

# Algoritmo e Programação



**Professor:** Gildárcio Gonçalves  
**E-mail:** [gildarciosousa@gmail.com](mailto:gildarciosousa@gmail.com)

# **Apresentação do Professor**

## **Formação Acadêmica**

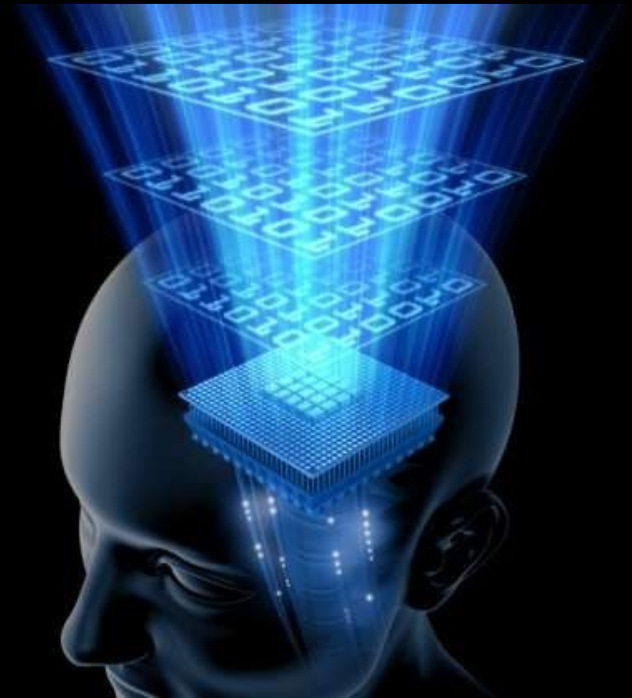
- **Mestrando em Engenharia Eletrônica e Computação no ITA**
- **Bacharel em Ciência da Computação**
- **Técnico em Informática em SENAI**

## **Experiência Profissional**

- **Professor do Centro Paula Souza (Etec) – Desde 2011;**
- **Pesquisador Colaborador do ITA no Grupo de Pesquisa em Engenharia de Software(GPES);**

# Agenda

- Lógica
- Algoritmos
- Linguagem Natural
- Pseudocódigo
- Fluxogramas
- Constantes, Variáveis, Palavras Reservadas, Identificadores
- Tipos de Dados
- Operadores



O que é

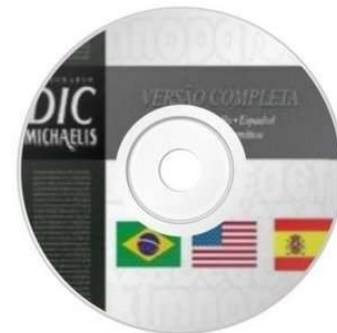
um

Problema?

# Conceito de Problema

## Problema (Dicionário Michaelis):

- Questão matemática proposta para ser resolvida.
- Questão difícil, delicada, suscetível de diversas soluções.
- Qualquer coisa de difícil explicação, mistério, enigma.
- Dúvida, questão.



# Exemplos de Problemas

- Trocar uma lâmpada
- Trocar o pneu de um carro
- Preparar-se para uma prova

**Sempre que nos  
deparamos com um  
problema buscamos  
o quê?**





# Exemplo de Solução

- Por exemplo, para trocar uma lâmpada devemos:
  - Adquirir uma lâmpada nova;
  - Localizar a lâmpada queimada;
  - Retirar a lâmpada queimada;
  - Colocar a lâmpada nova;
  - Descartar a lâmpada queimada.



**O que orientou a  
obtenção dos  
procedimentos  
para a solução  
proposta?**



# Lógica

# O que é a Lógica?

A lógica é o ramo da Filosofia e da Matemática que estuda os métodos e princípios que permitem fazer distinção entre raciocínios válidos e não válidos, determinando o processo que leva ao conhecimento

verdadeiro

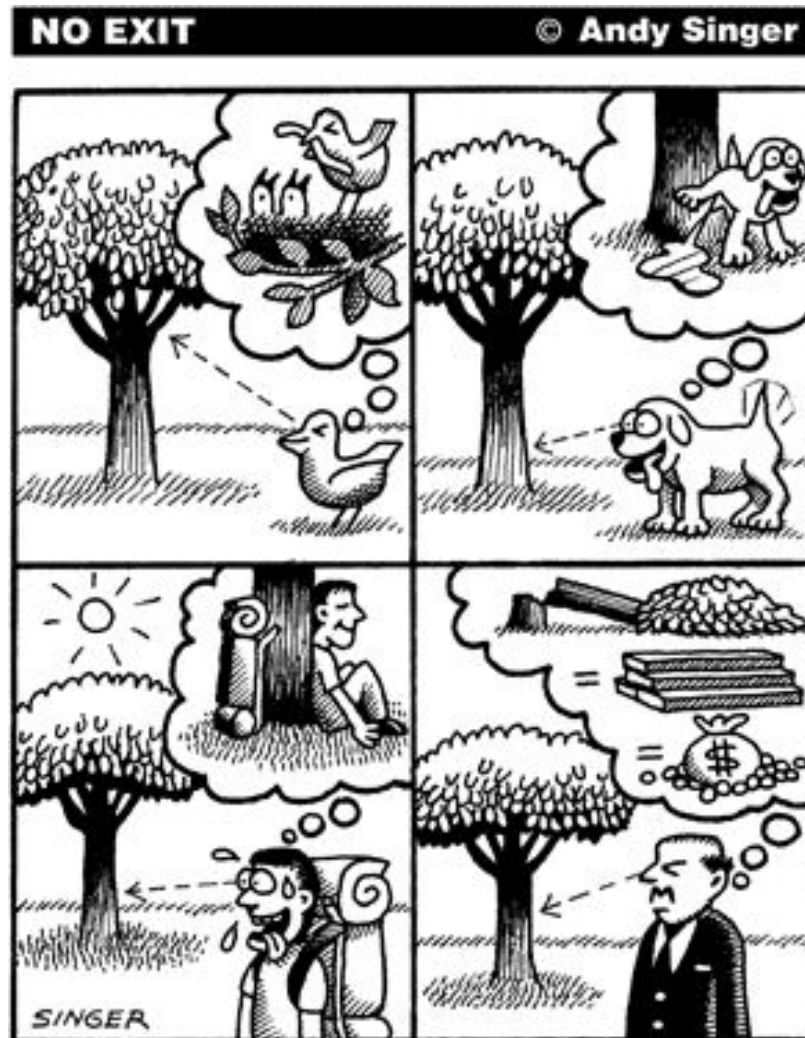


# Tome nota

- Ninguém ensina outra pessoa a pensar, mas sim a desenvolver e aperfeiçoar esta técnica, com persistência e constância.
- A lógica é primordial na solução de problemas e para alcançar objetivos com eficiência e eficácia.
- Desenvolvemos softwares



# Cada um pensa de uma forma



**O que construimos  
ao utilizar a lógica  
para solucionar um  
problema?**

# ALGORITMO



# Conceito de Algoritmo

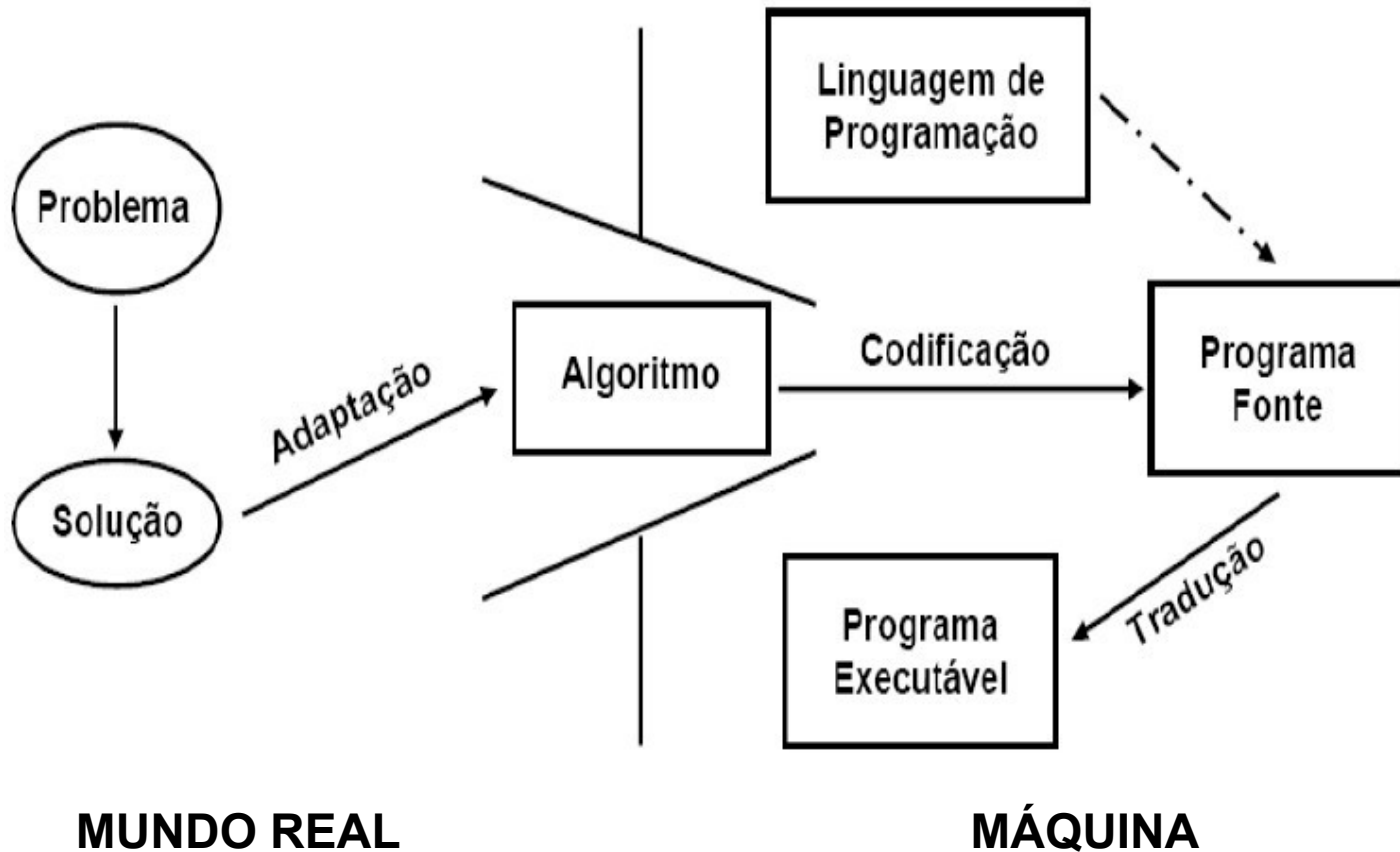
**Uma sequência de passos bem definida que deve ser seguida para a realizar uma tarefa ou solucionar um problema.**



# Métodos de Representação de Algoritmos

- Do ponto de vista computacional um algoritmo será implementado em uma linguagem de computação gerando um programa, o qual visa instruir um computador a executar determinada tarefa.
- O algoritmo tem como papel fundamental de ligar dois mundos (real e computacional). A atividade de programação tem início com a construção do algoritmo.

# Funcionalidade do Algoritmo



# Métodos de Representação de Algoritmos

## **Descrição Narrativa:**

Adquira uma lâmpada nova e localize a lâmpada a ser trocada. Em seguida retire a lâmpada queimada e coloque a lâmpada nova. Após a troca descarte a lâmpada queimada.

# Descrição Narrativa

- Vantagens
- Desvantagens



# Descrição Narrativa

Um exemplo de ambigüidade presente em uma sentença na linguagem natural é:

O policial escutou o barulho da porta.

Esta frase pode ter pelo menos três interpretações:

- 1 - O policial escutou o barulho produzido pela porta.**
- 2 - O policial estava junto à porta e escutou o barulho.**
- 3 - O policial escutou o barulho que veio através da porta.**

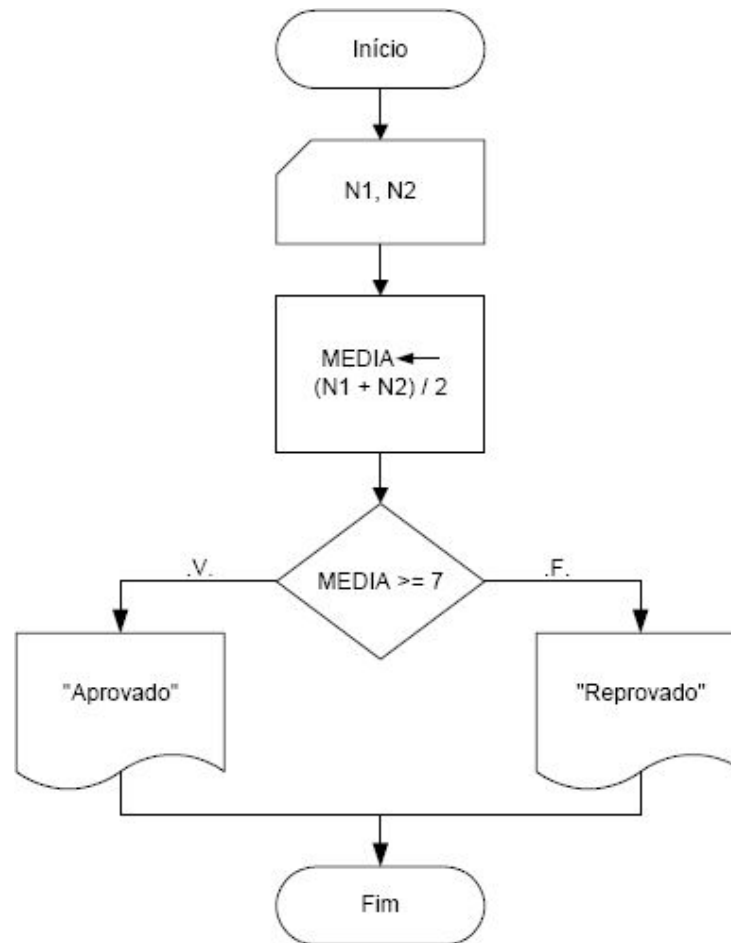


# Métodos de Representação de Algoritmos

- Fluxograma
- Pseudocódigo |Português estruturado  
|Portugol



# Fluxograma





# Exemplo de Pseudocódigo

## ALGORITMO: TROCAR UMA LÂMPADA

**PASSO 1:** Desligar a caixa de energia **PASSO 2:** Pegar a lâmpada nova **PASSO 3:** Pegar a escada  
**PASSO 4:** Posicionar a escada embaixo da lâmpada queimada  
**PASSO 5:** Subir na escada com a lâmpada nova  
**PASSO 6:** Retirar a lâmpada queimada **PASSO 7:** Colocar lâmpada nova **PASSO 8:** Descer da escada  
**PASSO 9:** Ligar a caixa de energia  
**PASSO 10:** Ligar o interruptor  
**PASSO 11:** Guardar a escada  
**PASSO 12:** Jogar a lâmpada velha no lixo



# Métodos de Representação de Algoritmos

- **Fluxograma:**

- Vantagem – a representação gráfica é mais concisa que a representação textual.
- Desvantagem – é necessário aprender a simbologia dos fluxogramas.

- **Português Estruturado:**

- Vantagem – sua transcrição para qualquer linguagem de programação é quase que direta.
- Desvantagem – é necessário aprender as regras do pseudocódigo.

# Estrutura de um Algoritmo

Algoritmo <nome do  
algoritmo>

<declarações iniciais> Inicio

<Corpo do algoritmo>

Fim

# Conceitos Básicos de Algoritmos

Antes de nos aprofundarmos nos métodos de representação de algoritmos, devemos ter de forma clara a compreensão de alguns conceitos como:

- Constante;
- Variável;
- Identificador;
- Palavra-reservada;
- Entrada;
- Saída.



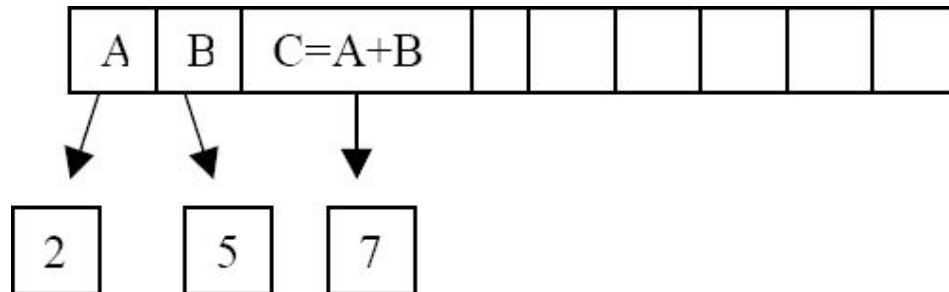
# Constantes

- Constantes são endereços de memória destinados a armazenar informações fixas, inalteráveis durante a execução do programa.

**Exemplo:**  
**PI = 3.1416**

# Variáveis

- Variáveis são endereços de memória destinados a armazenar informações temporariamente.



# Identificadores

- São os nomes dados as variáveis, constantes e programas.

**Exemplo:**  
**Var idade: Inteiro**



# Regras de Identificadores

- Não podem ter nomes de palavras reservadas (comandos da linguagem);
- Devem possuir como 1º caractere uma letra ou Underscore ( \_ );
- Ter como demais caracteres letras, números ou Underscore;
- Não possuir espaços em branco;
- A escolha de letras maiúsculas ou minúsculas depende da linguagem

# Palavras Reservadas

- São identificadores predefinidos que possuem significados especiais para o interpretador do algoritmo.

inicio	senao	para	enquanto
var	logico	se	ate
faca	inteiro	enquanto	real

# Tipos de Dados

- **NUMÉRICO** (Inteiro ou Real)
- **LITERAL**
- **LÓGICO**



# Exemplos de Armazenamento

- **Inteiro:** 0, -12, 15
- **Real:** 0.12, -1.13, 25.17
- **Literal:** “Claudio”, “a”, “4° Andar”
- **Lógico:** .V. ou .F.

# Operadores

- Os operadores são meios pelo qual incrementamos, decrementamos, comparamos e avaliamos dados dentro do computador. Temos três tipos de operadores:
  - Operadores Aritméticos
  - Operadores Relacionais
  - Operadores Lógicos

# Aritméticos

- **OPERAÇÃO SÍMBOLO**

- Adição +
- Subtração -
- Multiplicação \*
- Divisão /
- Exponenciação \*\*

# Relacionais

- **Descrição Símbolo**
- Igual a =
- Diferente de  $\neq$  ou #
- Maior que  $>$
- Menor que  $<$
- Maior ou igual a  $\geq$
- Menor ou igual a  $\leq$



# Lógicos

- E / AND Uma expressão AND (E) é verdadeira se todas as condições forem verdadeiras
- OR/OU Uma expressão OR (OU) é verdadeira se pelo menos uma condição for verdadeira
- NOT Um expressão NOT (NÃO) inverte o valor da expressão ou condição, se verdadeira inverte para falsa e vice-versa.

# Estrutura de um Algoritmo

## Algoritmo Somar

**Var** n1, n2, total : Inteiro;

### Inicio

**Escreva**("Digite um número");

**Leia**(n1);

**Escreva**("Digite outro número");

**Leia**(n2);

total  $\leftarrow$  n1 + n2;

**Escreva**("A soma dos números é {0}.", total);

**Fim**

# Estrutura de um Algoritmo

## **Algoritmo** CalcularMedia

**Var** n1, n2, media: **real** **Inicio**

**Escreva**("Digite um número");

**Leia**(n1);

**Escreva**("Digite outro número");

**Leia**(n2);

    media  $\leftarrow$  (n1 + n2) / 2;

**Se** (MEDIA  $\geq$  7) **entao**

**Escreva** ("Aprovado");

**Senao**

**Escreva** ("Reprovado");

**Fim\_se**

**Fim**

# Estrutura de um Algoritmo

**Algoritmo MostrarTabuada** Var i, n,  
total: real

**Inicio**

**Escreva**("Digite um número");

**Leia**(n1);

**Para** i  $\leftarrow$  0 **ate** 10 **faca**

total  $\leftarrow$  n \* i;

**Escreva**("{0} X {1} = {2}",n,i, total);

**Fim\_Para** **Fim**