

Introdução a Lógica de Programação

Representações de Algoritmos

- Pseudocódigo ou Portugol
 - Analisar o enunciado do problema e escrever, por meio de regras predefinidas, os passos a serem seguidos para a resolução do problema
 - Vantagem → a passagem para o código em linguagem de programação é quase imediata
 - Desvantagem → exige o aprendizado das regras do pseudocódigo

Pseudocódigo – Estrutura Básica

// DESCRIÇÃO DO PROBLEMA

INICIO

DECLARAÇÃO DAS VARIÁVEIS;

**Var nome_variável: tipo;
Ou
Tipo nome_variável;**

Instrução 1;
Instrução 2;
.....

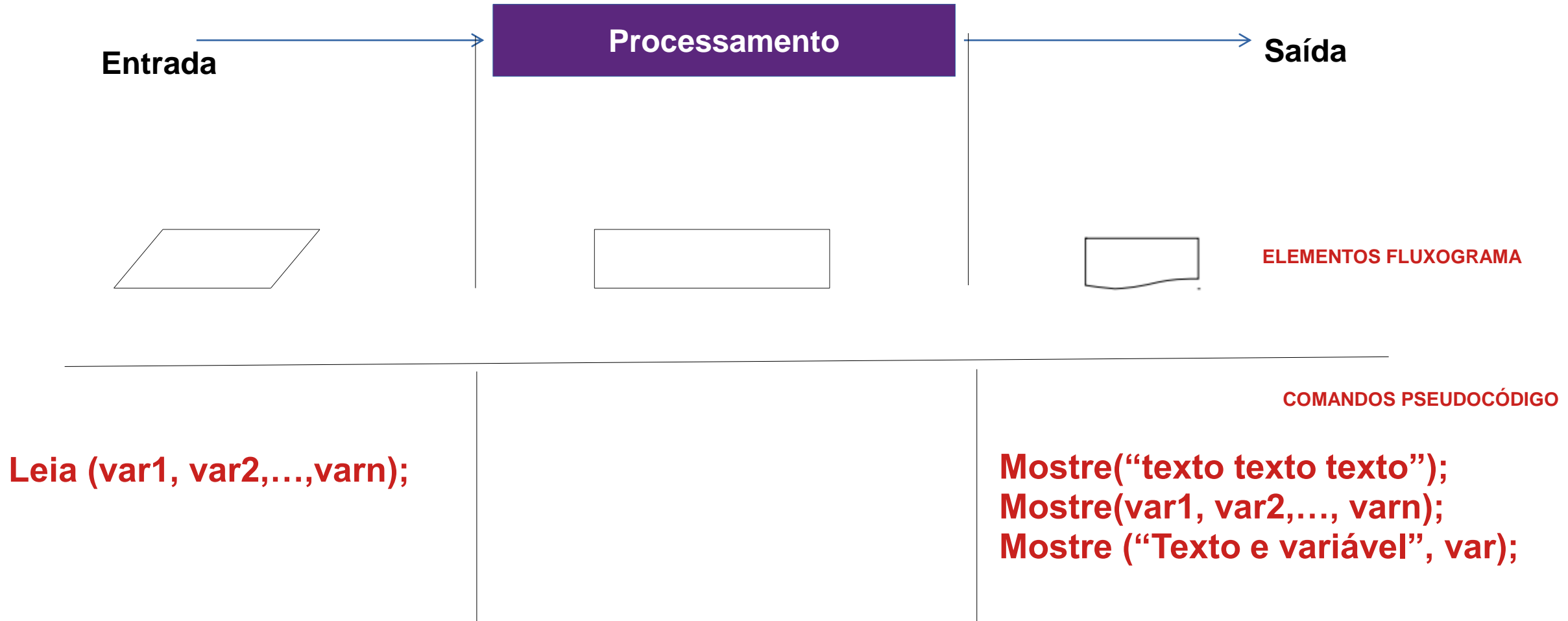
?

**SEQUENCIAIS: ENTRADA/SAÍDA;
CONDICIONAIS;
REPETIÇÕES**

Instrução n

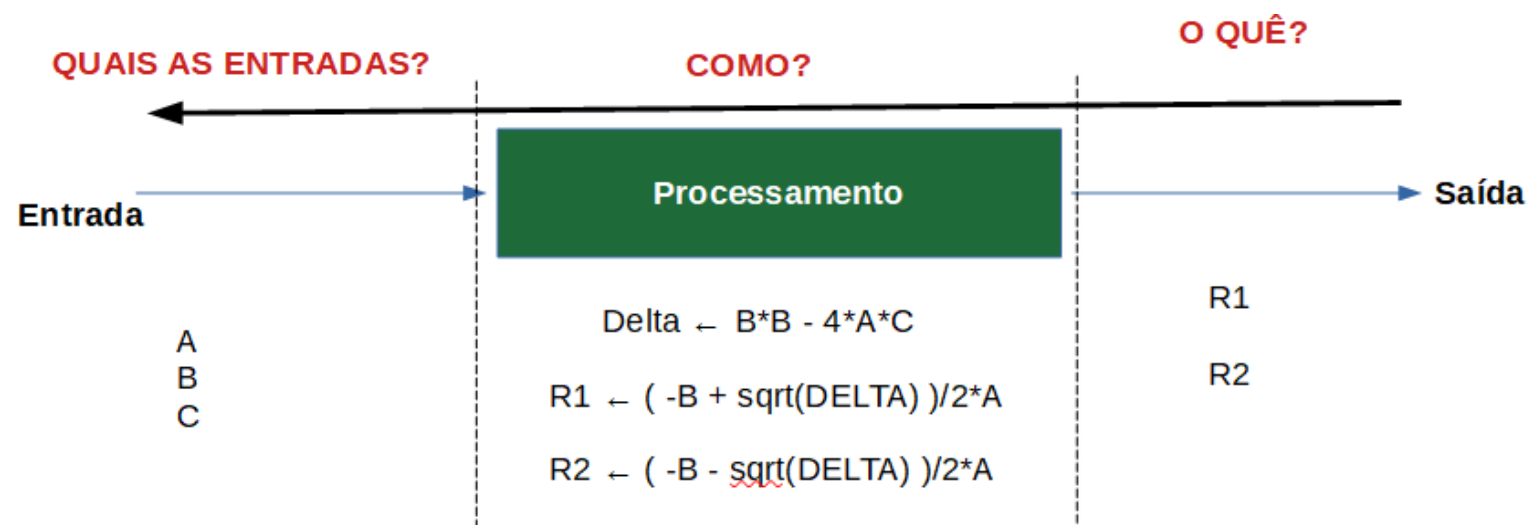
FIM

Instruções: Entrada e Saída

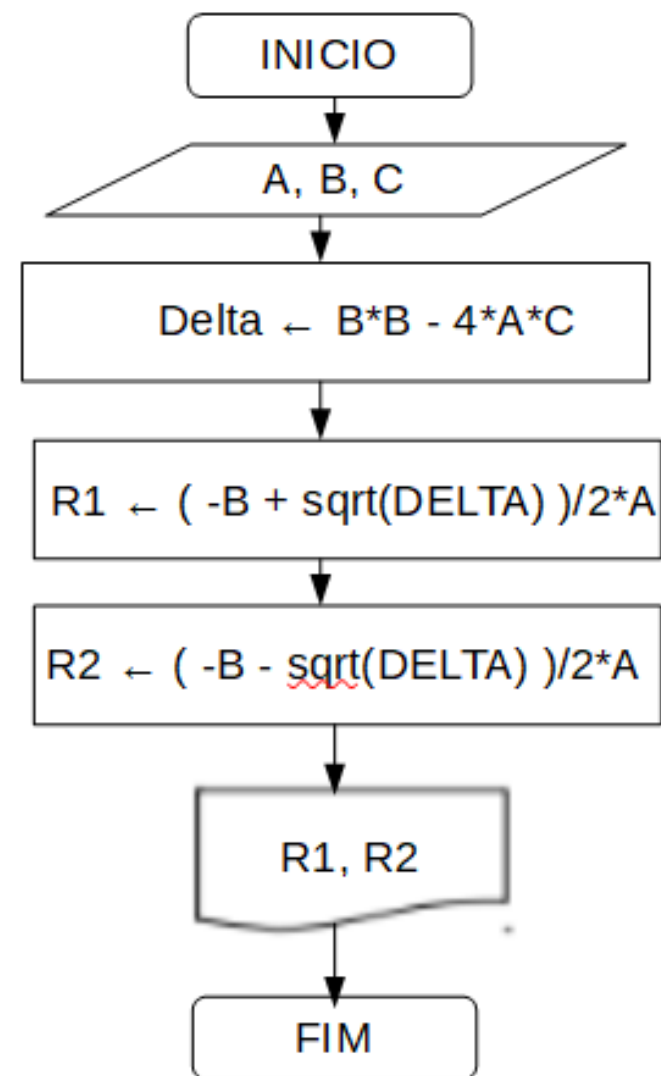


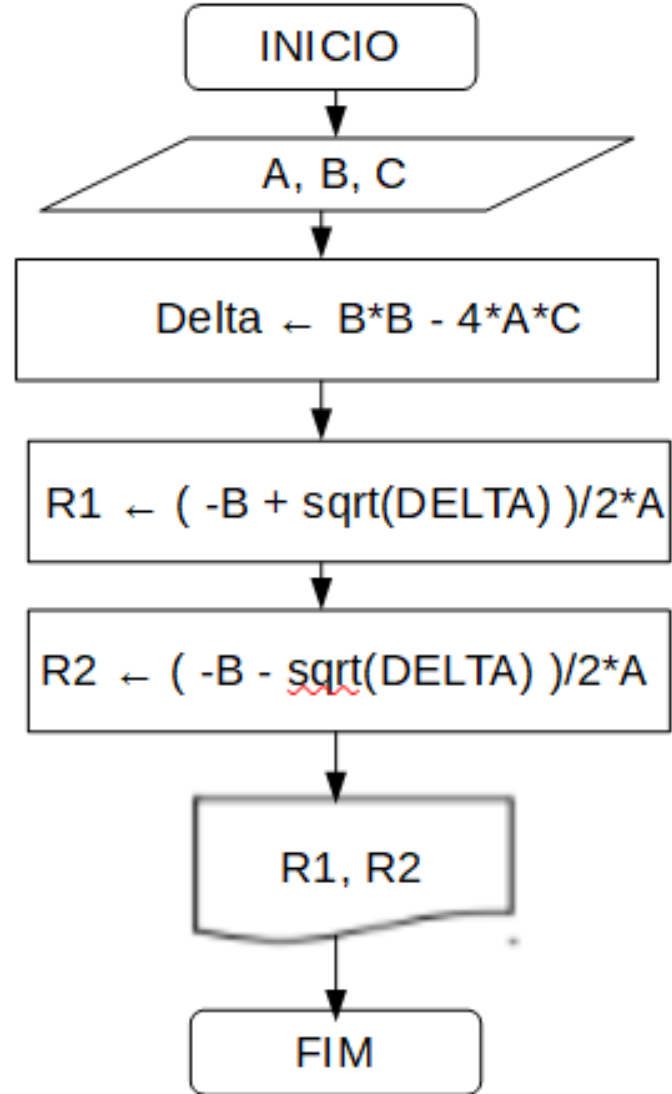
Exemplo

- Mostrar as etapas de entrada, processamento e saída que calcula as raízes de uma equação de segundo grau. Considere que existam duas raízes reais diferentes.
- Crie o respectivo pseudocódigo



FLUXOGRAMA





Tipos de Processamento

- Ao elaborar um algoritmo, devemos ter em mente qual o tipo de processamento será executado.
- Basicamente, existem 3 tipos de processamento

Processamento sequencial

Processamento condicional

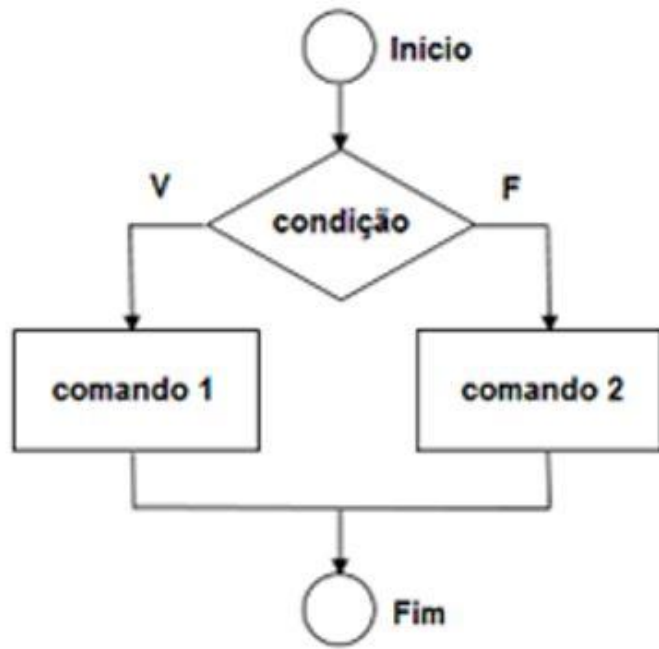
Processamento com repetição

- Repetição determinada
- Repetição indeterminada

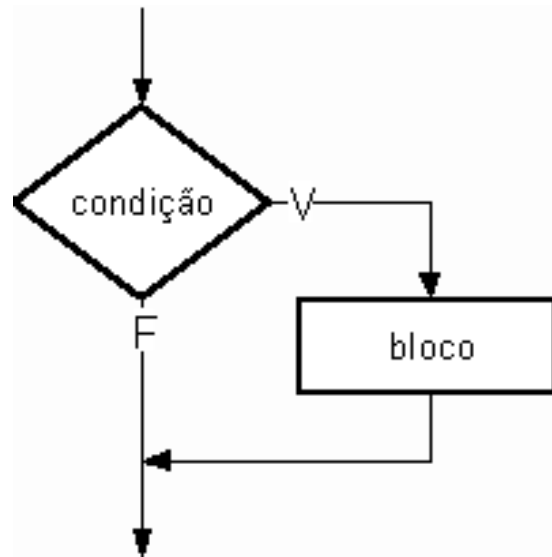
Processamento Condicional

- Um conjunto de instruções (pode ser apenas uma) pode ou não ser executado
- Depende de uma condição
- Se a condição testada for verdadeira, o conjunto de instruções é executado

Estrutura Condicional



```
Se (condição v)
v      Comando 1;
Senão
F      Comando 2;
Fim-Se
```



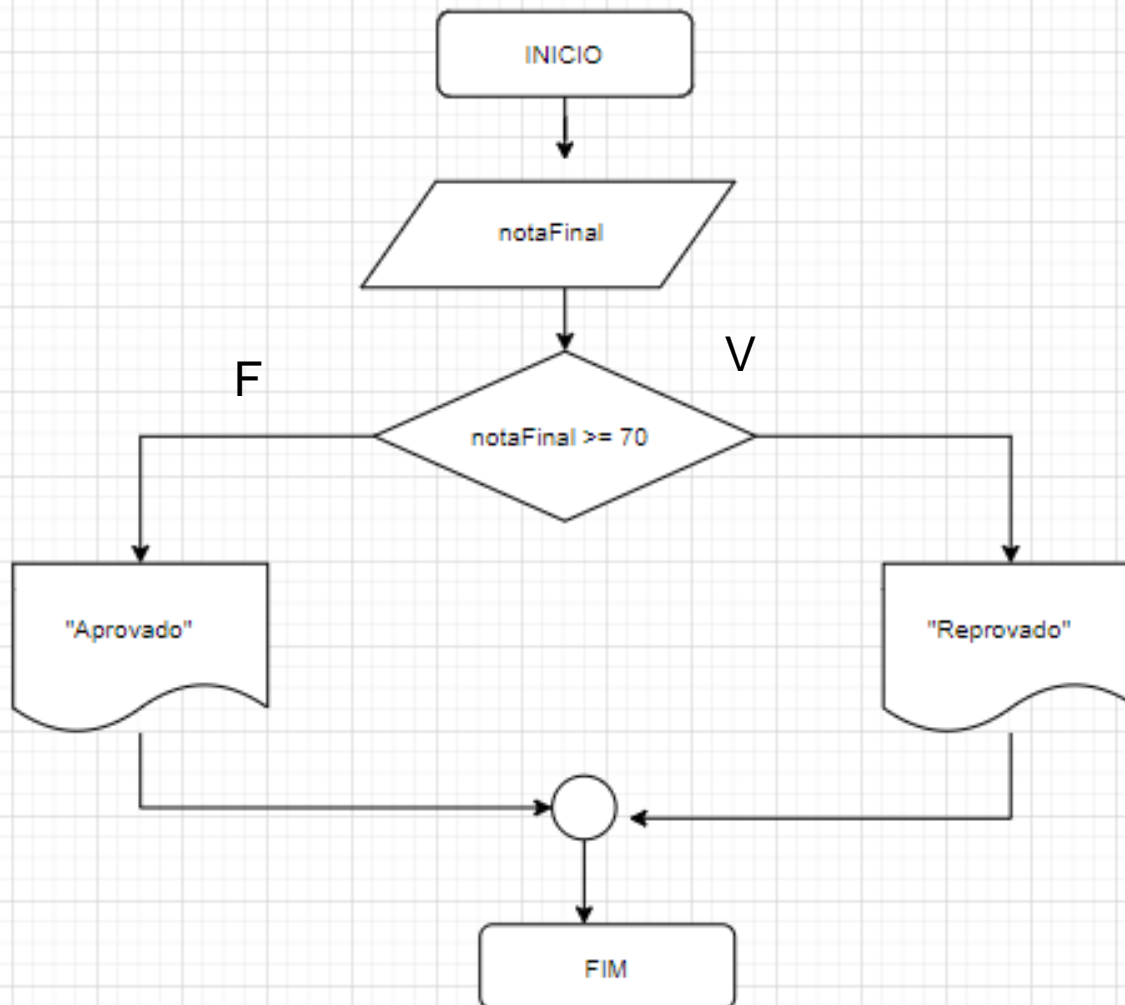
```
Se (condição v)
v      Bloco;
Fim-Se
```

Exemplo

- Leia a nota final de um aluno e mostre uma mensagem informando se ele foi aprovado ou reprovado em uma disciplina. Para ser aprovado, a nota final tem que ser maior ou igual a 70.

Exemplo

- Leia a nota final de um aluno e mostre uma mensagem informando se ele foi aprovado ou reprovado em uma disciplina. Para ser aprovado, a nota final tem que ser maior ou igual a 70.



INICIO

real notaFinal;

Mostre ("Digite a nota final da disciplina");

Leia (notaFinal);

Se (notaFinal >= 70)

Mostre("Aprovado");

Senão

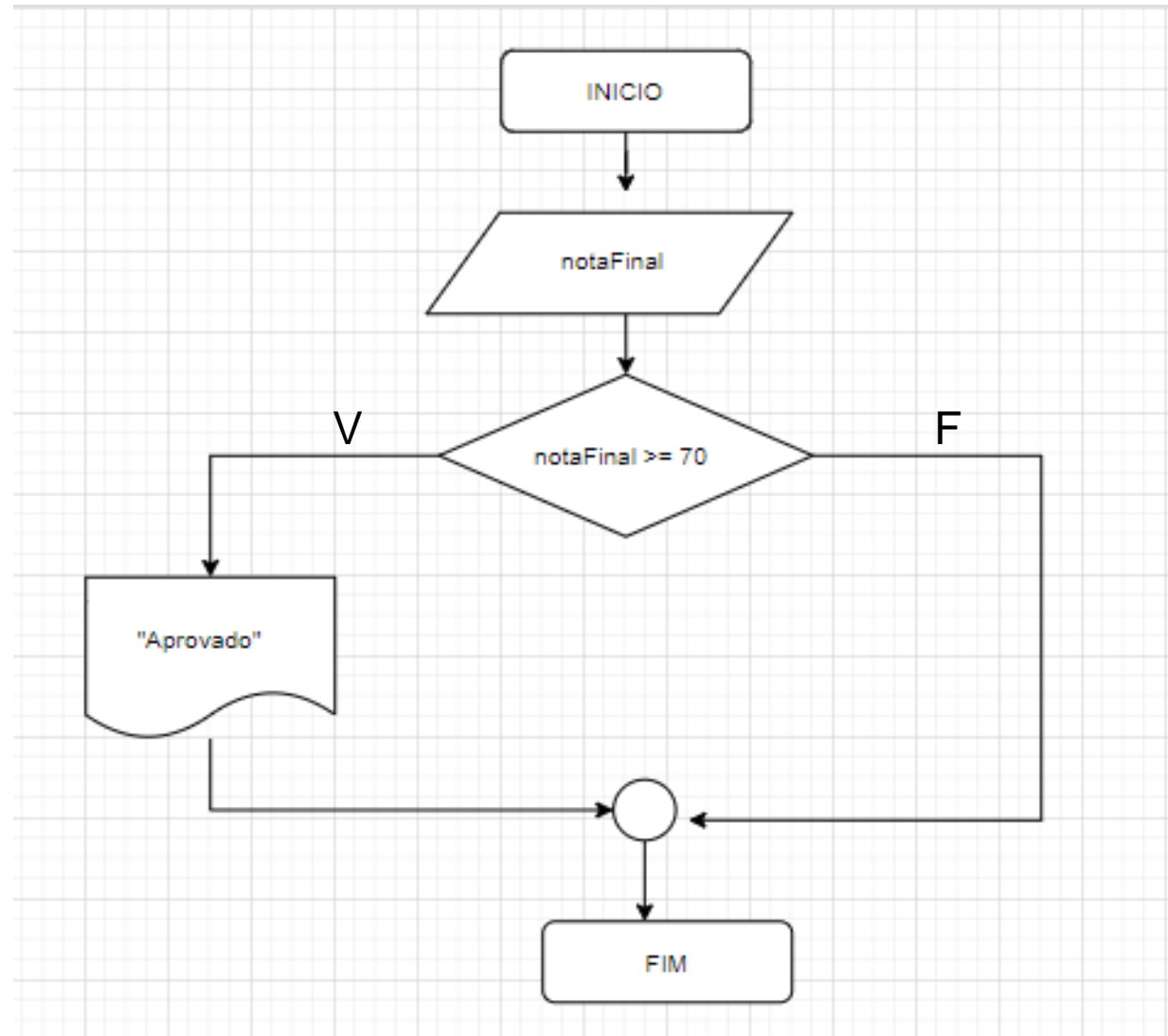
Mostre("Reprovado");

Fim-Se

FIM

Exemplo

- Leia a nota final de um aluno e mostre uma mensagem **informando se ele foi aprovado**. Para ser aprovado, a nota final tem que ser maior ou igual a 70.



Praticando...

- Elaborar um algoritmo (fluxograma e pseudocódigo) que classifique um triângulo em equilátero (3 lados iguais), escaleno (3 lados diferentes) ou isósceles (2 lados iguais).

Praticando...

- Elaborar um algoritmo (fluxograma e pseudocódigo) que efetue o cálculo do reajuste de salário de um funcionário. Considere que o funcionário deverá receber um reajuste de 15% caso seu salário seja menor que 500. Se o salário for maior ou igual a 500, mas menor ou igual a 1000, seu reajuste será de 10%, e caso seja ainda maior que 1000, o reajuste deverá ser de 5%.

Estruturas de Repetição

- Um conjunto de instruções (pode ser apenas uma) é executado um número definido ou indefinido de vezes
- Pode ser determinada por uma condição de parada
 - O conjunto de instruções é executado enquanto a condição for verdadeira
 - O teste da condição é realizado antes de qualquer operação

Estruturas de Repetição

- Também chamado de laços condicionais
- Repetem um conjunto de comandos em seu interior
- Exemplo
 - Imprimir a soma dos números inteiro de 1 a N
 - $Soma = 1 + 2 + 3 + \dots + N$
 - Necessidade de se identificar o que deve ser repetido no algoritmo

$$Soma = 1 \oplus 2 \oplus 3 \oplus \dots \oplus N$$

Comandos de Repetições

- Uma estrutura de repetição permite que uma sequência de comandos seja executada repetidamente, enquanto determinadas condições são satisfeitas.

Repetição Contada

```
for(início;condição;incremento) {  
    sequência de comandos;  
}
```

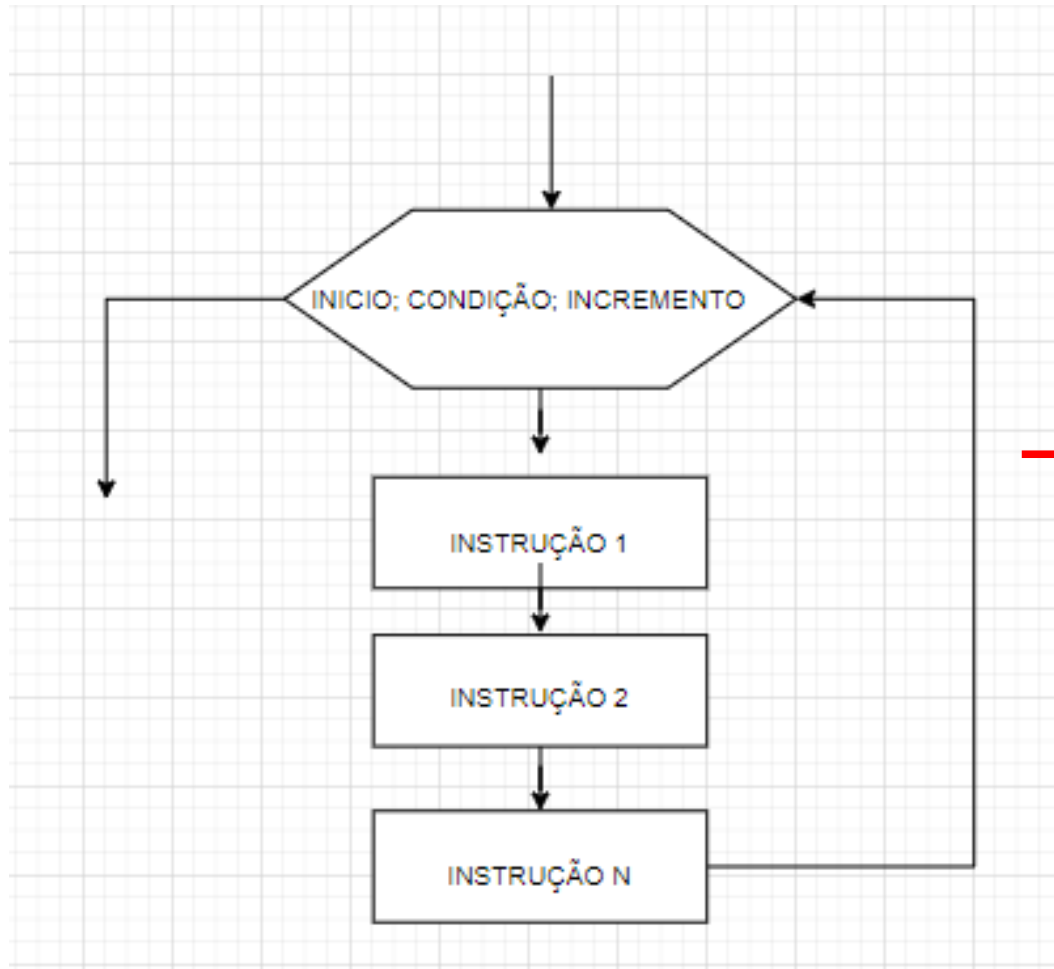
Repetição com Teste no Início

```
início;  
while (condição) {  
    sequência de comandos;  
    Incremento;  
}
```

Repetição com Teste no Final

```
do {  
    sequência de comandos;  
} while (condição);
```

Estrutura de Repetição: Para...Fim-Para



Para (valor inicial ; condição ; incremento)

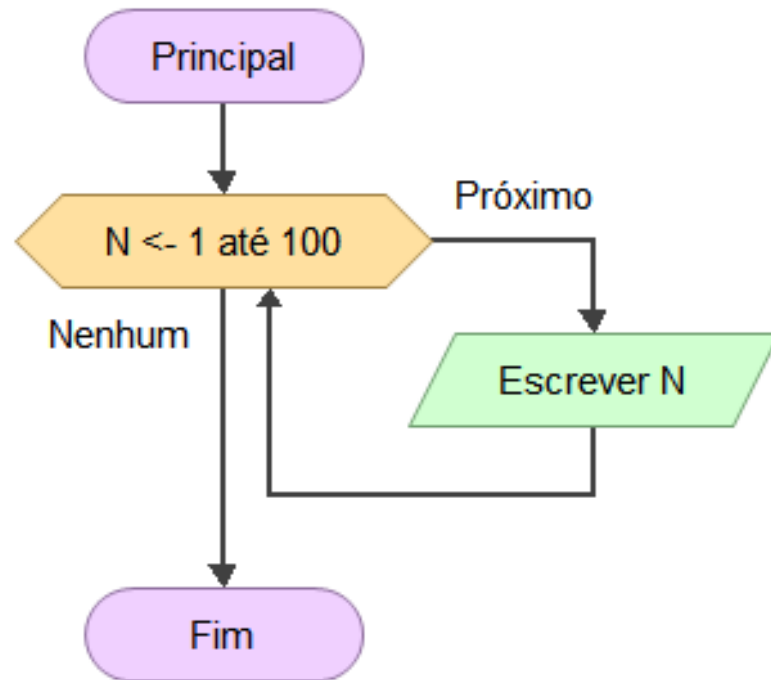
Instrução 1;
Instrução 2;

Instrução N;

Fim-Para

Exemplo

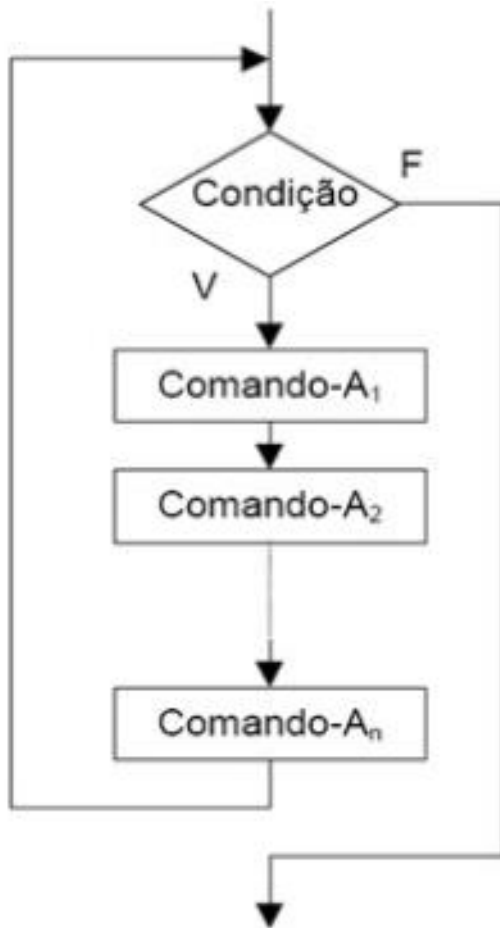
- Mostrar todos os números inteiros de 1 até 100.



```
INÍCIO
|
Inteiro N;
|
Para ( N=1; N<=100 ; 1)
|
    Mostre(N);
|
Fim-Para
|
FIM
```

Estrutura de Repetição: Enquanto/Fim_Enquanto

PRÉ-CONDIÇÃO



Enquanto (condição for V)

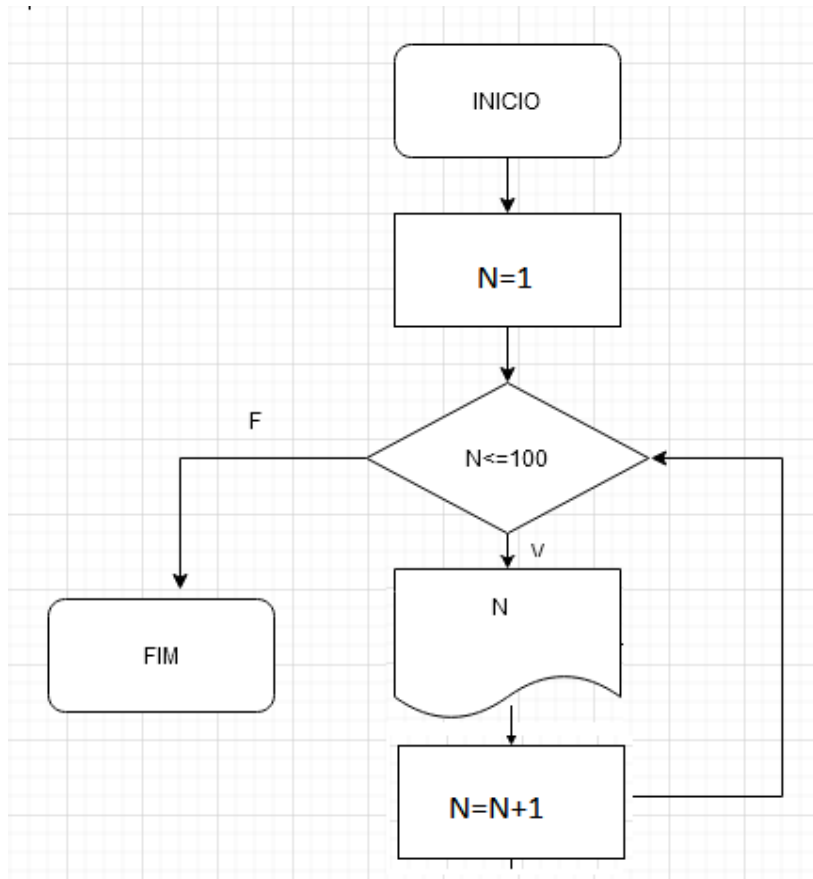
Comando-A1;
Comando-A2;

Comando-An;

Fim-Enquanto

Exemplo

- Mostrar todos os números inteiros de 1 até 100.



INÍCIO

INTEIRO N;

N=1;

Enquanto (N<=100)

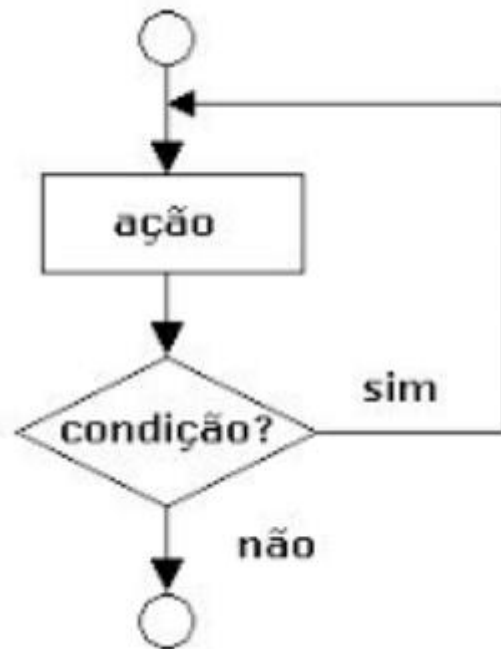
 Mostre (N);

 N=N+1;

Fim-Enquanto

FIM

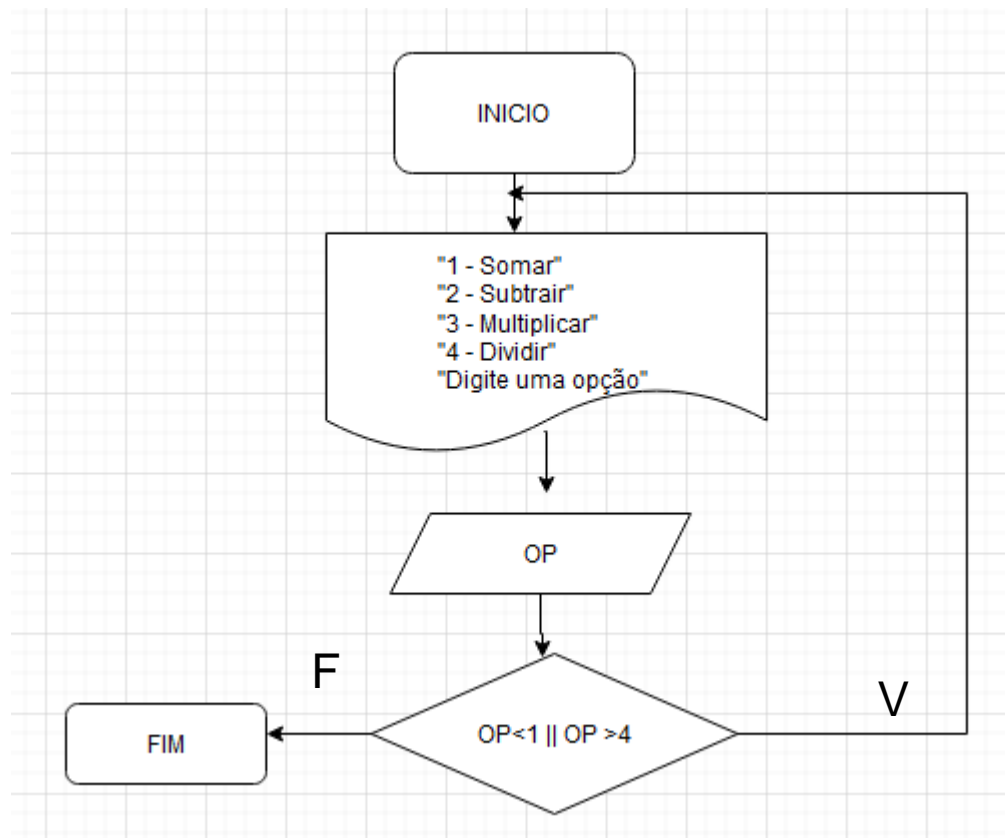
Estrutura de Repetição: Faça...Enquanto



```
Faça  
|  
Ação;  
Enquanto (condição for V)
```


Exemplo

- Faça um algoritmo que mostre um Menu com as opções das 4 operações aritméticas



INÍCIO

inteiro OP;

Faça

Mostre(" 1 – Somar
2 – Subtrair
3 – Multiplicar
4 – Dividir
Digite uma opção ");

Leia (OP);

Enquanto (OP < 1 || OP > 4)

FIM

Teste de Mesa

- Após desenvolver um algoritmo é preciso testá-lo. Uma maneira de se fazer isso é usando o **teste de mesa**
- Basicamente, esse teste consiste em seguir as instruções do algoritmo de maneira precisa para verificar se o procedimento utilizado está correto ou não
- Tentar utilizar um caso onde se conhece o resultado esperado
- Permite reconstituir o passo a passo do algoritmo

Teste de Mesa

- Criar uma tabela de modo que
- Cada coluna representa uma variável
- As linhas correspondem as alterações naquela variável (de cima para baixo)

Exercícios – Faça o teste de mesa

```
Leia A;  
Leia B;  
Enquanto ( A < B)  
    A= A + 1;  
    Mostre( A);  
Fim Enquanto
```

```
X = 4;  
enquanto (X < 5)  
    X= X - 1;  
    Mostre(X) ;  
fim enquanto
```

```
X=4;  
enquanto (X < 5)  
    Mostre(X) ;  
fim enquanto
```

Créditos

- Adaptações dos trabalhos dos professores
- André Backes
- Cleyton Caetano de Souza
- Jadsonlee da Silva Sá
- José Augusto Cintra

