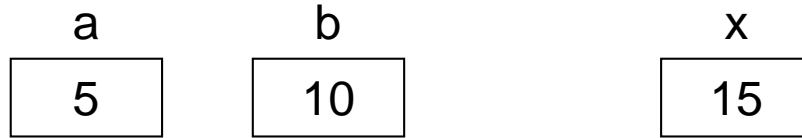
OPERADORES BÁSICOS

Processamento: Operações

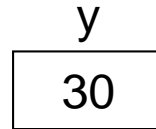
- ❑ O processamento de dados geralmente são realizados com o uso de **operadores**
- ❑ Os símbolos dos **operadores aritméticos** são:
 - adição → +
 - subtração → -
 - multiplicação → *
 - divisão → /
- ❑ Um operador importante é o de **operador de atribuição** que possibilita armazenar “valores” e “resultados” nas variáveis
- ❑ O símbolo do **operador de atribuição** é ←
 - Dependendo da linguagem de programação usa-se o “igual” (=)

Exemplos de Operações

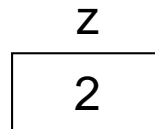
$$x \leftarrow a + b$$



$$y \leftarrow (a * 4) + b$$



$$z \leftarrow y / x$$



Algoritmo - Exemplo

Calcular a média aritmética a partir de 3 notas.

nota1, nota2, nota3, soma, media: real

1. Leia (nota1)
2. Leia (nota2)
3. Leia (nota3)
4. $soma \leftarrow nota1 + nota2 + nota3$
5. $media \leftarrow soma / 3$
6. Escreva (media)

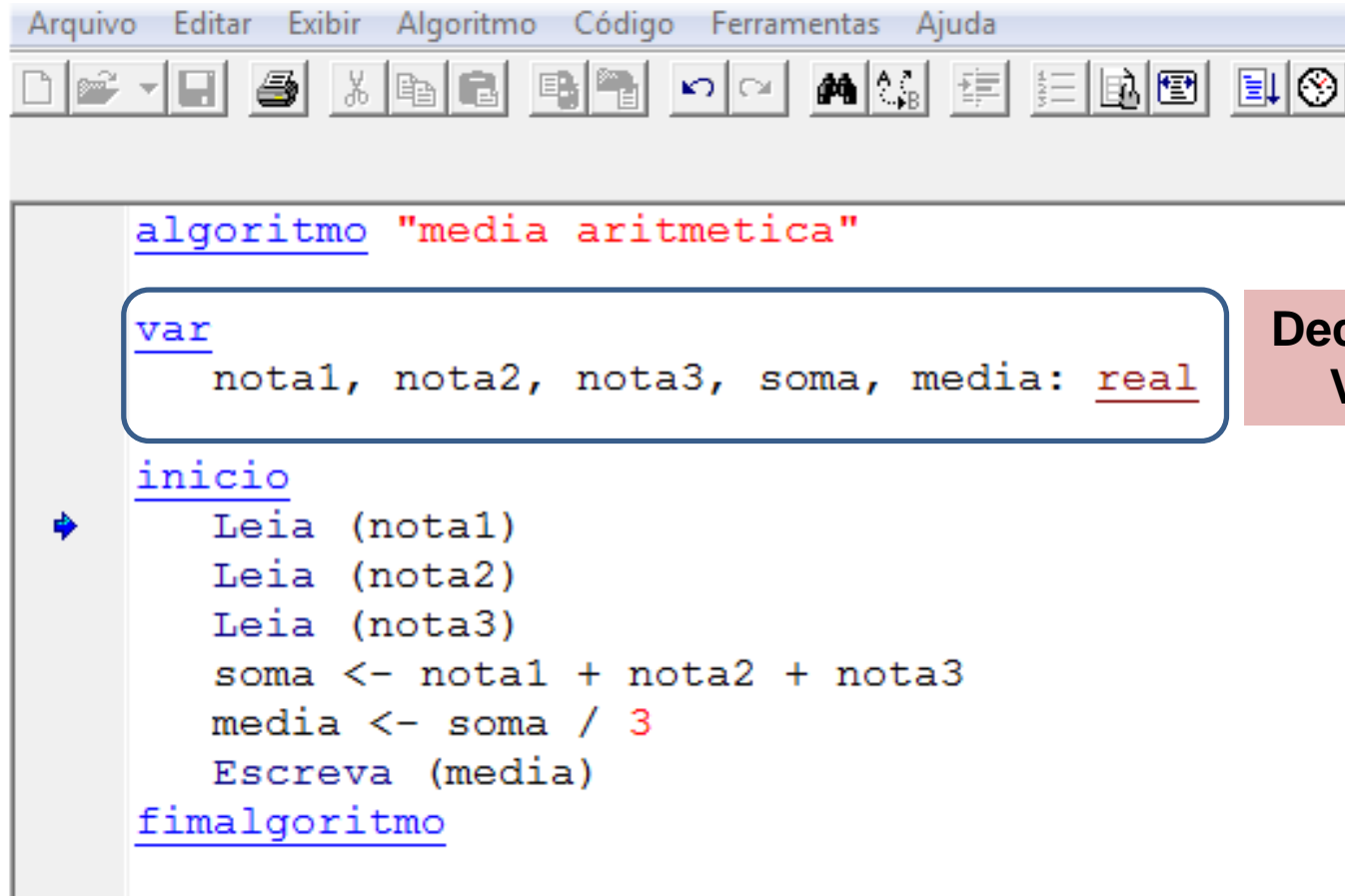
Note que para as operações não é necessário o uso de um verbo imperativo

Algoritmo

- Nas primeiras aulas vamos utilizar um Software chamado VisualG
- O VisualG EDITA, INTERPRETA e EDITA programas em pseudocódigo
- Depois vamos migrar para o JAVA
- O VisualG está disponível no Tidia-Ae (PI-2012-Geral)
- Pode ser baixado em sua página oficial:
<http://www.apoioinformatica.inf.br/visualg/objetivos.htm>

Exemplo no VisualG (Interpretador)

- ❑ O VisualG é um software interpretador de programas escritos em **PORTUGOL**



```
Arquivo  Editar  Exibir  Algoritmo  Código  Ferramentas  Ajuda
[Icons]
algoritmo "media aritmetica"
var
    nota1, nota2, nota3, soma, media: real
inicio
    Leia (nota1)
    Leia (nota2)
    Leia (nota3)
    soma <- nota1 + nota2 + nota3
    media <- soma / 3
    Escreva (media)
finalgoritmo
```

Declaração de Variáveis

Algoritmo – Solução mais amigável

Calcular a média aritmética a partir de 3 notas.

nota1, nota2, nota3, soma, media: real

1. **Escreva**("Digite a primeira nota :")
Leia (nota1)
2. **Escreva**("Digite a segunda nota :")
Leia (nota2)
3. **Escreva**("Digite a terceira nota :")
Leia (nota3)
4. $soma \leftarrow nota1 + nota2 + nota3$
5. $media \leftarrow soma / 3$
6. **Escreva** ("A média é = ", media)