

# Introdução a algoritmos

**Prof.:** Jesus

# Introdução a algoritmos

- Algoritmo é uma sequência de passos que visa a atingir um objetivo.
- Algoritmos são usados como meio para chegar a uma solução de um problema.
- Na programação, algoritmos devem seguir regras básicas para serem compatíveis com as linguagens de programação

# Propriedades de um algoritmo

- Possui um número finito de passos
- Cada passo é precisamente definido, sem possíveis ambiguidades
- Possui zero ou mais entradas tomadas de conjuntos bem definidos
- Possui uma ou mais saídas
- Existe uma condição de fim sempre atingida para quaisquer entradas e num tempo finito.

# Elaboração de um algoritmo

- Definir ações claras e específicas
- Partir de um estado inicial para um estado final em período de tempo finito
- Fixa um padrão, um caminho a ser seguido para se chegar a resolução de um problema
- Sempre que for executado, levará a solução do problema

# Por que usar algoritmo?

- Representa de forma mais eficiente o raciocínio envolvido na programação
- Abstrair detalhes computacionais que serão acrescentados posteriormente
  - Construção de algoritmos
- Algoritmos podem ser traduzidos para qualquer linguagem de programação
  - Codificação

# Exemplo de algoritmo

- Algoritmos são comuns no nosso cotidiano
- Podem ser descritos usando português coloquial
- Podem ser utilizados para descrever a resolução de problemas simples
  - Troca de uma lâmpada

# Troca de lâmpada

- Pegar uma escada;
- Posicionar a escada embaixo da lâmpada;
- Buscar um lâmpada nova;
- Subir na escada;
- Retirar a lâmpada velha;
- Colocar a lâmpada nova.

# Troca de lâmpada

- Neste algoritmo foram definidas **sequência** de ações que poderia ser seguido por qualquer pessoa
- Sequenciação: definir qual a primeira ação a ser executada e qual ação vem a seguir
- Sequencia é linear, ou seja, de cima para baixo
- **E se a lâmpada não estivesse queimada?**
  - A lâmpada seria trocada também



# Troca de lâmpada com teste

- Pegar uma escada;
- Posicionar a escada embaixo da lâmpada;
- Buscar um lâmpada nova;
- Acionar o interruptor;
- Se a lâmpada não acender, então
  - Subir na escada;
  - Retirar a lâmpada queimada;
  - Colocar a lâmpada nova.

# Troca de lâmpada com teste

- Nesta nova versão do algoritmo da lâmpada foi feito um teste para verificar se a lâmpada estava queimada
- Assim se a lâmpada não estiver queimada, não haverá uma troca **desnecessária** da lâmpada
- Foi incluído neste algoritmo um **teste seletivo**, para determinar quais ações devem ser feitas dependendo do resultado de uma condição
  - Uma condição pode ser **verdadeira ou falsa**

# Troca de lâmpada com teste

- Acionar o interruptor;
- Se a lâmpada não acender, então
  - Pegar uma escada;
  - Posicionar a escada embaixo da lâmpada;
  - Buscar um lâmpada nova;
  - Subir na escada;
  - Retirar a lâmpada queimada;
  - Colocar a lâmpada nova.

# Algoritmo da troca de lâmpada

- Existem muitas formas de resolver o mesmo problema
- O bom senso e a prática é que definirão a solução mais adequada
- E se lâmpada nova não funcionar?

# Algoritmo da troca de lâmpada com repetição

- Podemos alterar o algoritmo de forma que, após executada a ação “colocar outra lâmpada”, voltemos a executar o teste “se a lâmpada não acender”
- Usando a repetição, fazemos com essas ações sejam executadas quantas vezes for necessário sem ter reescrevê-las

# Algoritmo da troca de lâmpada com repetição

- Acionar o interruptor;
- Se a lâmpada não acender, então
  - Pegar uma escada;
  - Posicionar a escada embaixo da lâmpada;
  - Buscar um lâmpada nova;
  - Subir na escada;
  - Retirar a lâmpada queimada;
  - Colocar a lâmpada nova.
  - Enquanto a lâmpada não acender
    - Retirar a lâmpada queimada;
    - Colocar uma lâmpada nova;

# Formas de representação de algoritmos

- Descrição Narrativa
- Fluxograma convencional
- Diagramas de Chapin
- Pseudo-código (ou Pseudo-linguagem)

# Descrição Narrativa

- Nessa representação os algoritmos são expressos como linguagem natural

## Classificação do aluno

Obter a Nota do aluno

Se a nota for inferior a 5 : Escrever "Mau"

Se a nota estiver entre 5 e 10: Escrever "Medíocre"

Se a nota estiver entre 10 e 15: Escrever "Suficiente "

Se a nota estiver entre 15 e 20: Escrever "Bom"

Se não, escrever Nota inválida.

## Cálculo da média de um aluno

Obter as suas duas notas de prova

Calcular a média aritmética

Se a média for maior que 9.5

O aluno aprovou

Senão, o aluno reprovou



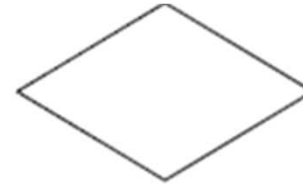
# Fluxograma

- É uma representação gráfica de algoritmos.
- •As formas geométricas representam instruções, comandos, etc.
- •Representação intermédia entre a forma narrativa e o pseudo-código.

# Fluxograma



Início ou fim do fluxograma



Decisão



Entrada de dados ou  
Atribuição de valores

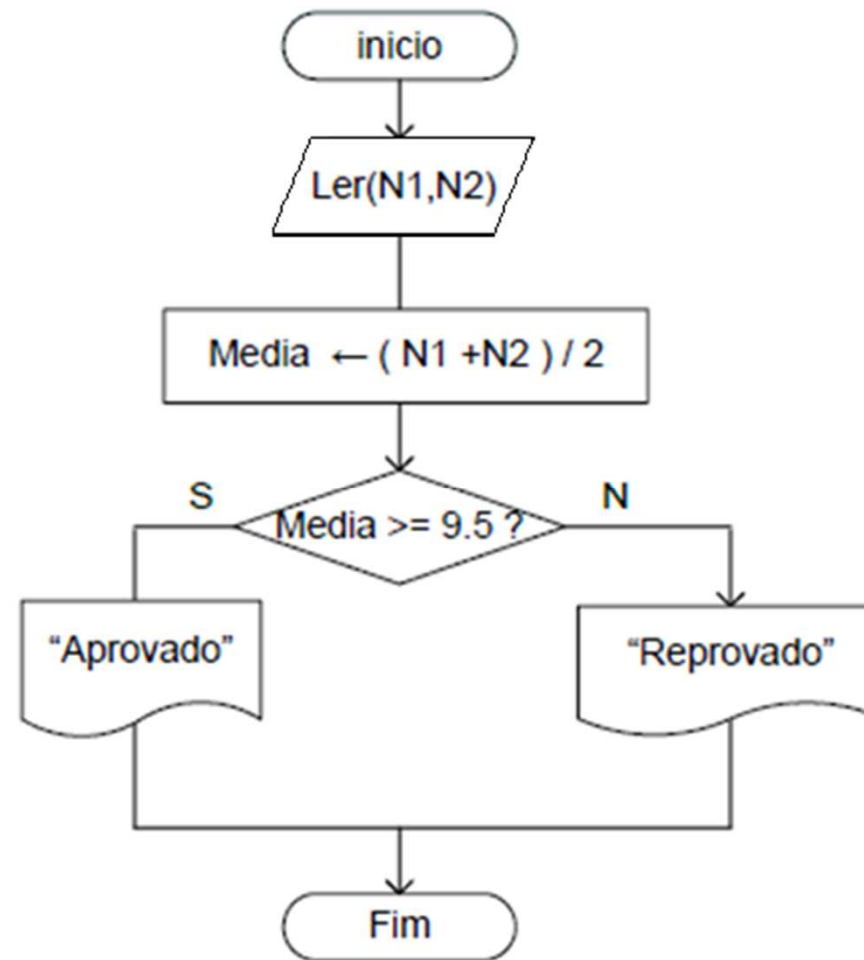


Atribuição de valores,  
processamento, cálculos, etc

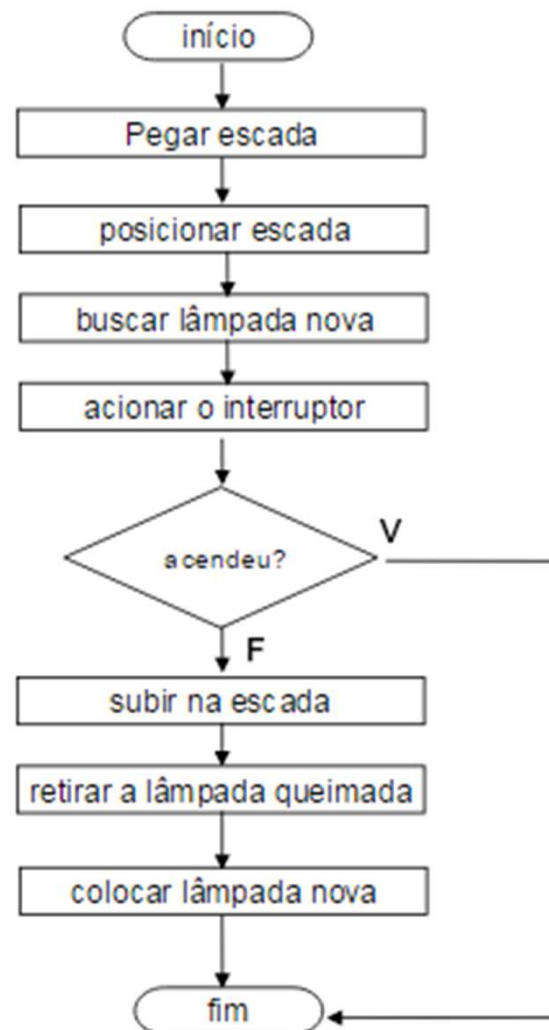


Operação de saída de dados

# Fluxograma



# Fluxograma



# Diagramas de Chapin

- Substituir o fluxograma tradicional por um diagrama que apresenta uma visão hierárquica e estruturada da lógica do programa.

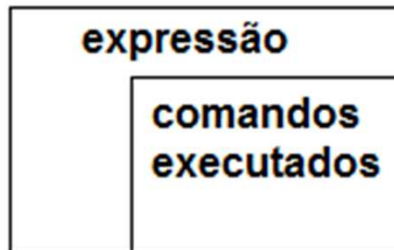
## Diagrama de Chapin - (Nassin-Shneiderman)



Início, fim, processamento (leitura, escrita, atribuição).

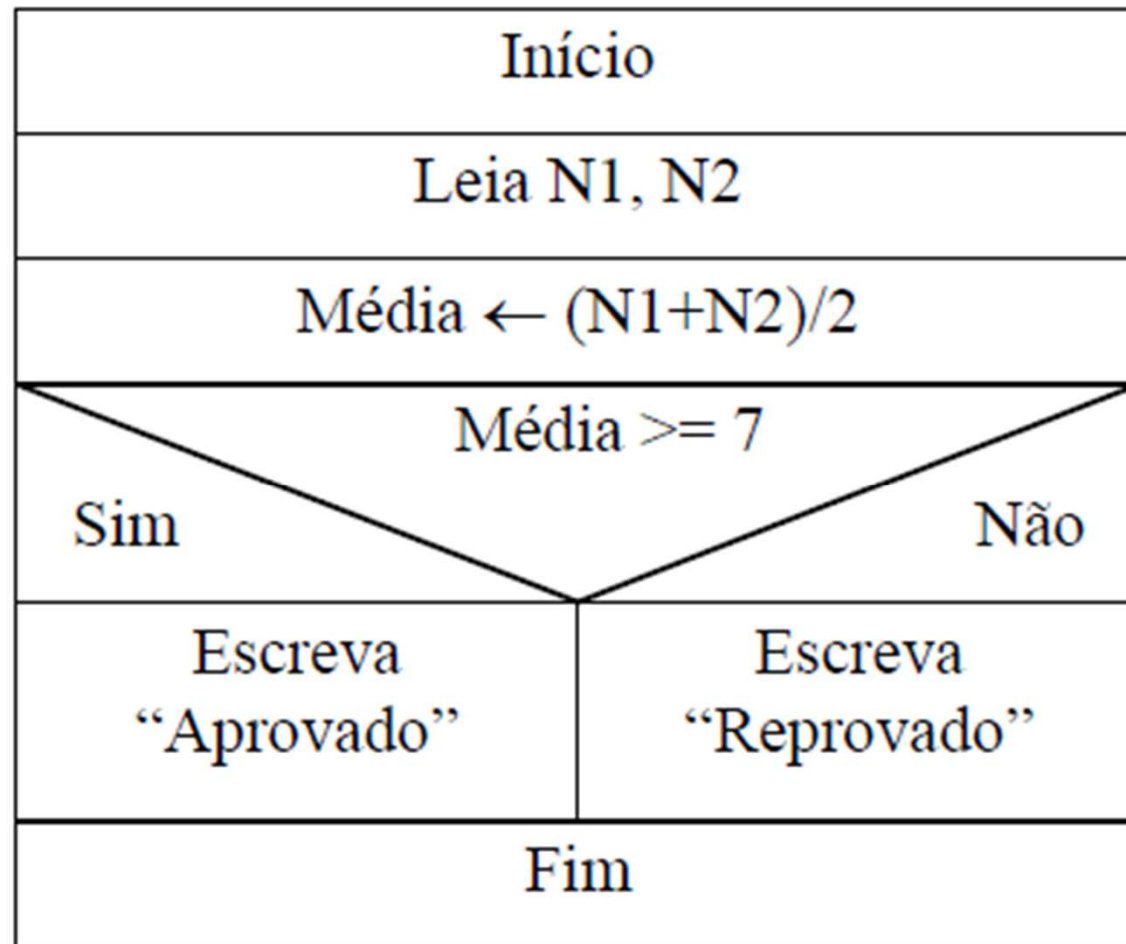


Decisão (se e senão)

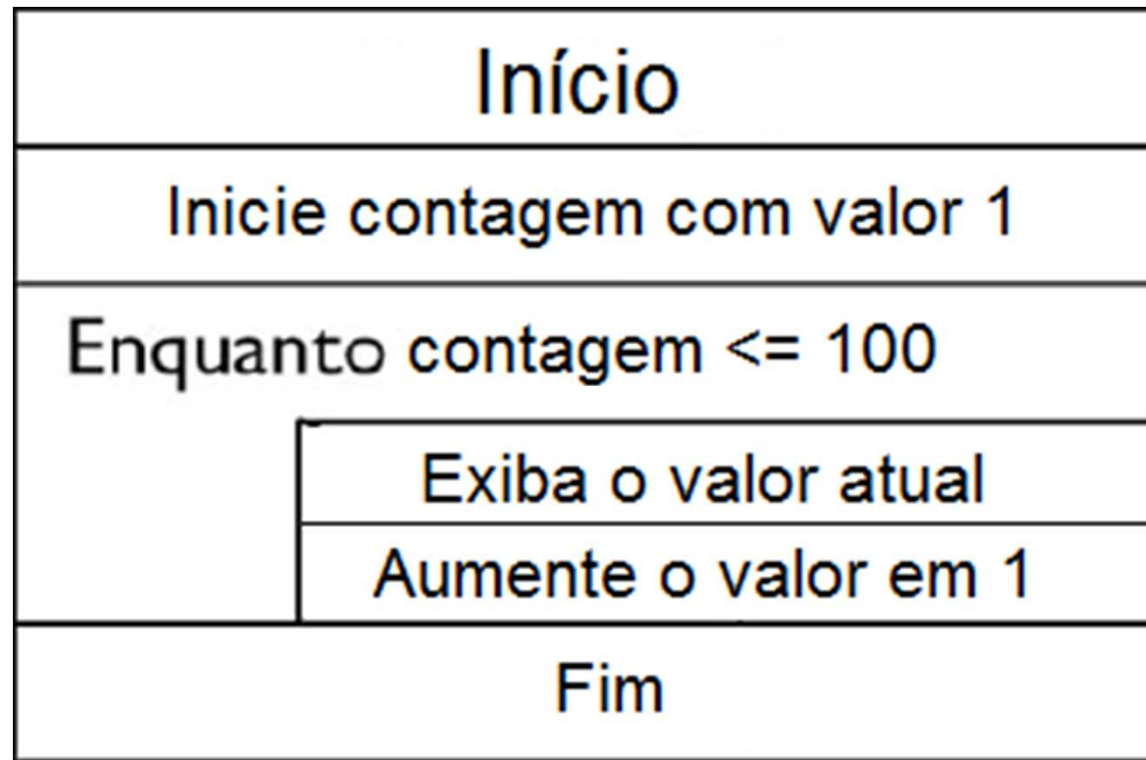


Repetição com teste no início (enquanto)

# Diagramas de Chapin



# Diagramas de Chapin



# Pseudo-Código

- Forma de representação de algoritmos rica em detalhes
- Assemelha-se bastante da forma como os programas são escritos

**prog** <nome\_do\_programa>

Variáveis

{Declaração das variáveis}

{Inicializar as variáveis}

{Entrada de dados}

{Cálculos/Processamento}

{Saída da informação}

**fimprog**

**prog** Calculo\_Media

**real** N1, N2, MEDIA;

**imprima** "Digite a nota N1:\n";

**leia** N1;

**imprima** "Digite a nota N2:\n";

**leia** N2;

MEDIA <- (N1+N2) / 2;

**se** (MEDIA >= 5.0)

{imprima "Aprovado.\n";}

**senão**

{imprima "Reprovado.\n";}

**fimprog**