



**Profa. Michele Fúlvia Angelo**

**PGCA 028 – Tópicos Especiais em Tecnologia  
Computacional I – Introdução à Programação de  
Computadores  
Aula 1**

**Universidade Estadual de Feira de Santana**

# Sumário

- Etapas da criação de um programa
- Formas de representação de um algoritmo, com foco em Fluxograma e Pseudocódigo

# Um Programa de Computador

- Um programa nada mais é do que uma sequência de instruções que possui significado para o computador.
- O nosso foco será entender como criar um programa.

# Criação de um Programa

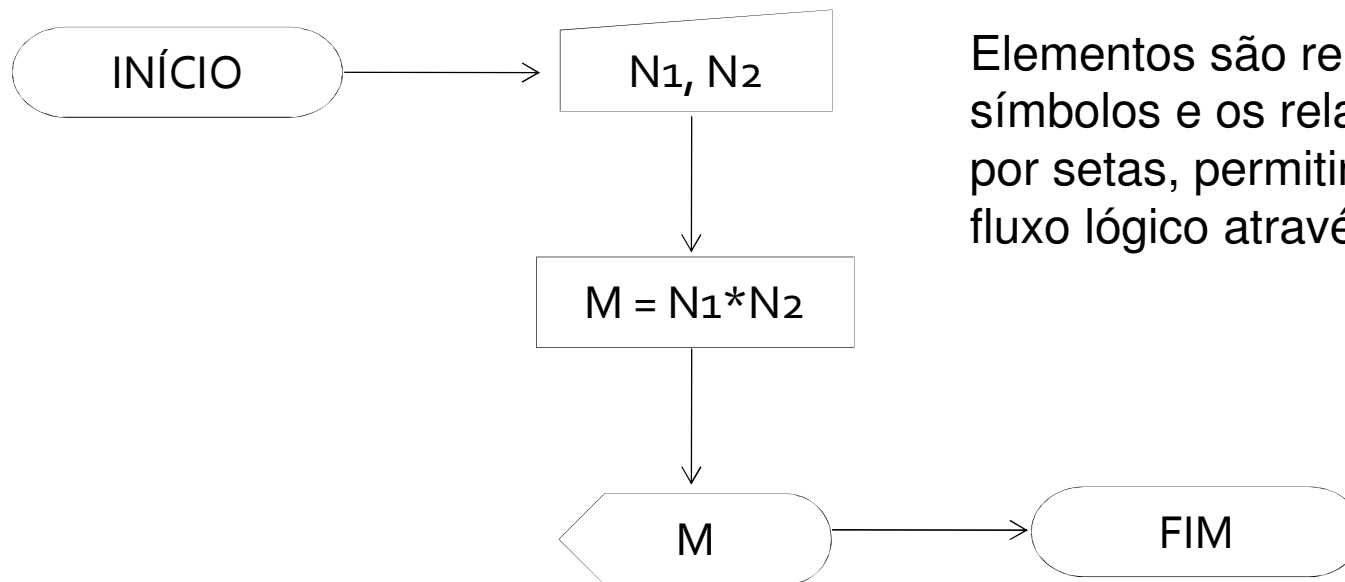
- Uma etapa da criação do programa é a descrição do **Algoritmo**, ou seja, dos passos a serem realizados.
- Um algoritmo pode ser escrito de três formas:
  - Descrição Narrativa
  - Fluxograma
  - Pseudocódigo

# Descrição Narrativa

- Escrever aquilo que se quer fazer, assim como em uma receita de bolo:
  - Passo 1 – Receber os dois números que serão multiplicados
  - Passo 2 – Multiplicar os números
  - Passo 3 – Mostrar o resultado obtido na multiplicação

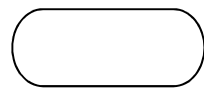
# Fluxograma

- São diagramas que representam graficamente o algoritmo, enfatizando os passos individuais e o fluxo da execução.



Elementos são representados por símbolos e os relacionamentos por setas, permitindo visualizar o fluxo lógico através do tempo.

# Fluxograma



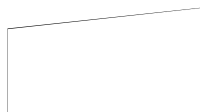
início / fim



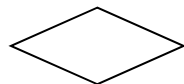
ação



conector



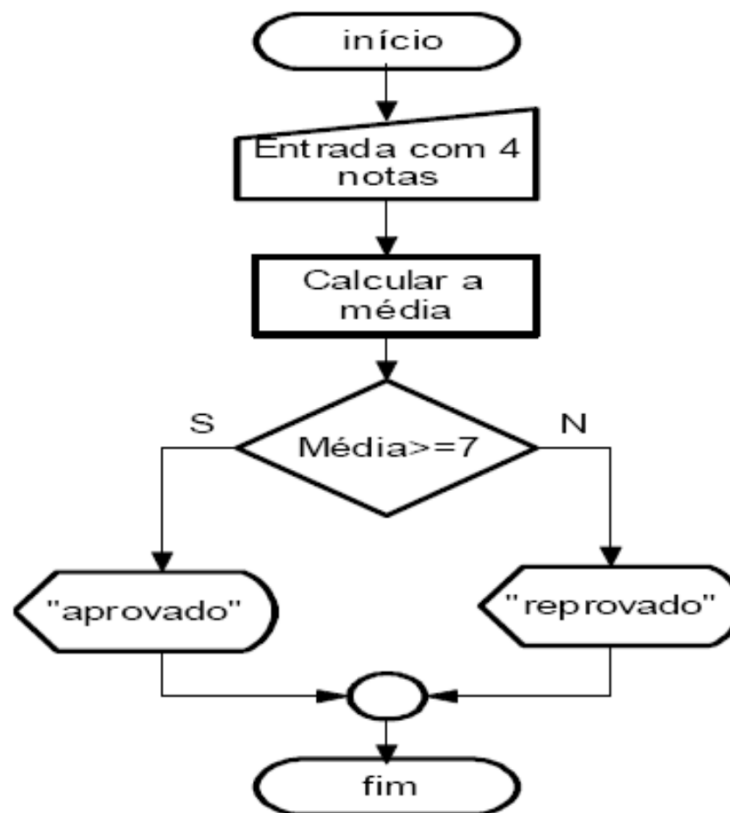
leitura



decisão



impressão



# Pseudocódigo

- Escrever (em português) o programa utilizando algumas regras:

```
ALGORITMO  
DECLARE N1, N2, M NUMÉRICO  
ESCREVA "Digite dois números"  
LEIA N1, N2  
 $M \leftarrow N1 * N2$   
ESCREVA "Multiplicação = ", M  
FIM_ALGORITMO.
```



# Princípios de Resolução de Problemas

- 1) Os diagramas de blocos devem ser feitos e quebrados em vários níveis;
- 2) Os primeiros devem conter apenas as idéias gerais, deixando para as etapas posteriores os detalhamentos necessários;
- 3) Sempre que possível deve ser feito de cima para baixo e da esquerda para a direita;
- 4) Transcrever o diagrama de bloco em pseudocódigo.

# Princípios de Resolução de Problemas

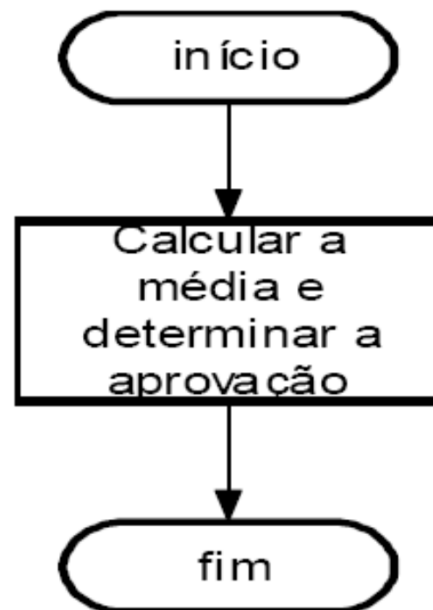
- Exemplo:

Uma escola qualquer, onde o cálculo da média é realizado por quatro notas bimestrais que determinam a aprovação ou reprovação dos seus alunos.

**Obs.:** Considerar que o valor da média deve ser maior ou igual a 7 para que haja aprovação.

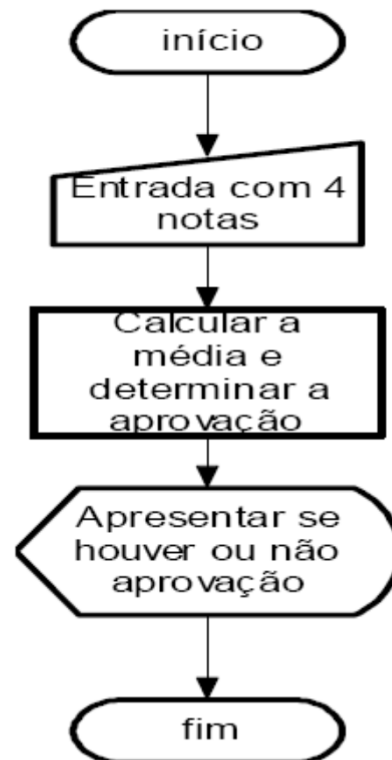
# Princípios de Resolução de Problemas

- A primeira etapa se inicia e termina com um terminador e existe apenas um processamento que indica a idéia geral do problema:



# Princípios de Resolução de Problemas

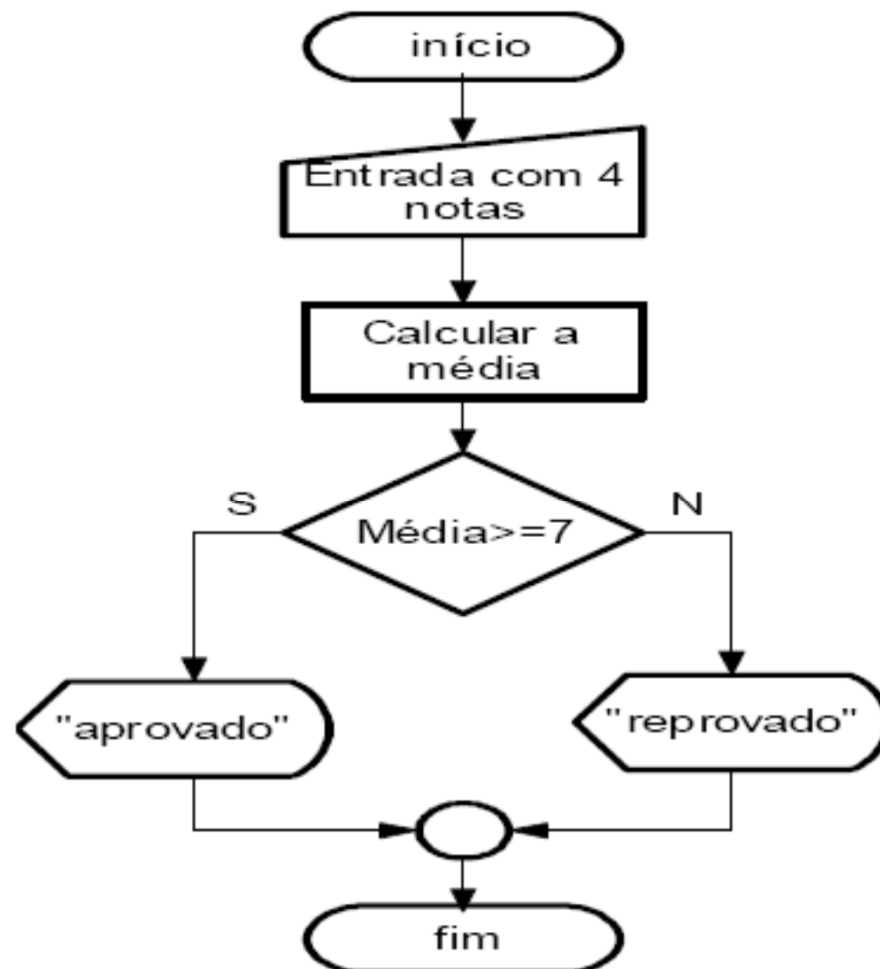
- O segundo detalhamento está no que se refere a entrada e saída dos dados do problema:



# Princípios de Resolução de Problemas

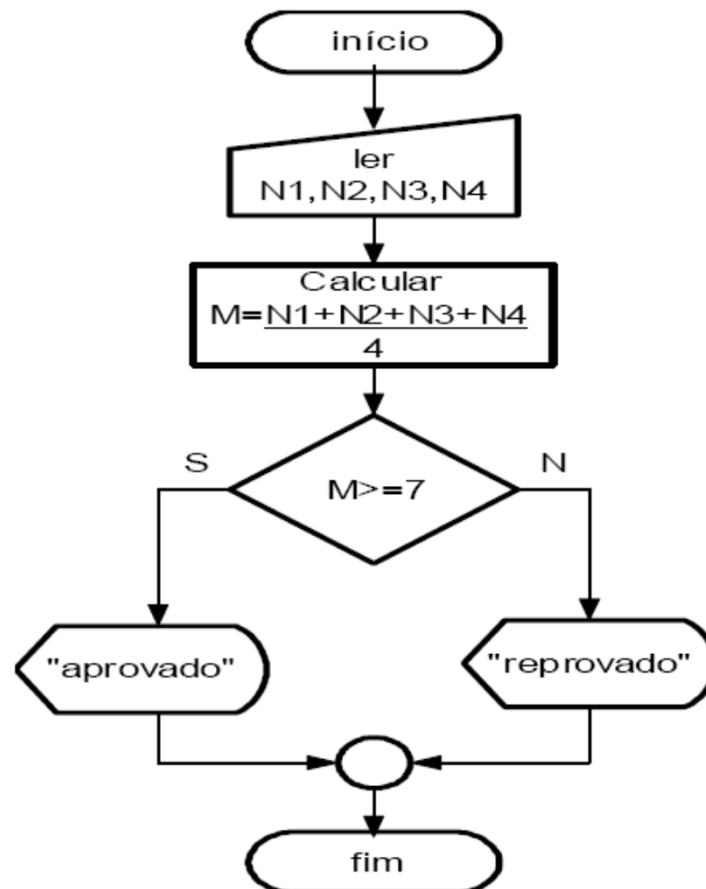
- A terceira etapa consiste em trabalhar o termo “determinar a aprovação”.
- Para ser possível determinar algo é necessário estabelecer uma condição.
- Esta condição decide sobre o resultado da aprovação.

# Princípios de Resolução de Problemas



# Princípios de Resolução de Problemas

- Esta terceira etapa deve ser aperfeiçoada para trabalhar com variáveis:



# Princípios de Resolução de Problemas

- A quarta fase consiste em escrever o diagrama de bloco na forma de pseudocódigo.

```
Programa média
Var
    Resultado : caractere
    n1,n2,n3,n4,soma,media : real
Inicio
    Leia(n1,n2,n3,n4)
    Soma<-n1+n2+n3+n4
    Media <-soma/4
    Se (media >=7) então
        Resultado <-“APROVADO”
    Senão
        Resultado<- “REPROVADO”
    Fim_se
    Escreva(Resultado)
fim
```



# Formalizações:

## Estrutura de um Pseudocódigo

```
Programa média } Cabeçalho do programa
Var
    Resultado : caractere
    n1,n2,n3,n4,soma,media : real
Inicio
    Leia(n1,n2,n3,n4)
    Soma<-n1+n2+n3+n4
    Media <-soma/4
    Se (media >=7) então
        Resultado <-“APROVADO”
    Senão
        Resultado<- “REPROVADO”
    Fim_se
    Escreva(Resultado)
fim
```

Área de declarações

Corpo do programa

# Formalizações:

## O que são Variáveis?

- Em programação, variáveis são regiões da memória do computador previamente identificadas, que têm por finalidade armazenar informações (dados) de um programa temporariamente.

# Formalizações:

## Nomes de Variáveis

- O nome de uma variável é utilizado para sua identificação e posterior uso dentro de um programa. Sendo assim, é necessário estabelecer algumas regras de utilização:
  - Poderão ser atribuídos com um ou mais caracteres;
  - O primeiro caractere sempre deverá ser uma letra;
  - Não poderá possuir espaços em branco;
  - Não poderá ser uma instrução de programa;
  - Não poderão ser utilizados outros caracteres a não ser letras e números.
  - Obs.: o caracter “\_” Underline é considerado como uma letra.

# Formalizações:

## Tipos de Dados

- Definem a natureza do dado, as operações que podem ser realizadas com o dado, e o espaço a ser ocupado na memória:
  - **Tipo Inteiro:** dado numérico positivo ou negativo, excluindo-se qualquer fracionário. Ex.: 2,0,-57;
  - **Tipo Real:** dado numérico positivo, negativo e números fracionários. Ex.: 35,0,-56,1.2,-45.897
  - **Tipo Character:** sequência contendo letras, números e símbolos especiais. Uma sequência de caracteres deve ser indicada entre aspas(""). Ex.: "PROGRAMAÇÃO","Rua Alfa"," ", "98".
  - **Tipo Lógico ou Booleano:** dado com valor verdadeiro ou falso. Ex.: Falso, Verdadeiro.

# Formalizações:

## Constantes

- Tem-se como definição de constante tudo aquilo que é fixo ou estável.
- O valor de uma **constante** não se altera após sua definição.
- Por exemplo, o valor 1.23 da fórmula a seguir pode ser uma constante:

RESULTADO<-ENTRADA\*1.23

# Formalizações:

## Operadores Aritméticos

Operador	Operação	Prioridade Matemática
**	Exponenciação	1
-	Inversão de sinal	1
*	Multiplicação	2
/	Divisão	2
+	Soma	3
-	Subtração	3

# Formalizações:

## Operadores Relacionais

Operador	Operação
<	Menor
>	Maior
<=	Menor ou Igual
>=	Maior ou Igual
==	Igual
!=	Diferente

# Formalizações:

## Operadores Lógicos

Operador	Operação	Prioridade
!	NOT	1
E	AND	2
OU	OR	3



# Formalizações:

## Operadores Lógicos

Operador E		
verdadeiro	verdadeiro	verdadeiro
verdadeiro	falso	falso
falso	verdadeiro	falso
falso	falso	falso

Operador OU		
verdadeiro	verdadeiro	verdadeiro
verdadeiro	falso	verdadeiro
falso	verdadeiro	verdadeiro
falso	falso	falso

Operador !	
verdadeiro	falso
falso	verdadeiro

# Formalizações:

## Operadores de Atribuição e Comentário

- **Operador de Atribuição:** usado para atribuir um valor a uma variável.

<-

- **Operador de Comentário:** usado para inserir explicações sobre o código escrito.


// : Comentário de linha

/\* \*/ : Comentário de bloco

# Formalizações:

## Precedência entre os Operadores

- Ordem em que os operadores são avaliados:



<b>Maior Precedência</b>
NOT, -(inversão)
*, /
+, -
Operadores relacionais
E
OU
<-
<b>Menor Precedência</b>

# Formalizações:

## Entrada e Saída de Dados

- Entrada de dados

LEIA(*variável*)

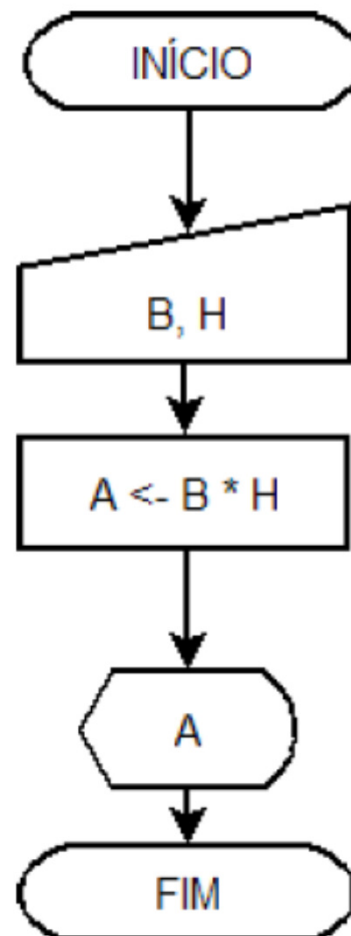
- Saída de dados

ESCREVA(*mensagem e/ou variável*)

# Problema 1: Área de um Retângulo

- Dadas a base e a altura de um retângulo, calcule e mostre na tela a área desta figura.

# Problema 1: Área de um Retângulo



# Problema 1: Área de um Retângulo

```
programa AREA_RETANGULO

var
    B, H: inteiro
    A: inteiro
início
    leia B, H
    A ← B * H
    escreva A
fim
```

# Dúvidas?

