

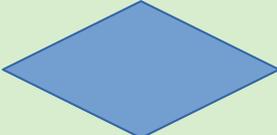
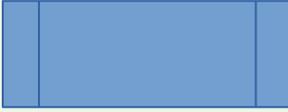
LÓGICA DE PROGRAMAÇÃO

PROF^a. M.Sc. JULIANA H Q BENACCHIO

Expressando a lógica em programação

- Ferramentas gráficas: diagramas e fluxogramas;
- Ferramentas textuais: pseudocódigos ou metalinguagens (Português estruturado)

Fluxograma

Símbolo	Descrição
	Terminal (início e fim do algoritmo)
	Entrada de dados (via teclado)
	Processamento de dados (cálculos)
	Saída de dados (via video)
	Tomada de decisão (condição)
	Execução de processo predefinido
	Ponto de conexão

Tipos de Dados Primitivos

- Os dados são elementos do mundo exterior, que representam dentro de um computador as informações manipuladas pelos seres humanos;
- Os dados a serem utilizados devem primeiramente ser abstraídos para serem processados;
- São classificados em três tipos básicos:
 - Numéricos, caracteres e lógicos.

- Os dados, inicialmente, precisam ser guardados em algum lugar do programa para serem manipulados;
- Cria-se então um espaço denominado variável, onde um dado pode permanecer pelo tempo que for necessário;
- Toda variável possui uma identificação (nome).
- Para usar o dado, primeiramente acessamos a variável através de seu nome.

Constantes

- Constante é tudo aquilo que é fixo, estável;
- Uma grandeza numérica fixa utilizada normalmente em uma expressão aritmética ou matemática, a qual define um valor que será inalterado na expressão, independentemente das variáveis envolvidas na operação a ser realizada.

Operadores Aritméticos

Operador	Descrição
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
<code>div</code>	Divisão inteira
<code>mod</code>	Resto da divisão
↑	Exponenciação

Expressões Matemáticas

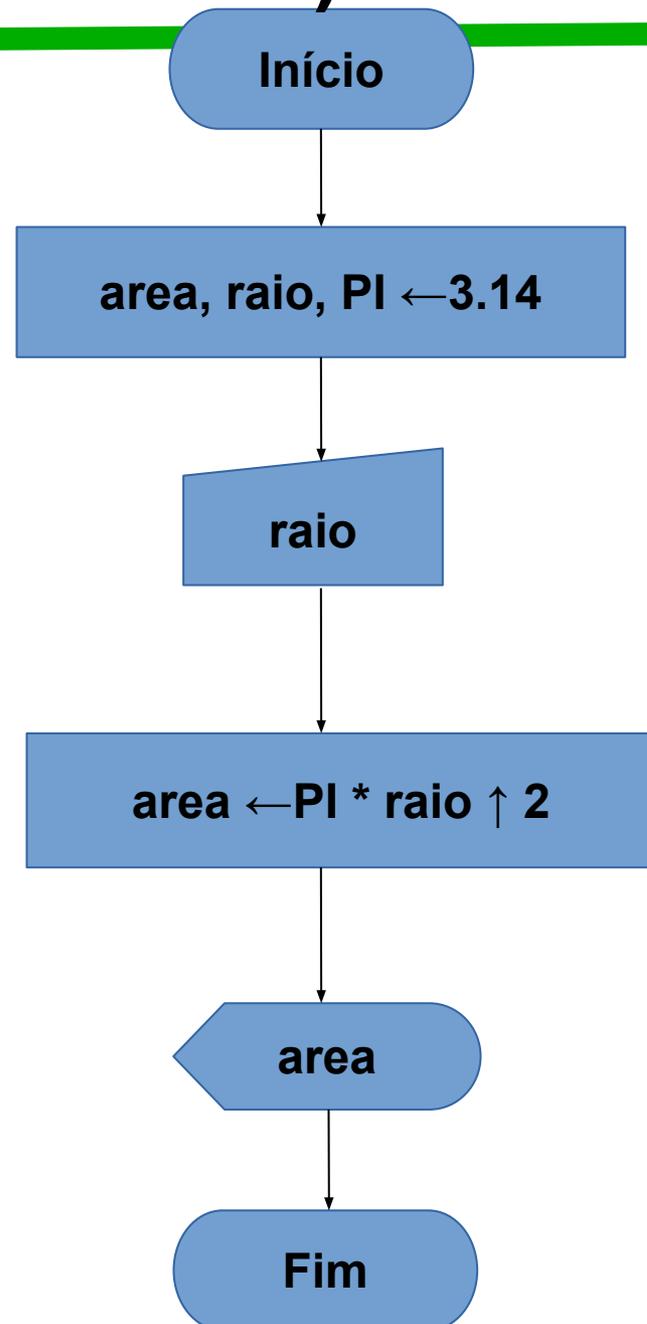
- Relacionamento existente entre variáveis e constantes numéricas com a utilização dos operadores aritméticos.
- Exemplo: Fórmula para calcular a área de um círculo.

$$\text{area} = \pi \cdot \text{raio}^2$$

- Em pseudocódigo ficaria assim:

```
area ← 3.14159 * raio ↑ 2
```

Em fluxograma (área de um círculo):



Exemplo 1

- Desenhe um fluxograma para calcular e exibir a área de um triângulo. O programa deve solicitar como dados de entrada o tamanho da base e a altura do triângulo.

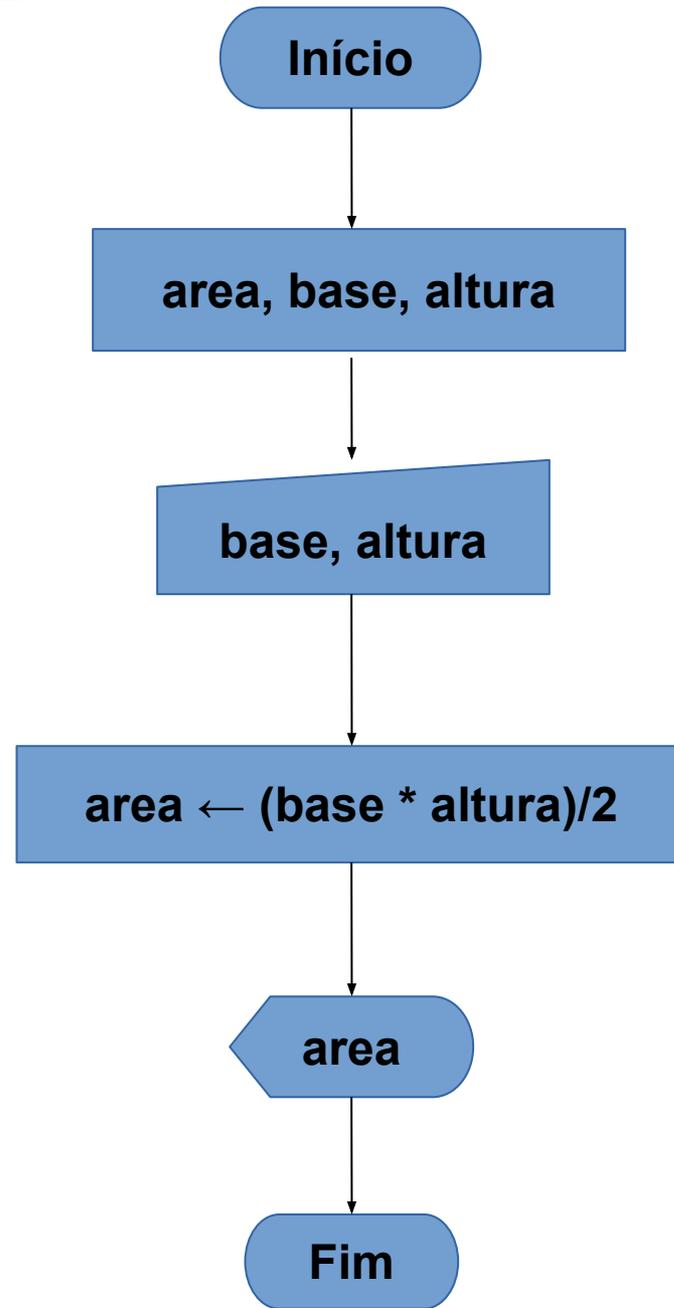
Exemplo 1

- Cálculo para definir a área de um triângulo:

$$\text{area} = \frac{\text{base} \cdot \text{altura}}{2}$$

$$\text{area} \leftarrow (\text{base} * \text{altura}) / 2$$

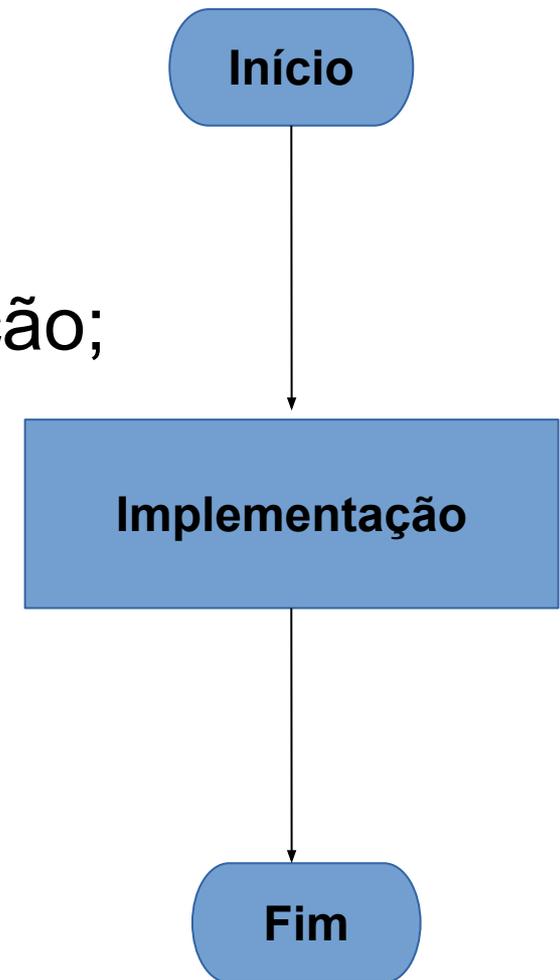
Resposta 1



Construindo um algoritmo

- Todo algoritmo deve ter:
 - Início;
 - Desenvolvimento; Implementação;
 - Fim;
 - Estrutura de bloco

```
inicio //início do bloco  
    //declaração das variáveis e constantes  
    //sequência de ações/comandos  
fim //fim do bloco
```



Declaração de variáveis

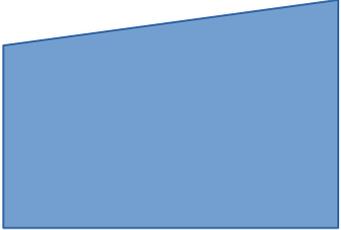
- No ambiente computacional, as informações variáveis são guardadas em dispositivos eletrônicos chamados de **memória**;
- Toda variável precisa de um nome e de um tipo;

```
inicio //início do bloco  
    //declaração das variáveis e constantes  
    //sequência de ações/comandos  
fim //fim do bloco
```

tipo pode ser:
inteiro
real
caractere
lógico

Entrada de Dados

- Para que um algoritmo possa receber dados de que necessita, é preciso adotar um comando de **entrada de dados**;
 - `leia (<identificador>) ;`
 - `leia (nome) ;`
 - `leia (nome, idade) ;`



```
início //início do bloco
    //declaração das variáveis e constantes
    tipo: identificador;
    //sequência de ações/comandos
    leia(identificador);
fim //fim do bloco
```

Saída de Dados

- Para que o algoritmo possa mostrar os dados que processou (calculou) ele deve fazer o uso de um **comando de saída**;

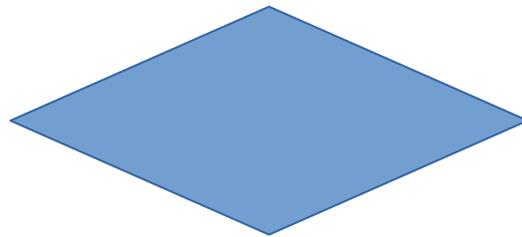
- `escreva (<identificador>, <expressão>);`
- `escreva (x)`
- `escreva ("Bom dia");`
- `escreva ("Bom dia", nome);`



```
inicio //início do bloco
//declaração das variáveis e constantes
tipo: identificador;
//sequência de ações/comandos
leia(identificador);
escreva(identificador);
fim //fim do bloco
```

Estruturas de Controle

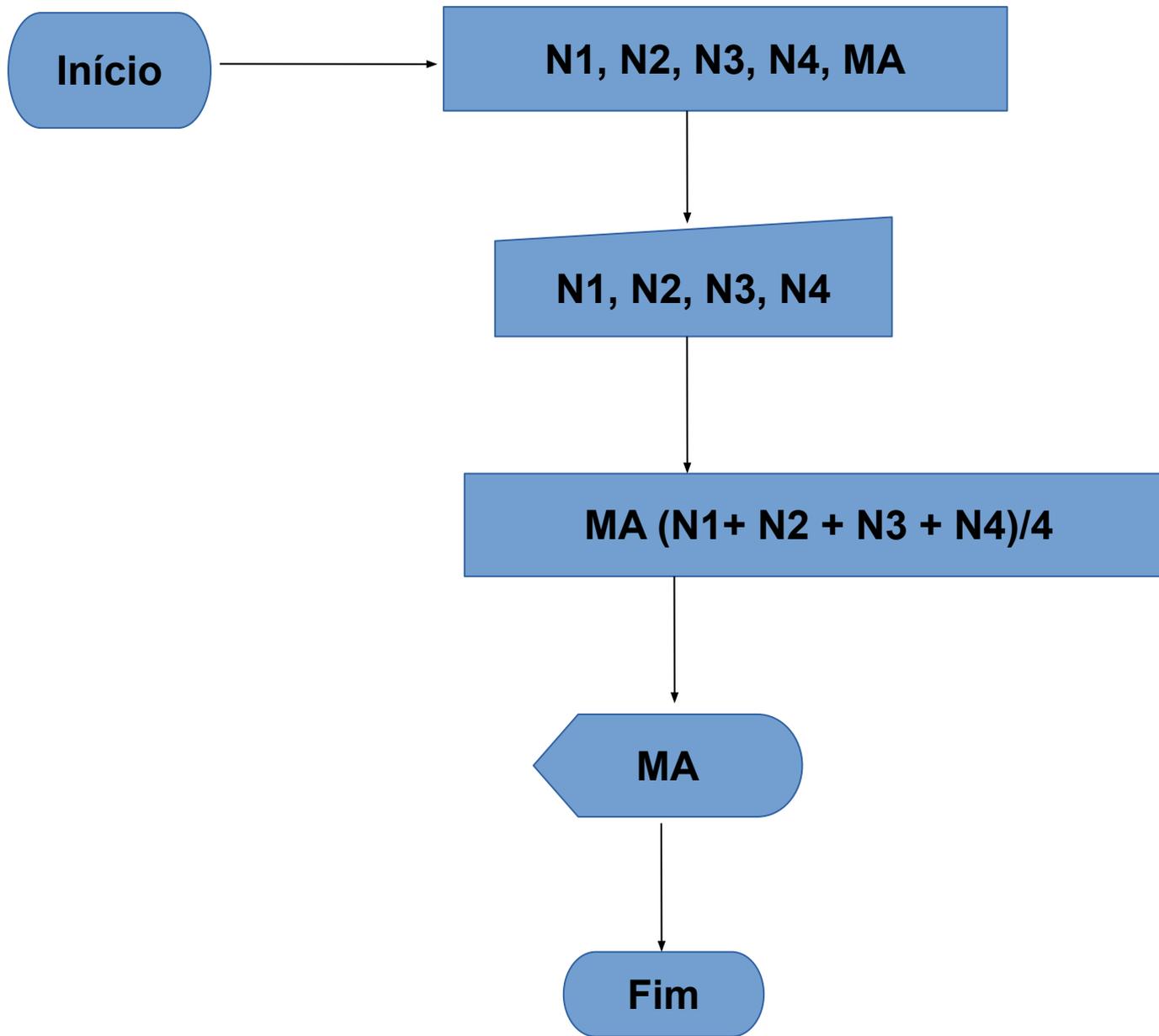
- Sequencial;
- Seleção ou Desvio Condicional;
- Repetição;



Estrutura Sequencial

- Quando um conjunto de ações primitivas será executado em uma sequência linear, de cima para baixo, da esquerda para a direita;
- Mesma ordem que foram escritas

Estrutura Visual - Sequencia



Estrutura Textual - Sequencia

inicio

//declaração de variáveis

real: N1, N2, N3, N4, MA;

//comando de entrada de dados

leia(N1, N2, N3, N4);

//processamento

MA $(N1 + N2 + N3 + N4)/4$;

//comando para saída de dados

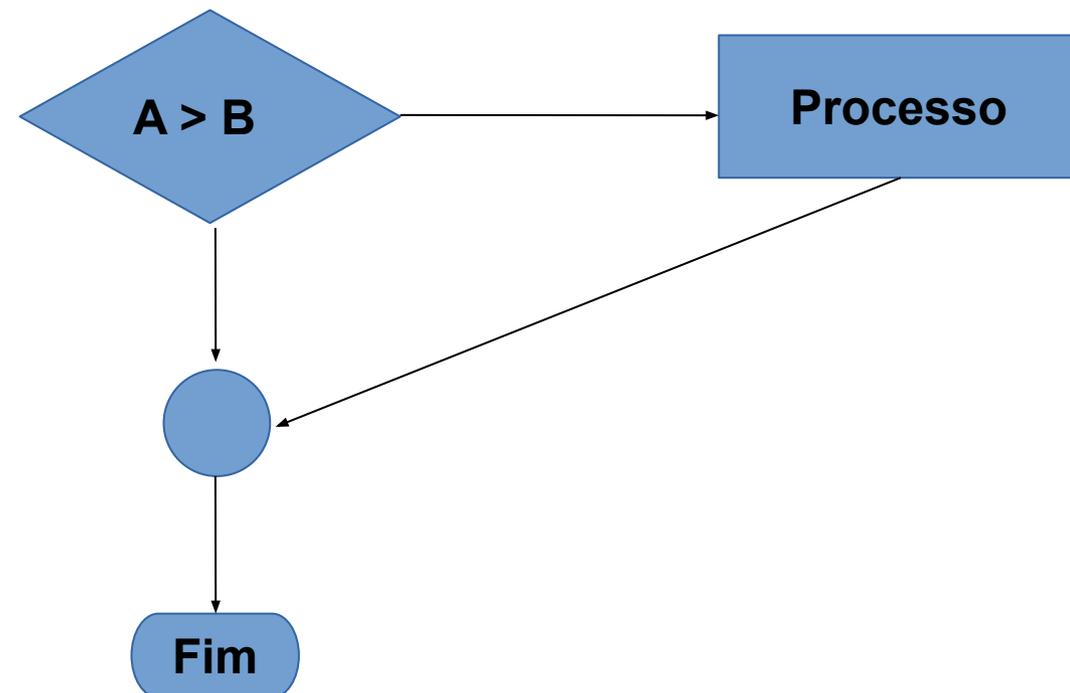
escreva(MA);

fim

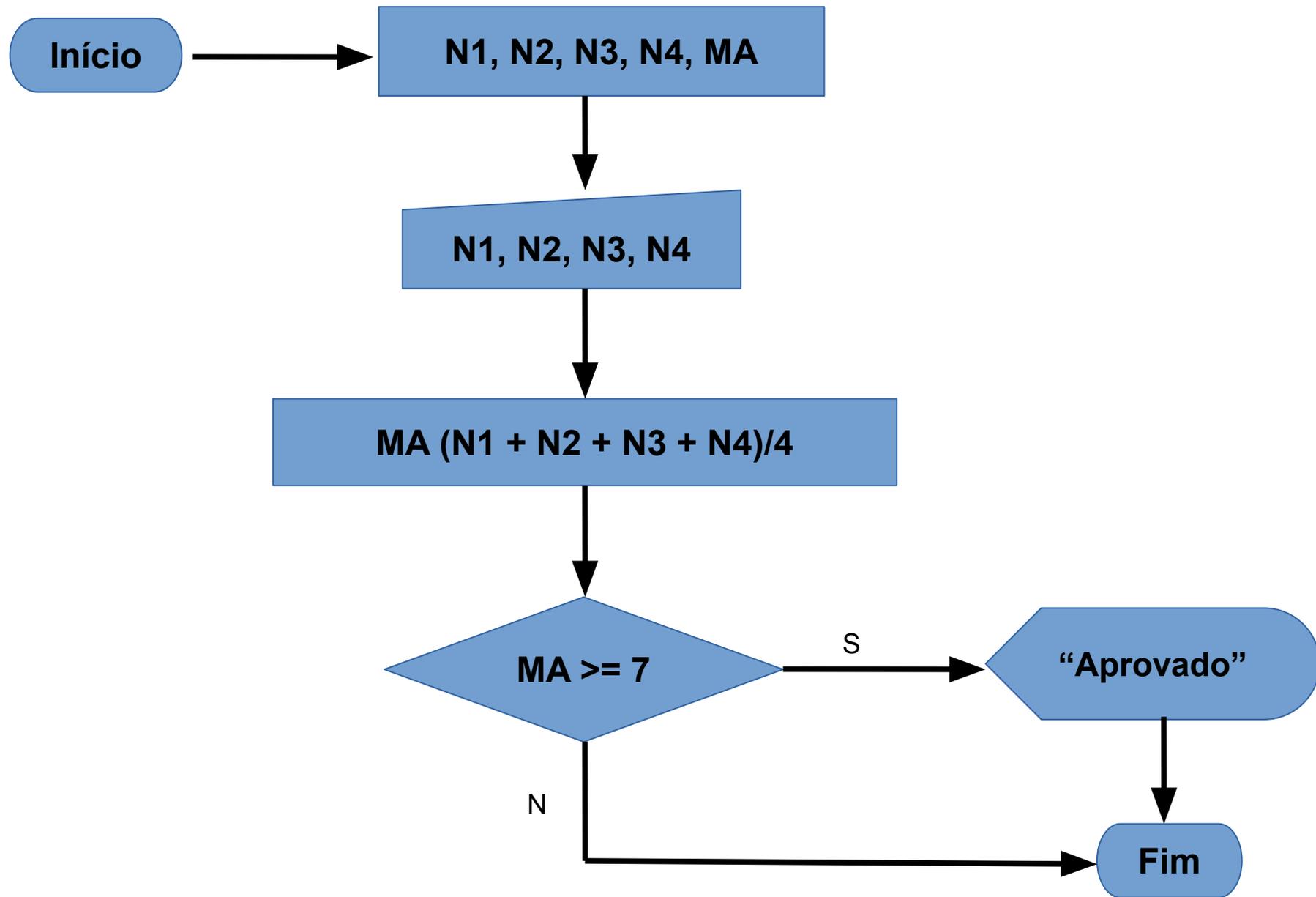
Estrutura de Seleção (Simple)

- Permite a escolha de um grupo de ações (bloco) a ser executado quando determinadas condições, representadas por expressões lógicas ou relacionais são ou não satisfeitas;

```
se <condição> então
    //comandos
fimse;
```



Estrutura Visual – Decisão Simples



Estrutura Textual – Decisão Simples

início

//declaração de variáveis

real: N1, N2, N3, N4, MA;

//comando de entrada de dados

leia(N1, N2, N3, N4);

//processamento

MA (N1 + N2 + N3 + N4)/4;

//comando para saída de dados

escreva(MA);

//desvio condicional

se (MA >= 7) **então**

escreva("Aprovado");

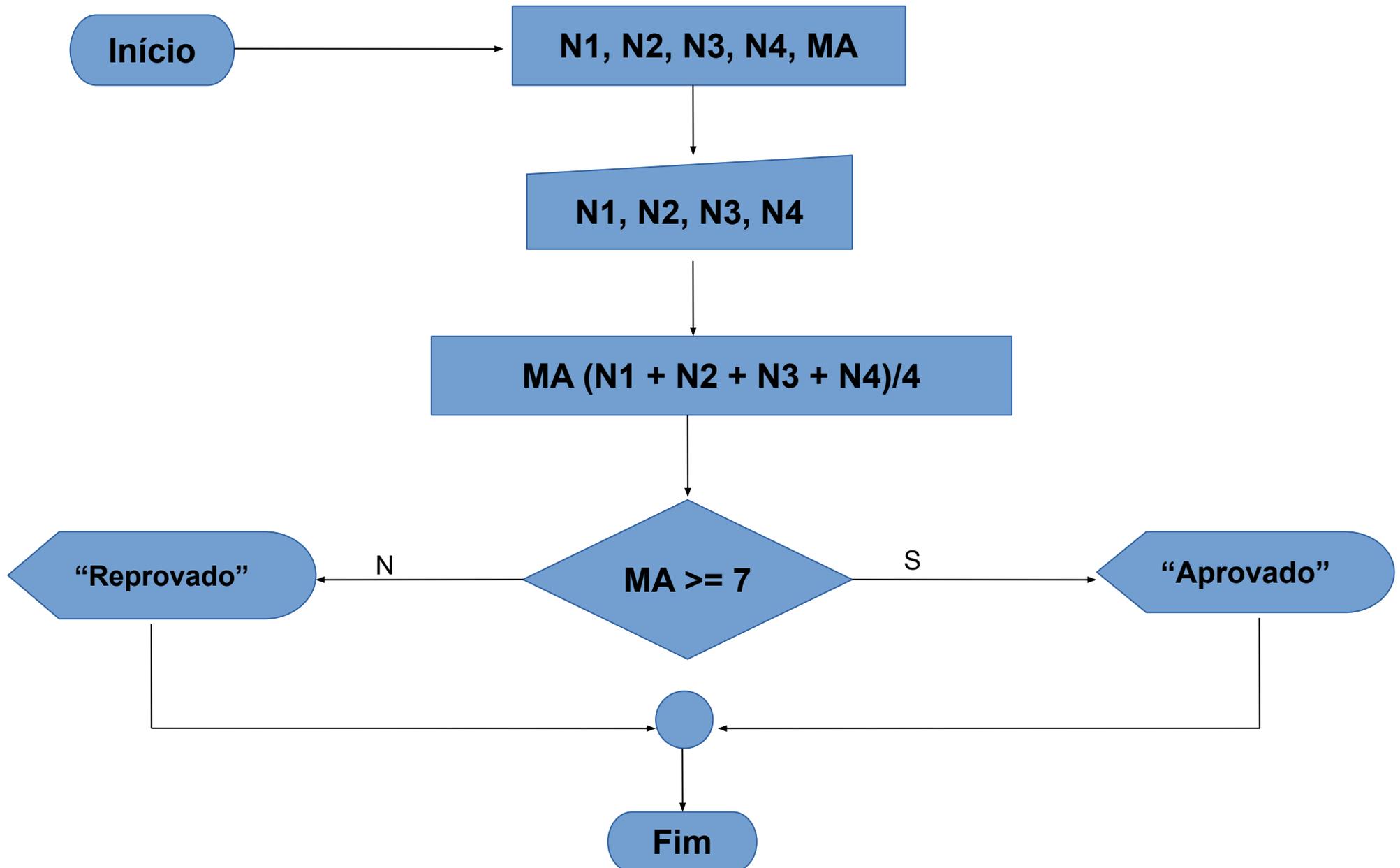
fimse

fim

Desvio Condicional Composto

- Acontece quando em uma dada situação existem duas alternativas que dependem de uma mesma condição.
- Se a condição for verdadeira, um caminho será seguido, se for falsa, outro caminho será tomado;

Estrutura Visual – Decisão Composta





inicio

//declaração de variáveis

real: N1, N2, N3, N4, MA;

//comando de entrada de dados

leia(N1, N2, N3, N4);

//processamento

MA (N1 + N2 + N3 + N4)/4;

//comando para saída de dados

escreva(MA);

//desvio condicional composto

se (MA >= 7) **então**

escreva("Aprovado");

senão

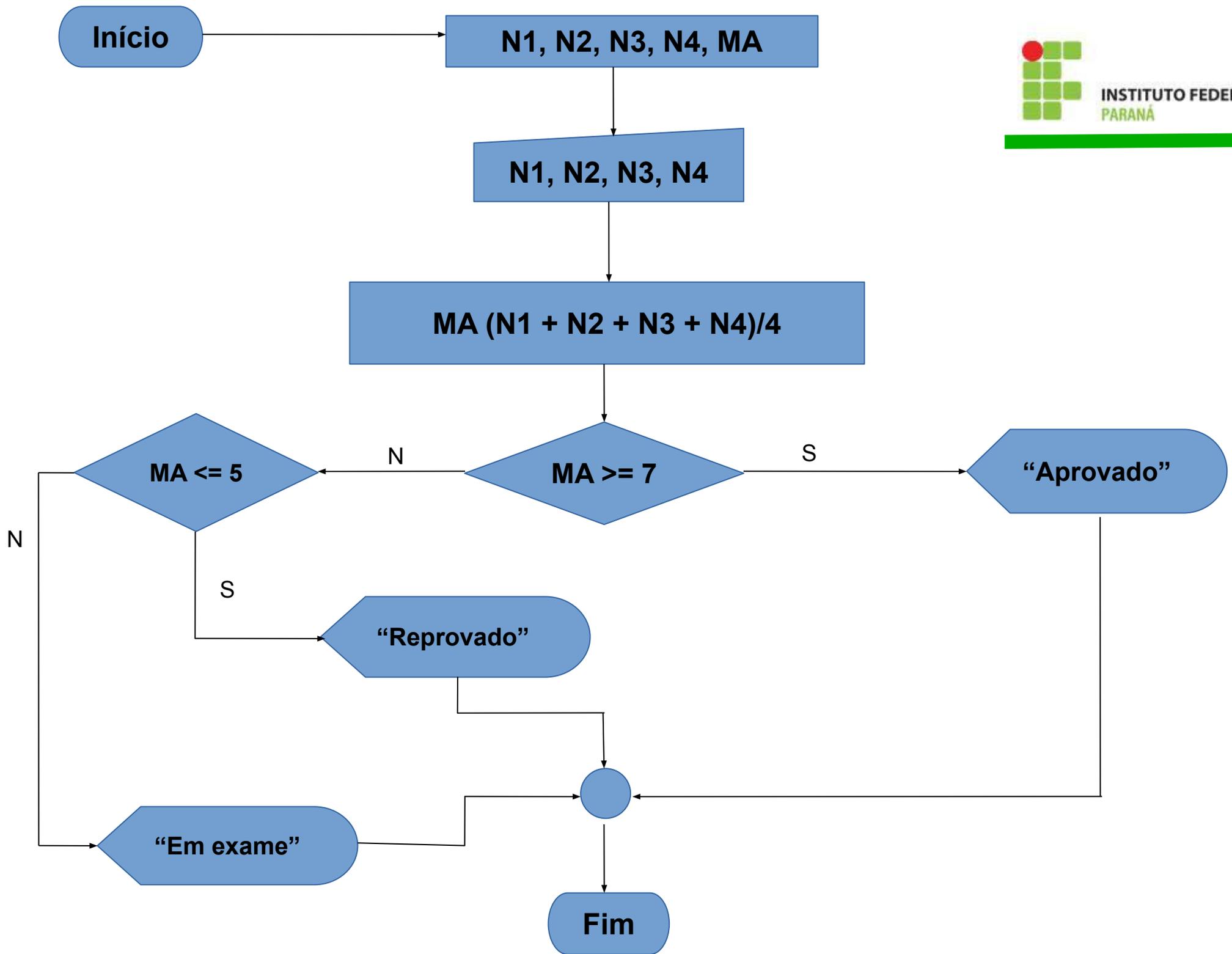
escreva("Reprovado");

fimse

fim

Seleção Encadeada

- Quando uma ação ou bloco deve ser executado se um grande conjunto de possibilidades ou condições forem satisfeitas;
- Quando a problemática sugere que seja utilizada decisão simples ou composta dentro de outras decisões já existentes;





início

```
//declaração de variáveis  
real: N1, N2, N3, N4, MA;  
//comando de entrada de dados  
leia(N1, N2, N3, N4);  
//processamento  
MA (N1 + N2 + N3 + N4)/4;  
//comando para saída de dados  
escreva(MA);  
//desvio condicional composto  
se (MA >= 7) então  
    escreva("Aprovado");  
senão  
    se (MA <=5) então  
        escreva("Reprovado");  
    senão  
        escreva ("Em exame");  
fimse  
fim
```