

# Introdução à Lógica de Programação

## OBJETIVO

- Aprender conceitos básicos para desenvolver programas para computadores;
- Exemplos: sistemas bancários, sistemas de restaurantes, cálculos avançados entre outros.
- E qual o roteiro para desenvolver programas?

**Problema → Solução → Algoritmo → Programa → Resultado**

## LÓGICA

- Do grego “*logos*”;
- Palavra, pensamento, ideia, argumento, relato, razão;
- Ordenação dos raciocínios;
- Ciência dos princípios do pensamento;
- Para usar: necessário domínio sobre o pensamento bem como saber pensar !!



## LÓGICA

Serve para guiar nosso pensamento;

Habilidade necessária para programação;

Ensina a pensar de forma estruturada;

Desenvolve e aperfeiçoa a técnica de pensamento;

Ajuda a seguir um pensamento lógico e matemático.



## LÓGICA

Todo mamífero é um animal.  
Todo cavalo é um mamífero.  
Portanto todo cavalo é um animal!

## LÓGICA

Se elefantes podem subir em árvores, então 3 é um número irracional.

\* Proposições e operadores lógicos

## RACIOCÍNIO LÓGICO

- Influenciam:
  - Calma
  - Conhecimento
  - Experiência
  - Criatividade



## DESAFIOS

## LÓGICA DE PROGRAMAÇÃO

- Essencial para desenvolvimento de sistemas de computadores
- Técnica de encadear instruções para atingir determinado objetivo
- Seqüência de instruções, que devem ser seguidas para se cumprir uma determinada tarefa

## DEFINIÇÕES

- Condições iniciais (estado de problema)
  - como as coisas são
- Objetivos (estado desejado)
  - como as coisas deveriam ser
- Recursos
  - meios ou métodos para transformar um estado de problema desde as condições iniciais até os objetivos

## ALGORITMO

Algoritmo é um texto (do tipo receita de bolo) onde cada linha contém uma ação primitiva (ação elementar passível de execução por um humano ou uma máquina).

A função do algoritmo, quando executado, é a de agir (operar) sobre os dados, transformando-os em informações (Pinto,1990).



## ALGORITMO

Algoritmo: Conjunto de regras e operações bem definidas e ordenadas, destinadas à solução de um problema, ou de uma classe de problemas, em um número finito de etapas.

Seqüência de passos (ações) com o objetivo de resolver um problema.



## ALGORITMO

Especificação precisa (não ambígua) de um comportamento que visa resolver um problema bem definido.

Seqüência finita de instruções precisas que podem se executadas mecanicamente num período de tempo finito e com quantidade de esforço finito.



## CARACTERÍSTICAS

- Finitude: um algoritmo deve terminar após um número finito de passos, de tempo e de recursos;
- Exatidão: passos definidos com precisão.
- Entradas e saídas bem determinadas: o algoritmo deve operar sobre um ou mais valores externos e algum resultado deve ser produzido.



## CARACTERÍSTICAS

- Efetividade: ele deve ser realizável!  
Todas as instruções devem ser:
  - Tão simples quanto possível.
  - Suficientes para resolver o problema.
  - Executáveis com lápis e papel.

## O QUE É NECESSÁRIO?

Raciocínio Lógico  
+  
Conhecimentos e Técnicas  
+  
Ferramentas



## COMO?

Experiência adquirida com a prática (exercícios)



Livros, tutoriais, etc.



Software, framework, etc.

## Organizando as coisas....

1. Leia atentamente o enunciado;
2. Retire do enunciado a relação das entradas de dados;
3. Retire do enunciado a relação das saídas de dados;
4. Determine as ações que levarão a atingir o resultado desejado;
5. Construa o algoritmo;
6. Execute o algoritmo.

## ALGORITMO

Exemplo 1: Preparar um bolo

- Início: Coloque no liquidificador a laranja picada (sem casca e sem o pavio central), os ovos, o óleo e o açúcar
- Ligue o liquidificador e bata por 3 minutos
- Despeje o conteúdo em uma vasilha e adicione a farinha e o fermento
- Misture bem

## ALGORITMO

- Asse em forno médio e pré-aquecido por 35 minutos
- Faça a cobertura, despejando o suco de laranja com açúcar e o leite sobre o bolo ainda quente
- Fim: Deixe esfriar e sirva

## De novo....

1. Leia atentamente o enunciado;
2. Retire do enunciado a relação das entradas de dados;
3. Retire do enunciado a relação das saídas de dados;
4. Determine as ações que levarão a atingir o resultado desejado;
5. Construa o algoritmo.

\* Os softwares em geral nada mais são do que algoritmos “traduzidos” para uma linguagem de programação.

## MÉTODOS DE REPRESENTAÇÃO

1. Descrição narrativa
2. Fluxograma
3. Pseudocódigo

## DESCRIÇÃO NARRATIVA

- Uso da linguagem natural
- Problemas:
  - Má interpretação;
  - Ambiguidade;
  - Imprecisões.

## DESCRIÇÃO NARRATIVA

- Exemplos: Criar um algoritmo para:
  - Trocar um pneu

## DESCRIÇÃO NARRATIVA

- Exemplos: Criar um algoritmo para:
  - Trocar um pneu
  - Trocar uma lâmpada

## DESCRIÇÃO NARRATIVA

- Em uma receita culinária....
  - "sal a gosto" x "1/4 colher (chá) de sal "
  - "bata bem" x "bata até a massa ficar homogênea "
  - "forno médio" x "forno a 220°C"

## FLUXOGRAMA

- Baseia-se na utilização de formas geométricas diferentes que indicam ações (instruções, comandos) distintas;
- Evita má interpretação;
- É a forma de representar os algoritmos de maneira gráfica.

## FLUXOGRAMA

- Um fluxograma deve ser claro, simples e de fácil leitura. Não deve haver ambigüidade na interpretação de um fluxograma
- A direção de descrição do fluxo de um sistema é de cima para baixo ou da esquerda para a direita
- Apenas uma linha de fluxo deve partir de ou chegar a um terminador.

# FLUXOGRAMA

Aritméticos	
Símbolo	Operação
+	adição
-	subtração
*	multiplicação
/	divisão
%	módulo

Relacionais	
Símbolo	Significado
<	menor que
>	maior que
<=	menor ou igual à
>=	maior ou igual à
==	igual
!=	diferente

Lógicos	
Símbolo	Operação
&&	AND
	OR
!	NOT

# FLUXOGRAMA



Início / fim do fluxograma



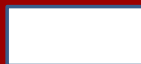
Saída de Dados, Resultado



Entrada de dados envolvidas



Conector



Cálculo, processamento e atribuição de valores



Orientação de fluxo



Representa um teste lógico, tomada de decisão



Atribuição

## FLUXOGRAMA

Exemplo 1: Faça o fluxograma de um algoritmo onde o usuário digita 2 números e o algoritmo exiba o resultado da soma deles no final.

Exemplo 2: Faça o fluxograma de um algoritmo onde o usuário digita as duas notas do aluno e o algoritmo exiba se o aluno está aprovado ou reprovado. Considere média 6 para aprovação



## PSEUDOCÓDIGO

Forma genérica de escrever um algoritmo, utilizando uma linguagem simples.

Não pode ser executado num sistema real .

\* Existem alguns interpretadores  
que compilam pseudocódigo em português.



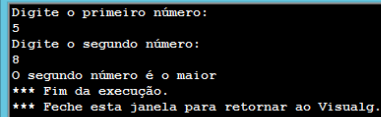


# PSEUDOCÓDIGO

```

algoritmo "exemplo-1"
var
  num1,num2: inteiro
inicio
  // Seção de Comandos
  escreval("Digite o primeiro número:")
  leia(num1)
  escreval("Digite o segundo número:")
  leia(num2)
  se(num1>num2)entao
    escreva("O primeiro número é o maior")
  senao
    escreva("O segundo número é o maior")
fimse
fimalgoritmo

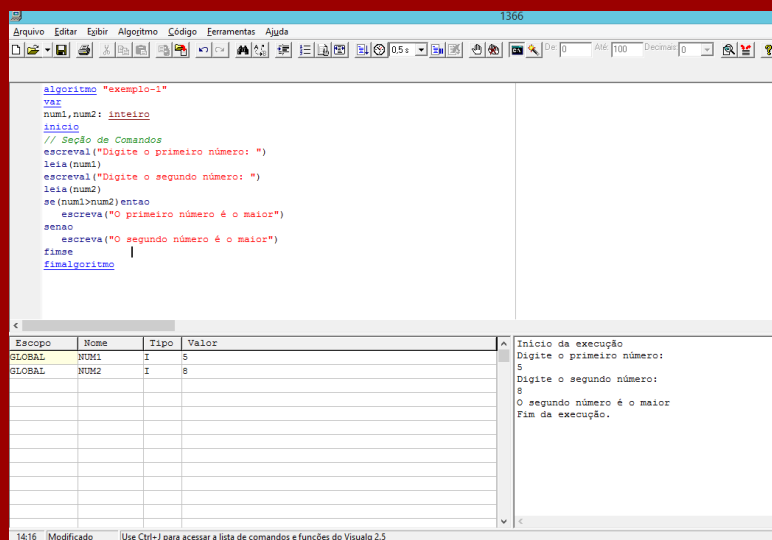
```



```

Digite o primeiro número:
5
Digite o segundo número:
8
O segundo número é o maior
*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.

```



Escopo	Nome	Tipo	Valor
GLOBAL	NUM1	I	5
GLOBAL	NUM2	I	8

14:16 | Modificado | Use Ctrl-F para acessar a lista de comandos e funções do Visualg 2.5

## VisualG