

# Introdução a Computação

Lógica de Programação  
Algoritmos

## Algoritmos

Vantagens da programação estruturada:

- 1 – Conjunto reduzido de estruturas de controle
- 2 – Não existem desvios de controle
- 3 – Simplifica a estrutura do programa
- 4 – Melhora a confiabilidade do programa
- 5 – Aumenta a legibilidade do programa
- 6 – Disciplina a programação
- 7 – Aumenta a produtividade da programação
- 8 - Simplifica a manutenção

## Algoritmos

### Qualidades de um algoritmo:

- 1 - Perfeitamente definido – descreve exatamente o que deve ser feito
- 2 - Não Ambíguo – não deixa duvidas
- 3 – Eficaz – resolve o problema em qualquer situação
- 4 – Eficiente – usa o mínimo de recursos

3

## Algoritmos

### Construção de um algoritmo:

- 1 – Analise preliminar
- 2 – Solução
- 3 – Teste de qualidade
- 4 – Alteração
- 5 – Produto final

4

## Algoritmos

### **Passos para a resolução de um problema:**

- Passo 1: Entenda o Problema!
- Passo 2: Identifique as Entradas.
- Passo 3: Identifique as Saídas.
- Passo 4: Identifique as transformações necessárias.
- Passo 5: Construa o fluxograma.
- Passo 6: Teste o algoritmo/fluxograma
- Passo 7: Converta para PORTUGOL
- Passo 8: Teste o algoritmo/PORTUGOL

5

## Algoritmos

- **Constante:**  
Elemento que não muda. Informação que não muda. Ex:  $PI = 3,141592654$
- **Variável:**  
Elemento que pode assumir diferentes valores.

6

## Variável

### Variáveis e Tipos de Dados:

**Cada um dos dados operados no programa e associado a uma variável. Cada variável tem por sua vez, uma faixa de valores que ela pode assumir. A esta Faixa de valores damos o nome de Tipo.**

7

## Tipos de Dados

### Variáveis e Tipos de Dados:

**Cada um dos dados operados no programa e associado a uma variável. Cada variável tem por sua vez, uma faixa de valores que ela pode assumir. A esta Faixa de valores damos o nome de Tipo.**

8

## Tipos de Dados

### **Simple:**

- **inteiros,**
- **reais,**
- **caracteres,**
- **booleanos**

### **Estruturado:**

- **vetor (array),**
- **registro (record),**
- **pilha (stack),**
- **fila (queue),**
- **arvore(tree)**