
GFM015 – Introdução à Computação

Algoritmos

Ilmério Reis da Silva

ilmerio@facom.ufu.br

www.facom.ufu.br/~ilmerio/ic

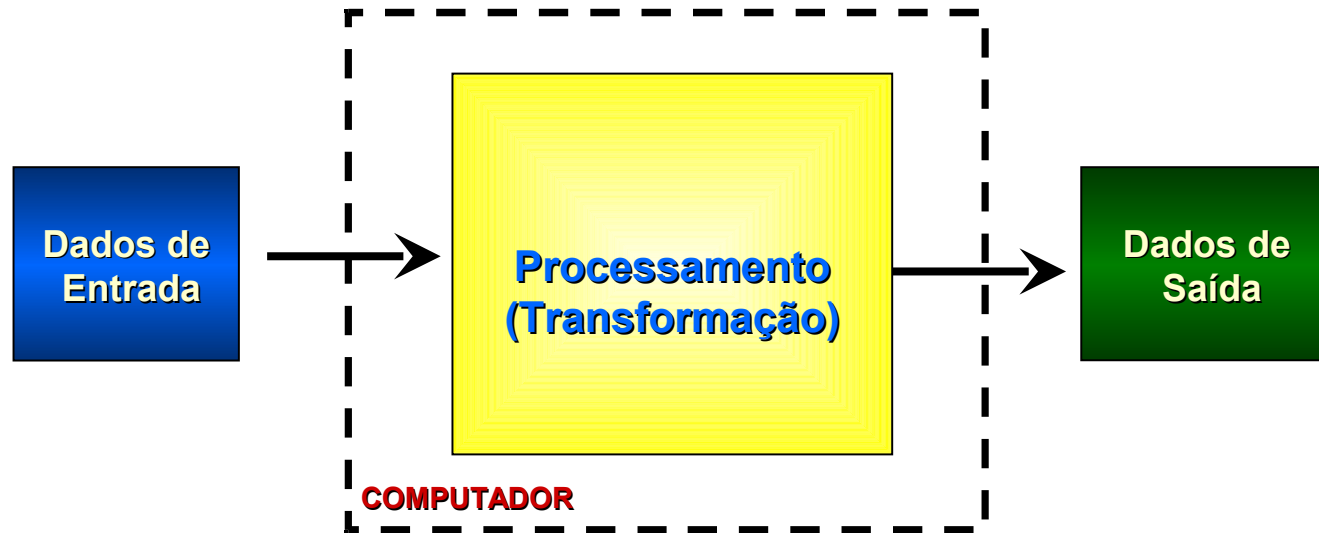
UFU/FACOM

Programa

1. Noções básicas sobre os componentes de micro computadores
2. Uso de Aplicativos
3. Algoritmos
 - 3.1 Abstração: representação do mundo real no computador
 - 3.2 Como escrever a solução de um problema para um computador: fluxograma, pseudocódigo
4. Fundamentos de programação
5. Estrutura de Dados
6. Modularização de programas

Abstração

Resolução de problemas por meio do computador



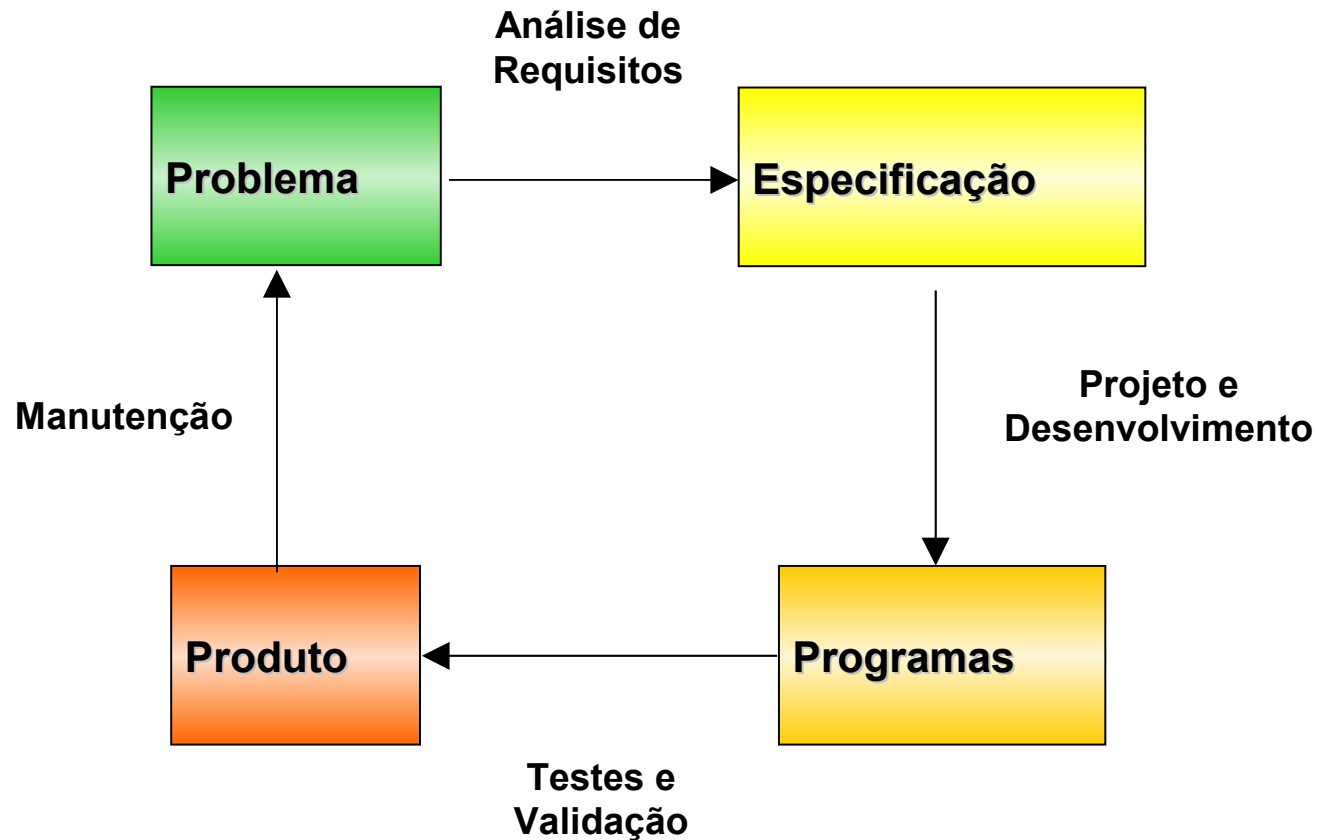
Abstração - Passos

Passos para resolução de problemas:

- Entendimento do Problema
- Criação de uma seqüência de operações para solução do problema
- **Execução desta seqüência**
- Verificação da adequação da solução

A **Execução** pode ser feita por um computador

Uma Abstração das Fases de Desenvolvimento



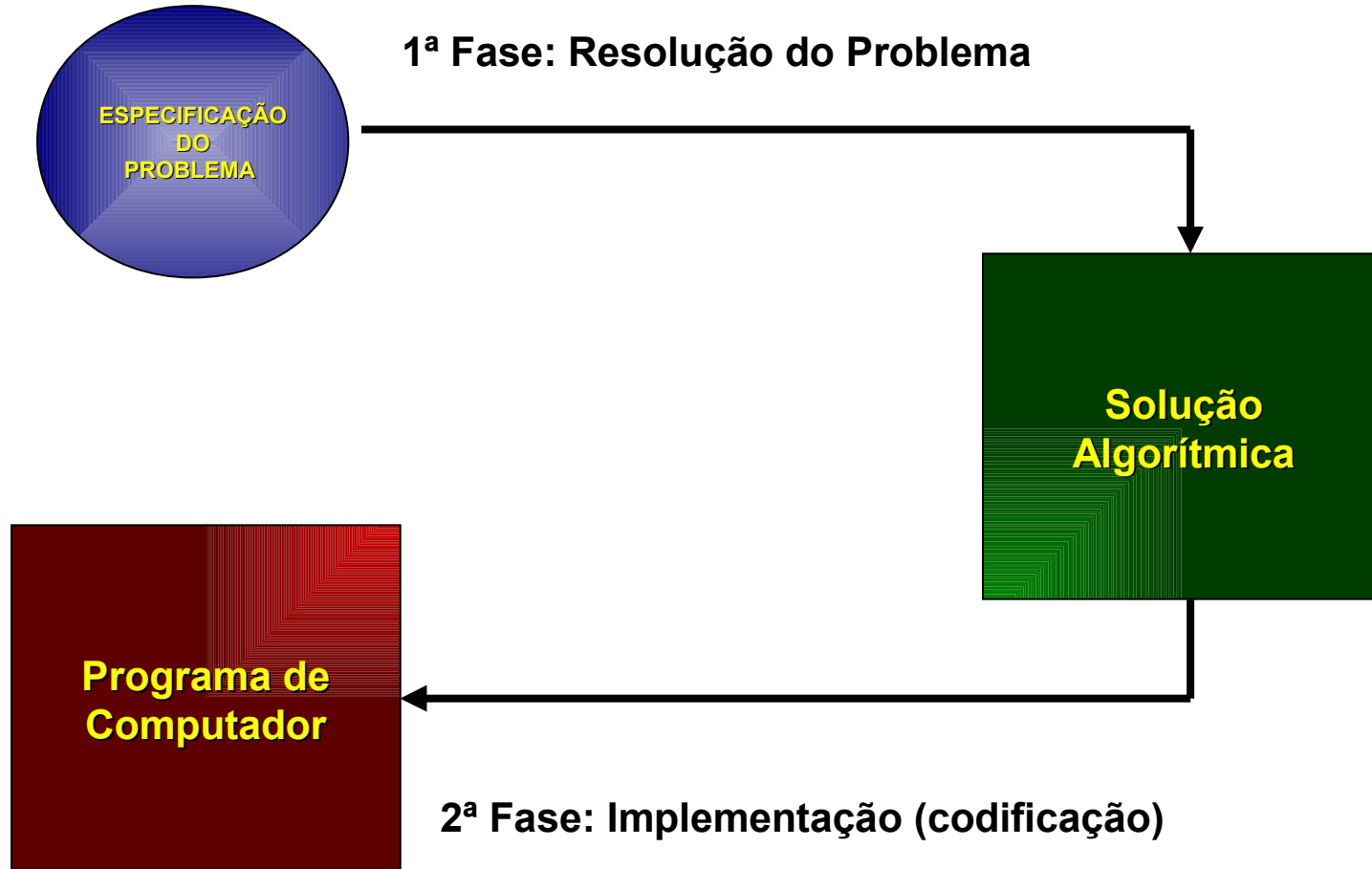
Análise e Especificação de Requisitos

Def. **Análise de Requisitos** é o levantamento e estudo das necessidades do usuário

Def. **Especificação de Requisitos** é a descrição das necessidades do usuário por meio da ação que deve ser executada pelo sistema.

Ex: Registrar as notas dos alunos;
Calcular a média final, etc.

Projeto e Desenvolvimento



Teste e validação

- Verificar se o sistema apresenta erros de execução
- Verificar se o sistema atende aos requisitos
- Validação junto ao usuário

Manutenção

- Correção de erros identificados no teste e validação
- Adaptação a novos requisitos

Algoritmos

Algoritmos - Definição

Em geral:

Def. Algoritmo: é uma seqüência finita de passos que deve ser seguida para a realização de uma tarefa.

Exemplo: Trocar um pneu furado

Em computação:

Def. Algoritmo é uma seqüência finita de instruções ou operações cuja execução, em tempo finito, resolve um problema computacional, qualquer que seja sua instância.

Exemplo: Somar três números

Algoritmos - Propriedades

- Cada passo representa uma ação bem definida, sem ambiguidade e “entendida” pelo executor
- A sequência de passos é ordenada
- O conjunto de passos é finito.

Algoritmos – Exemplo Geral – Executor Humano

Exemplo1 Trocar um pneu furado

Passo 1: Pegar o macaco e o estepe;

Passo 2: Levantar o carro usando o macaco;

Passo 3: Retirar o pneu furado;

Passo 4: Colocar o estepe em seu lugar;

Passo 5: Abaixar o carro;

Passo 6: Guardar o macaco e o pneu furado;

Algoritmos – Exemplo Computacional

Exemplo2: Somar três números

Passo 1: Ler os três números;

Passo 2: Somar os três números;

Passo 3: Mostrar o resultado obtido;

OBS: observe as operações de entrada e saída

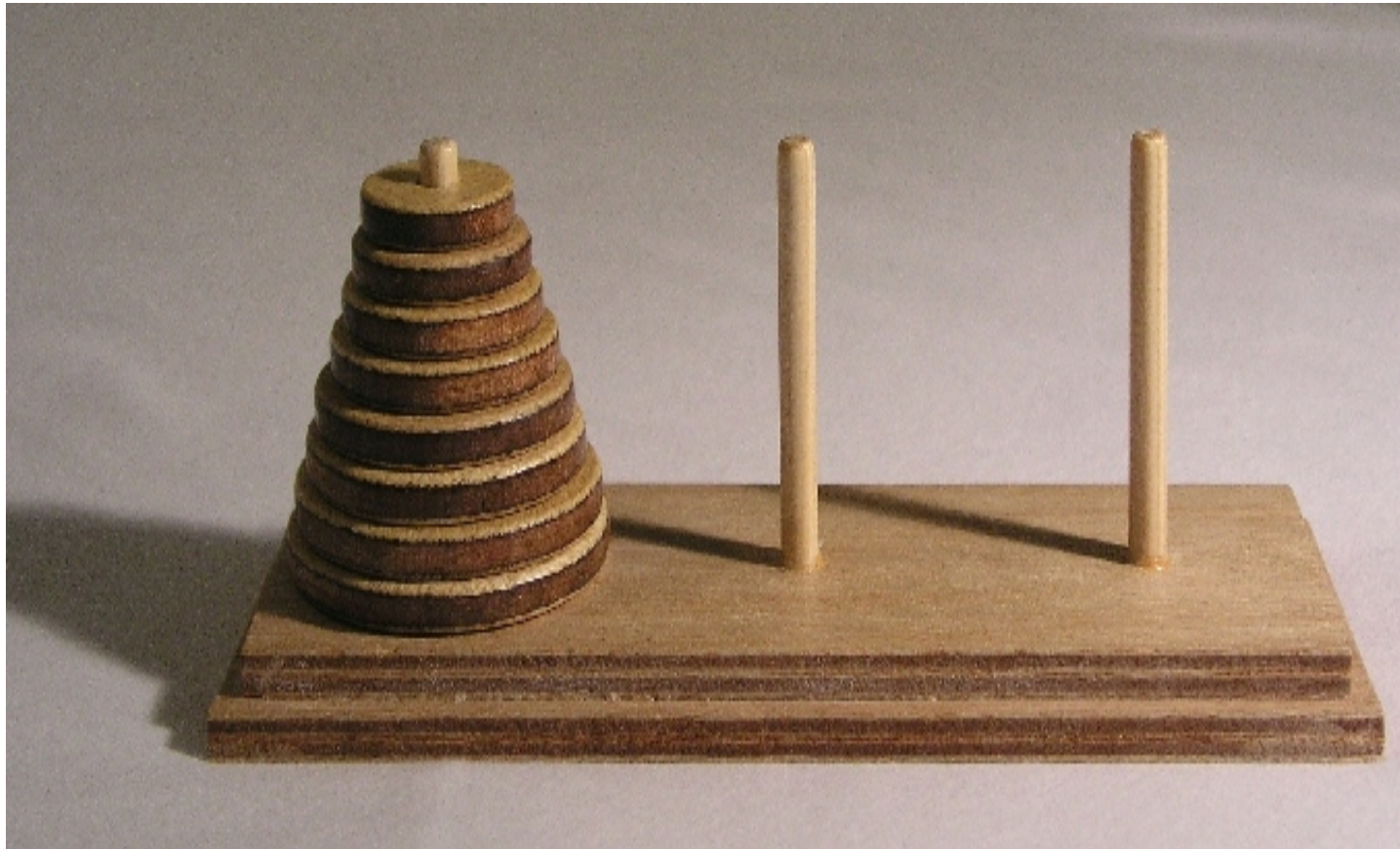
Outro Exemplo – Torre de Hanói

Dada uma base contendo três pinos, A, B e C, sendo que no Pino A são dispostos alguns discos uns sobre os outros, em ordem crescente de diâmetro, de cima para baixo,

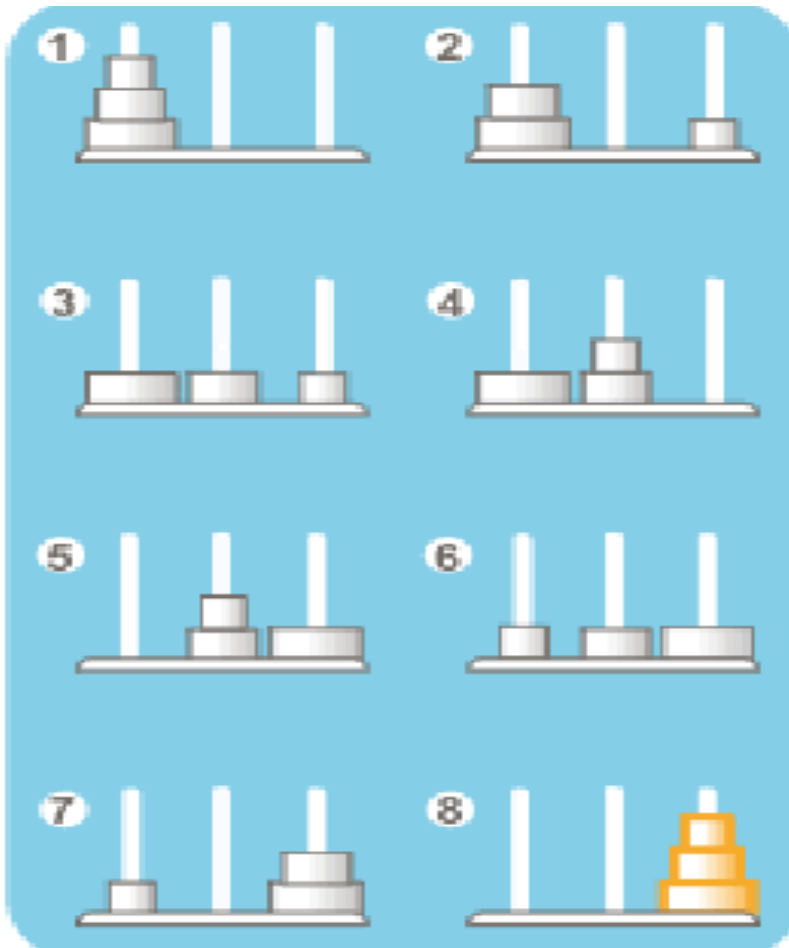
o problema

consiste em passar todos os discos para o Pino C, mantendo sempre a ordem crescente de diâmetro na disposição dos discos nos três pinos.

Torre de Hanói - Figura



Torre de Hanói – Solução para três discos



Algoritmo

- 1 Posição inicial;
- 2 Move o disco 1 para a Pino C;
- 3 Move o disco 2 para a Pino B;
- 4 Move o disco 1 para a Pino B;
- 5 Move o disco 3 para a Pino C;
- 6 Move o disco 1 para a Pino A;
- 7 Move o disco 2 para a Pino C;
- 8 Move o disco 1 para a Pino C;

Torre de Hanói – Curiosidades

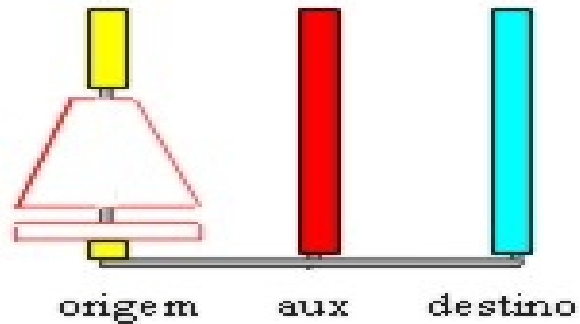
O número mínimo de "movimentos" para conseguir transferir todos os discos do primeiro pino para o terceiro é $2^n - 1$, sendo n o número de discos.

Logo:

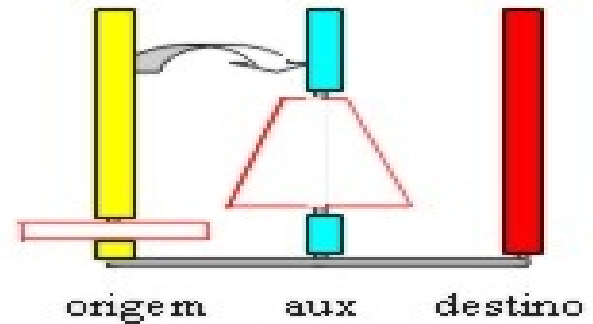
- 3 discos \Rightarrow 8 movimentos
- 4 discos \Rightarrow 15 movimentos
- 7 discos \Rightarrow 127 movimentos
- 15 discos \Rightarrow 32.767 movimentos
- 64 discos \Rightarrow 18.446.744.073.709.551.615 movimentos.

Entretanto, podemos desenvolver um algoritmo com poucos passos para solucionar esse problema.

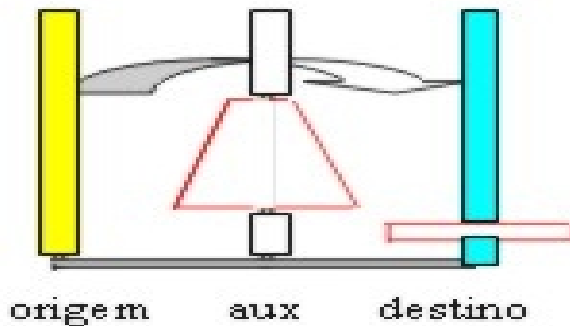
Torre de Hanói – Uma solução recursiva



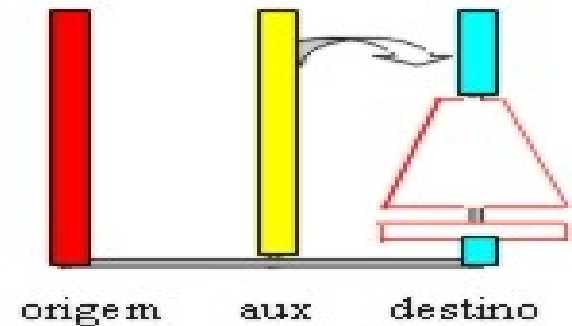
(a)



(b)



(c)



(d)

Solução da Torre de Hanoi com o disco de baixo e a torre de cima.

Torre de Hanói – Algoritmo Recursivo

```
módulo TRANSFERE (N, origem, destino, auxiliar)
{ transfere uma torre de N discos da origem para o destino,
  usando o pino auxiliar, se for necessário }
se N=1
  então move disco da origem para o destino
  senão início
    TRANSFERE (N-1, origem, auxiliar, destino)

    move disco de origem para o destino

    TRANSFERE (N-1, auxiliar, destino, origem)
  fim
```

Programa Principal:

```
se N>0
  então TRANSFERE (N, origem, destino, auxiliar)
fim
```

Exercícios algoritmos

Temos 3 recipientes de tamanhos distintos (8, 5 e 3 litros), sendo que o recipiente de 8 litros está totalmente cheio. Considerando que os recipientes não sejam graduados, deseja-se colocar 4 litros em dois recipientes.

Exercícios algoritmos

Um comerciante está transportando um lobo, um coelho e 500 kg de cenouras. Durante a viagem, ele se depara com um rio e um pequeno barco, no qual só é possível transportar um elemento por vez. Descreva quais serão as ações tomadas pelo comerciante para atravessar o rio, de modo que ele nunca deixe o lobo e o coelho ou o coelho e as cenouras sozinhos em uma das margens.

Características de Algoritmos Computacionais

- Abstrai-se de alguns detalhes da linguagem de programação, mas o Executor será um computador
- Logo, as instruções devem ser entendidas pelo computador
- Características
 - Operações de entrada e saída
 - Estruturas de dados
 - Estruturas de controle

Diretrizes para o Desenvolvimento de Algoritmos

- *Identificar o problema*
- *Identificar as entradas*
- *Identificar as saídas*
- *Definir os passos: Regras; Limitações; Ações*
- *Registrar a sequência de passos*
- *Testar por meio de execução manual*

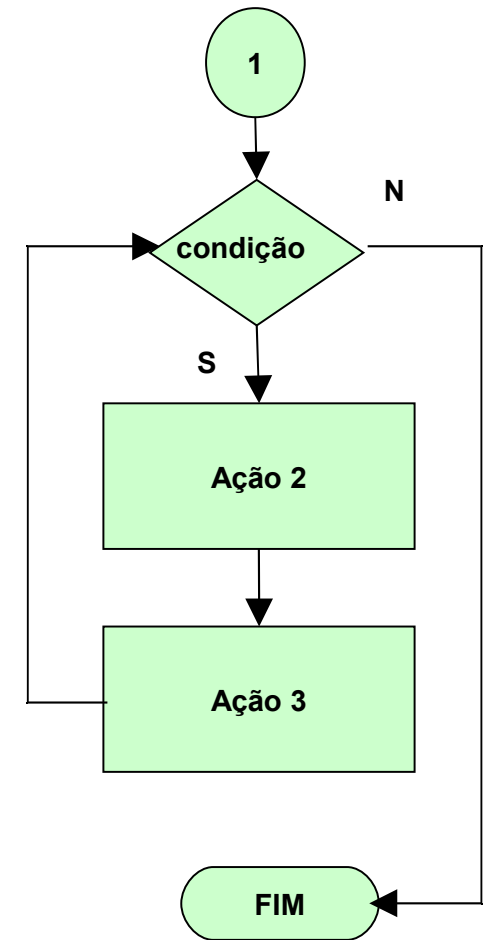
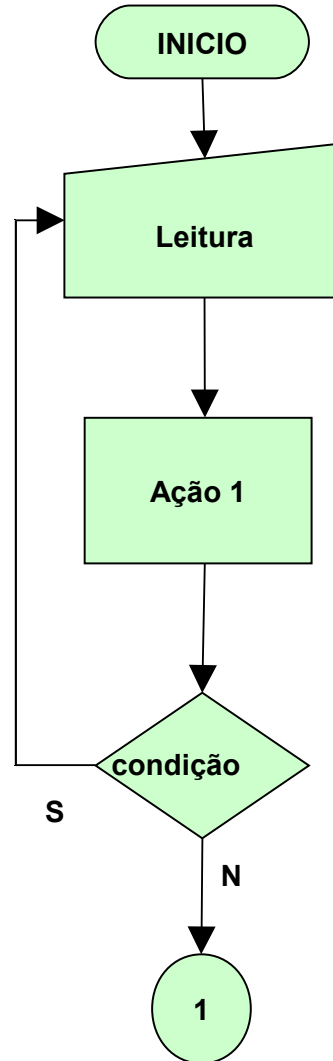
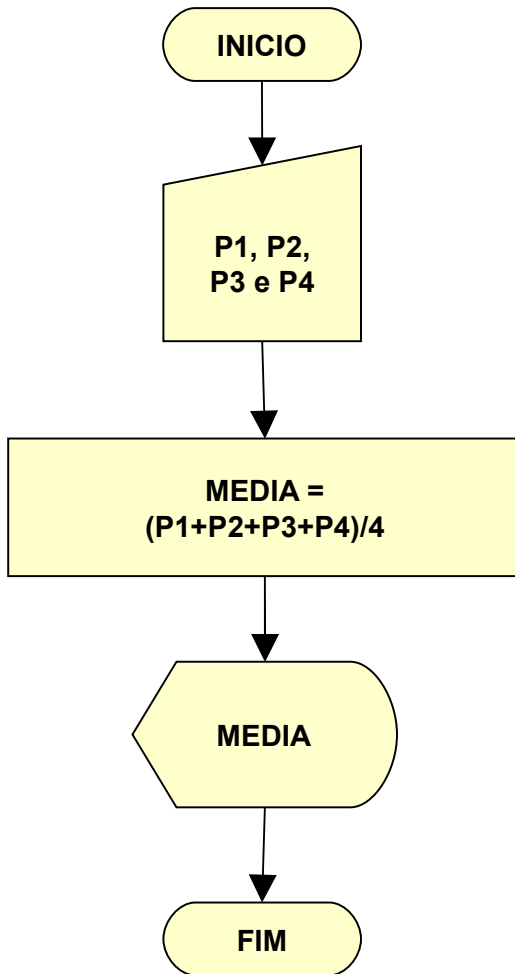
Exemplo: calcular a média final do aluno, sendo que o mesmo realizou quatro provas:

- *E: p_1, p_2, p_3, p_4 ;*
 - *S: media;*
 - *Sequência:*
 - *obter P_1, P_2, P_3, P_4 ;*
 - *somar P_1, P_2, P_3, P_4 e dividir a soma por 4;*
 - *apresentar o resultado da divisão*
-

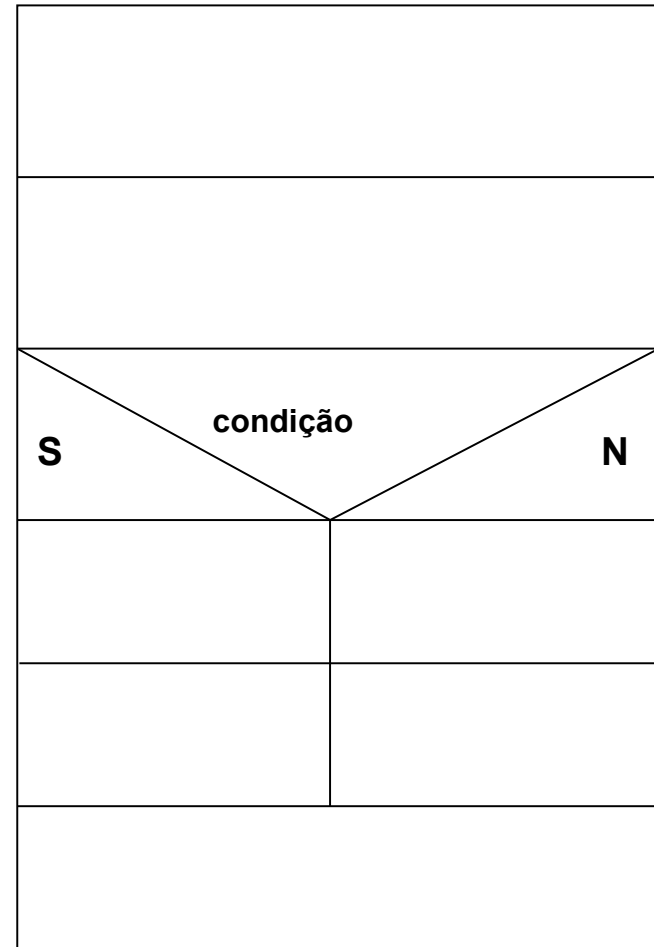
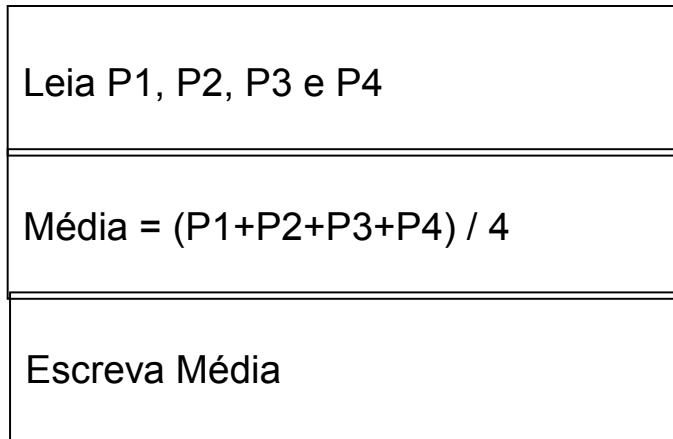
Formas de Representação de Algoritmos

- Narrativa em linguagem natural
 - Imprecisa
 - Dificulta a codificação
- Fluxograma: gráfico representando passos individuais por meio de objetos gráficos e fluxo de execução por meio de setas
- Diagramas de Chapin ou Nassi_Shneiderman: gráfico representando cada passo por uma caixa
- Pseudocódigo: linguagem específica que utiliza expressões pré-definidas para representar ações e controle, facilitando a codificação futura

Fluxogramas



Diagramas de Chapin ou Nassi_Shneiderman



Pseudocódigo

Características

- Texto estruturado por meio de regras
- Palavras-chave (escreva, se-então, etc.)
- Finalizador de comandos (;)
- Etc.

Obs: utilizaremos pseudocódigo para representar nossos algoritmos computacionais.

Exemplo de Pseudocódigo

```
algoritmo_Media_Final
  declare
    inteiro: P1, P2, P3, P4;
    real: media;
  inicio
    leia(P1, P2, P3, P4);
    Media ← (P1+P2+P3+P4)/4;
    escreva(media);
  fim
fim_algoritmo
```

Bibliografia

Disponível em:

http://www.facom.ufu.br/~ilmerio/ic/ic_s3_algoritmos.pdf

Material de apoio em:

http://www.facom.ufu.br/~ilmerio/ic/ic_introducao.pdf

FIM – Algoritmos
