

**INSTITUTO FEDERAL**

Paraná

Campus Assis Chateaubriand

# Algoritmos e Lógica de Programação

## Estruturas de Decisão

# Estruturas de Decisão

## Conteúdo

- Decisões com operadores relacionais;
- Estruturas de decisão simples: SE... ENTÃO;
- Estruturas de decisão duais: SE... ENTÃO... SENÃO;
- Estruturas de decisão aninhadas.

# Considerações Iniciais – Estruturas Sequenciais (1)

Até o momento, nos dedicamos a algoritmos que eram sequenciais:

- Os comandos eram executados em **ordem de execução linear**, um após o outro;
- Quaisquer que fossem os valores das variáveis de entrada e de quaisquer resultados intermediários, *os mesmos comandos eram executados e sempre na mesma ordem*;
- Após o último comando, o algoritmo encerrava.

**Algoritmo** Calcula\_IMC;

**Variaveis**

vlPeso, vlAltura, vlIMC: **real**;

**Início do algoritmo**

```
1. Escreva ("Entre com a altura");
2. Leia (vlAltura);
3. Escreva ("Entre com o peso");
4. Leia (vlPeso);
5. vlIMC ← vlPeso/vlAltura^2;
6. Escreva ("O IMC é", vlIMC);
Fim do algoritmo.
```

Este algoritmo usa o peso em quilogramas de uma pessoa e a sua altura em metros para determinar seu IMC. Quaisquer que sejam os valores digitados, o algoritmo executará sequencialmente os passos 1 a 6, sempre nessa ordem, todas as vezes. Mesmo que os valores informados pelo usuário sejam inválidos (como uma altura igual a zero ou um peso em gramas ao invés de quilos), o algoritmo irá executar sequencialmente. Claro, se a altura informada for zero, o passo 5 irá falhar com um erro (divisão por zero) mas em certas situações é conveniente que o próprio programador avoque para si a responsabilidade de criticar os valores das variáveis, para que o algoritmo **tome decisões** com base nesses valores.

# Considerações Iniciais – Estruturas Sequenciais (2)

O algoritmo `Calcula_IMC` é *estritamente linear*: o usuário fornece seu peso e altura (`vlPeso` e `vlAltura`), os quais são utilizados numa fórmula que calcula o IMC. Contudo, o IMC é utilizado para fornecer um diagnóstico conforme mostrado na tabela abaixo, e isso é impossível de ser feito com um fluxo de processamento sequencial. Para obter um diagnóstico, claramente é necessário *desviar o fluxo de processamento* de acordo com o IMC – ou seja, é necessário utilizar certas **estruturas de decisão**.

**Algoritmo** `Calcula_IMC`;

**Variaveis**

`vlPeso, vlAltura, vlIMC: real;`

**Início do algoritmo**

1. **Escreva** ("Entre com a altura");

2. **Leia** (`vlAltura`);

3. **Escreva** ("Entre com o peso");

4. **Leia** (`vlPeso`);

5. `vlIMC`  $\leftarrow$  `vlPeso/vlAltura^2`;

6. **Escreva** ("O IMC é", `vlIMC`);

**Fim do algoritmo.**

IMC	Diagnóstico
Abaixo de 18,5	Abaixo do peso
Entre 18,5 e 24,9	Peso normal
Entre 25 e 29,9	Sobrepeso
Entre 30 e 34,9	Obesidade grau 1
Entre 35 e 39,9	Obesidade grau 2
Mais do que 40	Obesidade grau 3

É o uso de estruturas de decisão que permite comparar o valor do IMC com as faixas de valores mostradas na tabela.

# Testes Lógicos e Estruturas de Decisão (1)

O diagnóstico varia de acordo com a faixa do IMC. Sem estruturas de decisão, não há como obter um diagnóstico!

**Algoritmo** Calcula\_IMC;

**Variaveis**

vlPeso, vlAltura, vlIMC: **real**;  
dsGrauObes : **caractere**;

**Início do algoritmo**

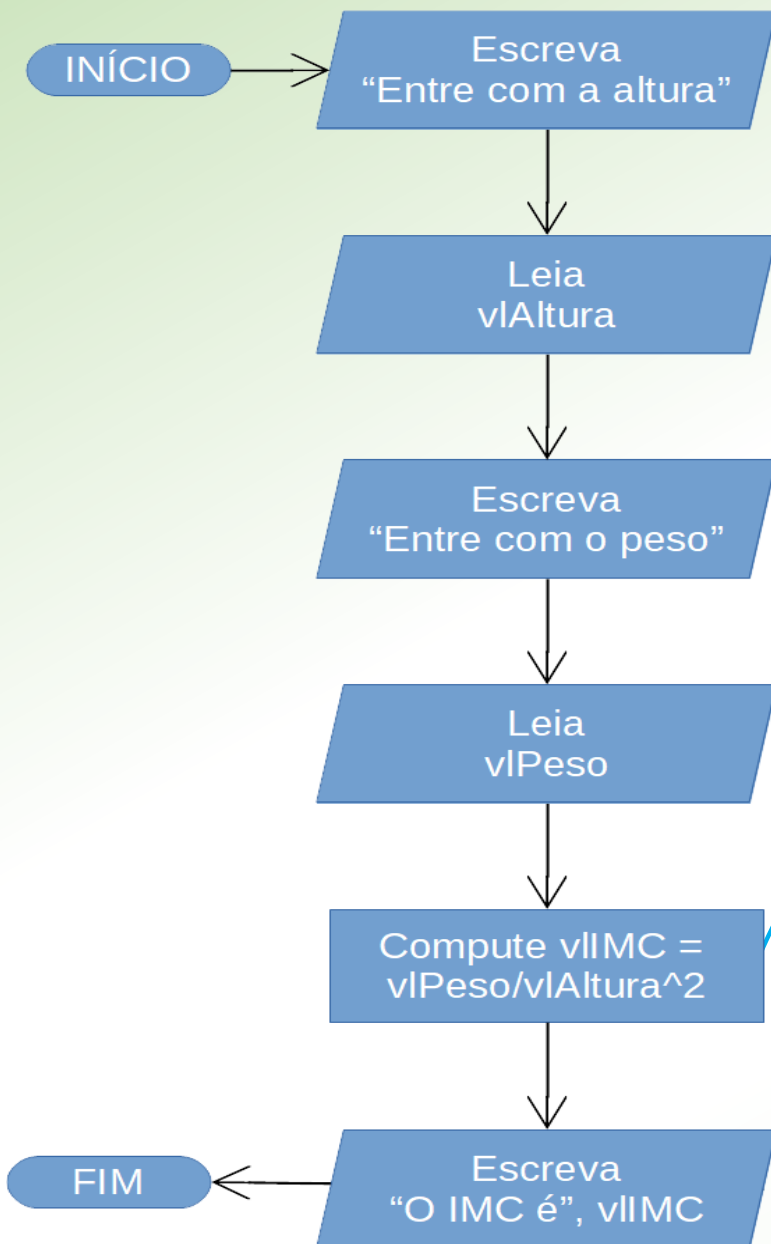
```
1. Escreva ("Entre com a altura");
2. Leia (vlAltura);
3. Escreva ("Entre com o peso");
4. Leia (vlPeso);
5. vlIMC ← vlPeso/vlAltura^2;
6. Escreva ("O IMC é", valorIMC);
7. Se (vlIMC<18,5) então
8.     dsGrauObes ← "Abaixo do peso";
9. Senão se (vlIMC<25) então
10.    dsGrauObes ← "Peso normal";
11. Senão se (vlIMC<30) então
12.    dsGrauObes ← "Sobrepeso";
13. Senão se (vlIMC<35) então
14.    dsGrauObes ← "Obesidade grau 1";
15. Senão se (vlIMC<40) então
16.    dsGrauObes ← "Obesidade grau 2";
17. Senão
18.    dsGrauObes ← "Obesidade grau 3";
19. Escreva ("Diagnostico:", dsGrauObes);
Fim do algoritmo.
```

## Estruturas de Controle

Uma **estrutura de controle** determina o **fluxo do processamento** de um programa. A **estrutura de controle** mais óbvia e intuitiva é a **estrutura sequencial**, onde um comando é executado após o outro. É o que acontece no algoritmo ao lado, da linha 1 a 6, onde *a ordem de execução segue um padrão estritamente sequencial*. A partir daí, entre as linhas 7 e 15, *o fluxo de processamento sofre desvios* conforme o valor dos **testes lógicos** destacados em amarelo – uma análise do pseudo-código revela que atribui-se valores diferentes à variável dsGrauObes nas linhas **8, 10, 12, 14, 16 e 18**, mas que somente *uma* dessas linhas é executada, a depender do valor do IMC (vlIMC) calculado na linha 5. Isso faz sentido no contexto desse algoritmo, visto que o diagnóstico da pessoa (ou seja, o valor de dsGrauObes) depende do IMC dessa pessoa (vlIMC). Esse tipo de **estrutura de controle** é chamado de **estrutura de decisão** (ou **de seleção**).

As estruturas de decisão utilizam os comandos **SE**, **ENTÃO** e **SENÃO** associados a um teste lógico.

# Testes Lógicos e Estruturas de Decisão (2)

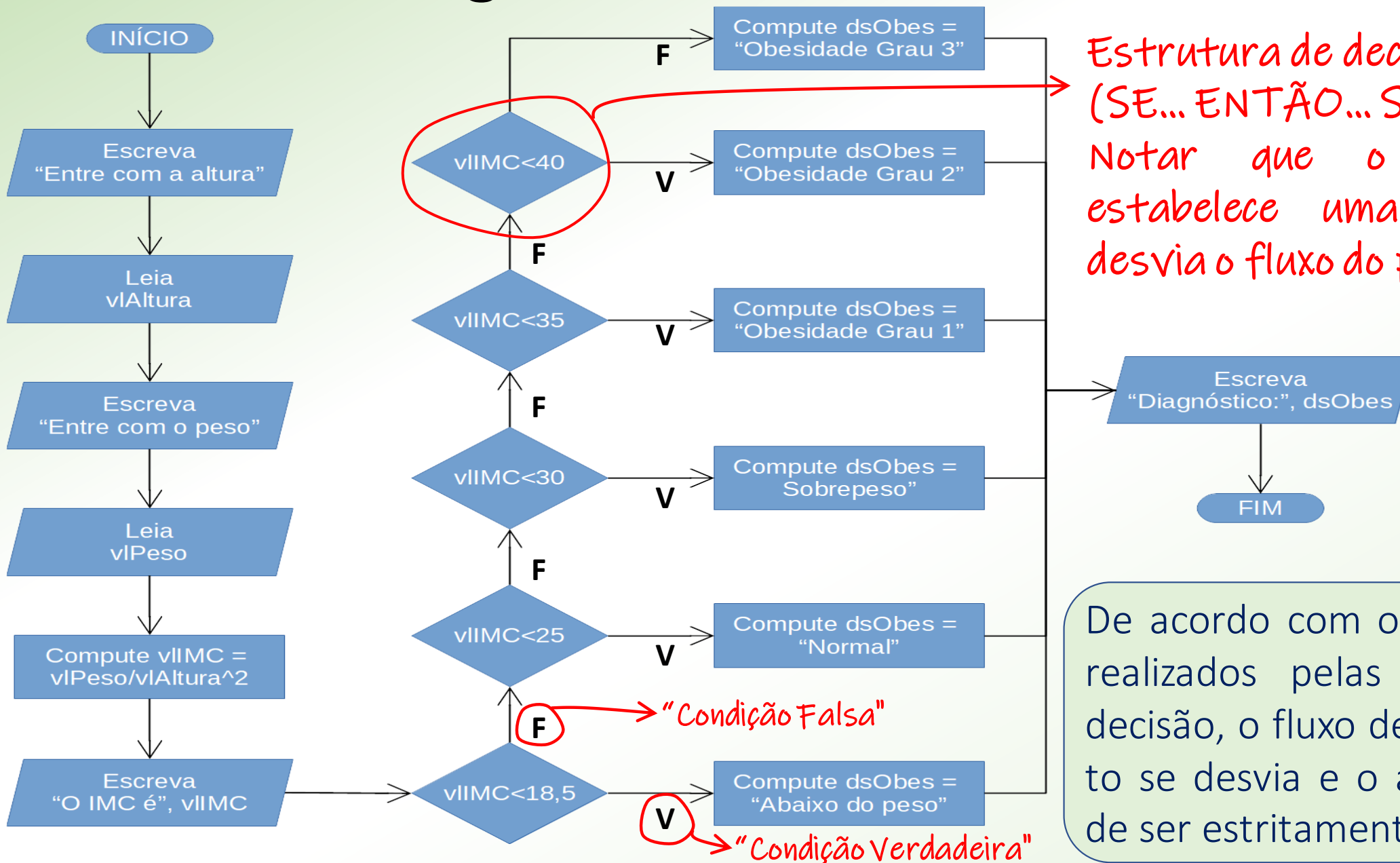


Este fluxograma descreve um fluxo de processamento puramente sequencial (estrutura sequencial), o que é evidenciado pelas setas: todos os comandos são executados linearmente, do primeiro ao último, até que o algoritmo seja concluído.

Como existem vários diagnósticos possíveis ("Abaixo do Peso", "Peso normal", "Sobre-peso", etc), claramente não é possível determinar o diagnóstico de uma pessoa com um algoritmo puramente sequencial, pois o fluxo de processamento tem que ser desviado de acordo com o valor de vlIMC.



# Testes Lógicos e Estruturas de Decisão (3)



Estrutura de decisão (SE... ENTÃO... SENÃO).  
Notar que o teste lógico estabelece uma condição que desvia o fluxo do processamento.

De acordo com os testes lógicos realizados pelas estruturas de decisão, o fluxo de processamento se desvia e o algoritmo deixa de ser estritamente sequencial.

# Testes Lógicos e Estruturas de Decisão (4)

Suponha por um momento que o programador queira minimizar a possibilidade de erros no cálculo do IMC, criticando os valores fornecidos pelo usuário. Especificamente, o usuário deveria entrar com a altura da pessoa em metros e o peso em quilogramas, mas e se a altura for dada em centímetros ou com o peso em gramas?

**Algoritmo** Calcula\_IMC;

**Variaveis**

vlPeso, vlAltura, vlIMC: **real**;

**Inicio do algoritmo**

1. **Escreva** ("Entre com a altura");

2. **Leia** (vlAltura);

3. **Escreva** ("Entre com o peso");

4. **Leia** (vlPeso);

5.  $vlIMC \leftarrow vlPeso/vlAltura^2$ ;

6. **Escreva** ("O IMC é", vlIMC);

**Fim do algoritmo.**

Como saber se a altura fornecida é dada em metros, e não em centímetros (1 m = 100 cm)?

Como saber se o peso fornecido é dado em quilogramas, e não em gramas (1 Kg = 1.000 g)?



# Testes Lógicos e Estruturas de Decisão (5)

Suponha por um momento que o programador queira minimizar a possibilidade de erros no cálculo do IMC, criticando os valores fornecidos pelo usuário. Especificamente, o usuário deveria entrar com a altura da pessoa em metros e o peso em quilogramas, mas e se a altura for dada em centímetros ou com o peso em gramas?

```
Algoritmo Calcula_IMC;  
Variaveis  
    vlPeso, vlAltura, vlIMC: real;  
Inicio do algoritmo  
1.  Escreva ("Entre com a altura");  
2.  Leia (vlAltura);  
3.  Se vlAltura > 100 então  
4.      vlAltura ← vlAltura/100;  
5.  Escreva ("Entre com o peso");  
6.  Leia (vlPeso);  
7.  Se vlPeso > 1000 então  
8.      vlPeso ← vlPeso/1000;  
9.  vlIMC ← vlPeso/vlAltura^2;  
10. Escreva ("O IMC é", vlIMC);  
Fim do algoritmo.
```

Sabemos que 1 metro tem 100 centímetros. Obviamente, ninguém tem mais de 100 metros de altura; logo, é razoável assumir que se a altura fornecida for maior do que 100, provavelmente foi fornecida em centímetros ao invés de metros.

Sabendo que um quilograma possui 1000 gramas e que é absolutamente impossível uma pessoa pesar mais de uma tonelada (1000 Kg), se o peso fornecido para o cálculo do IMC for maior do que 1000, presumivelmente ela foi fornecido em gramas, não em quilogramas.

# Testes Lógicos e Estruturas de Decisão (6)

Suponha por um momento que o programador queira minimizar a possibilidade de erros no cálculo do IMC, criticando os valores fornecidos pelo usuário. Especificamente, o usuário deveria entrar com a altura da pessoa em metros e o peso em quilogramas, mas e se a altura for dada em centímetros ou com o peso em gramas?

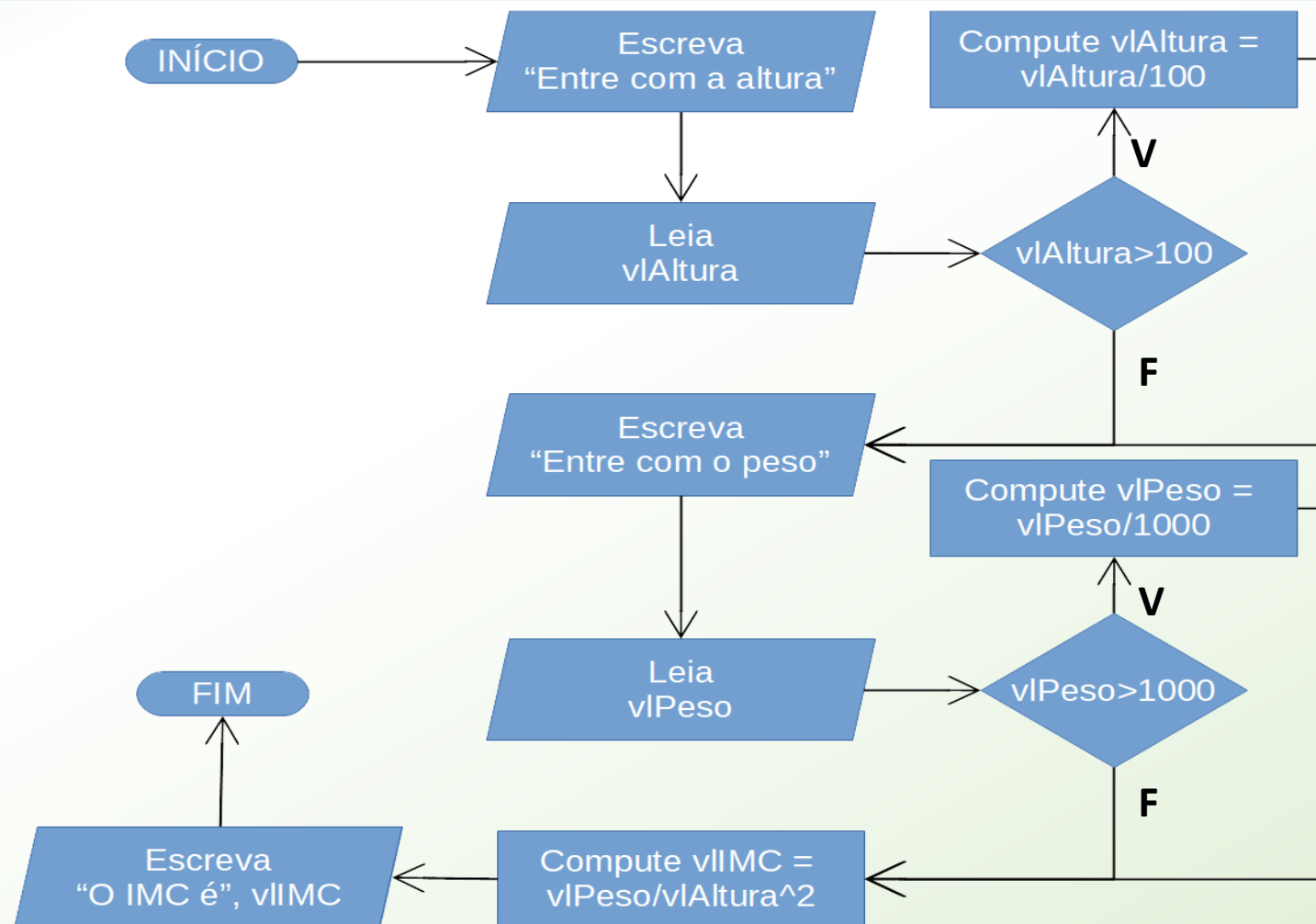
**Algoritmo** Calcula\_IMC;

**Variaveis**

vlPeso, vlAltura, vlIMC: **real**;

**Inicio do algoritmo**

```
1. Escreva ("Entre com a altura");
2. Leia (vlAltura);
3. Se vlAltura > 100 então
4.     vlAltura ← vlAltura/100;
5. Escreva ("Entre com o peso");
6. Leia (vlPeso);
7. Se vlPeso > 1000 então
8.     vlPeso ← vlPeso/1000;
9. vlIMC ← vlPeso/vlAltura^2;
10. Escreva ("O IMC é", vlIMC);
Fim do algoritmo.
```



# Testes Lógicos e Estruturas de Decisão (7)

Suponha por um momento que o programador queira minimizar a possibilidade de erros no cálculo do IMC, criticando os valores fornecidos pelo usuário. Especificamente, o usuário deveria entrar com a altura da pessoa em metros e o peso em quilogramas, mas e se a altura for dada em centímetros ou com o peso em gramas?

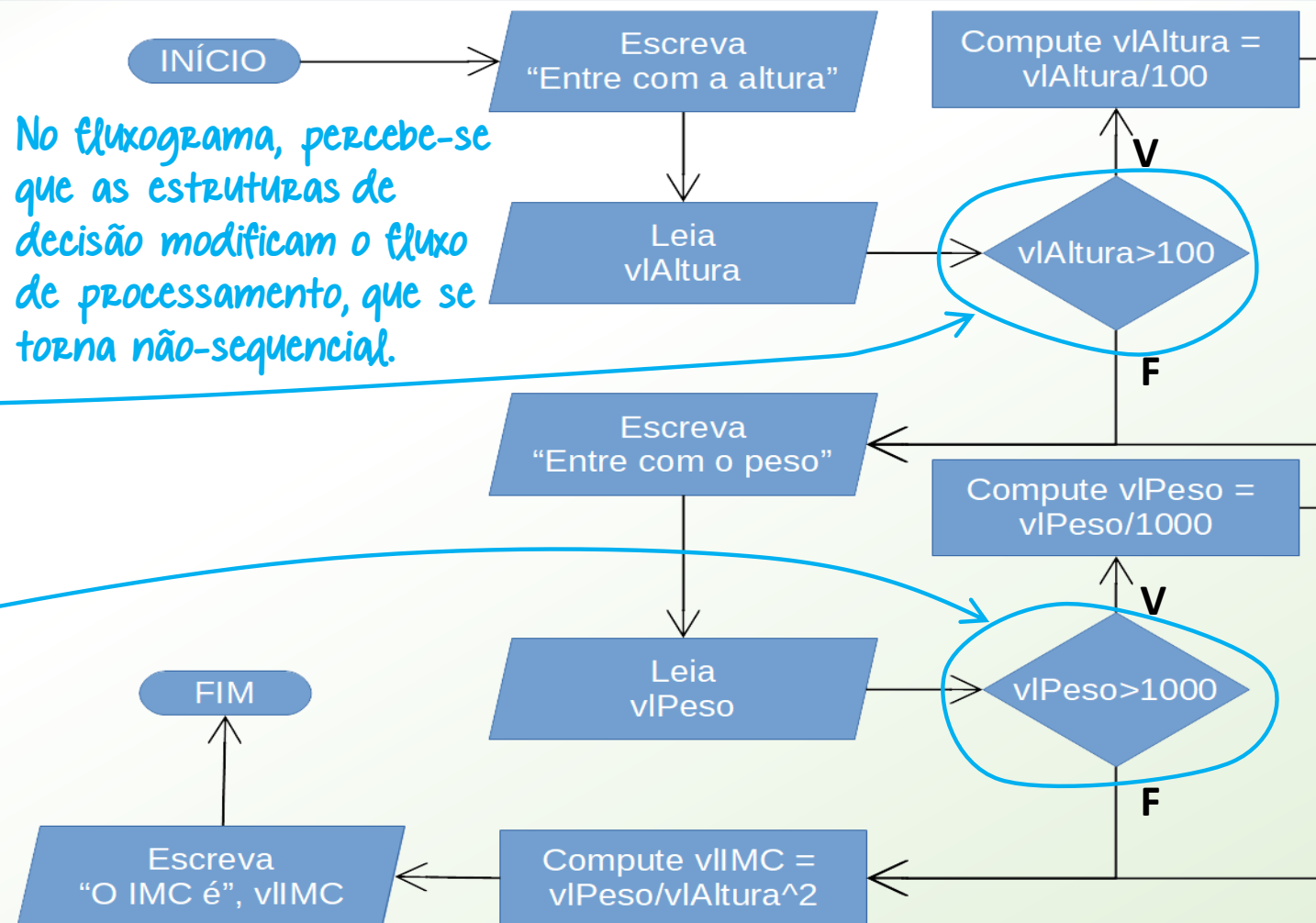
**Algoritmo** Calcula\_IMC;

**Variaveis**

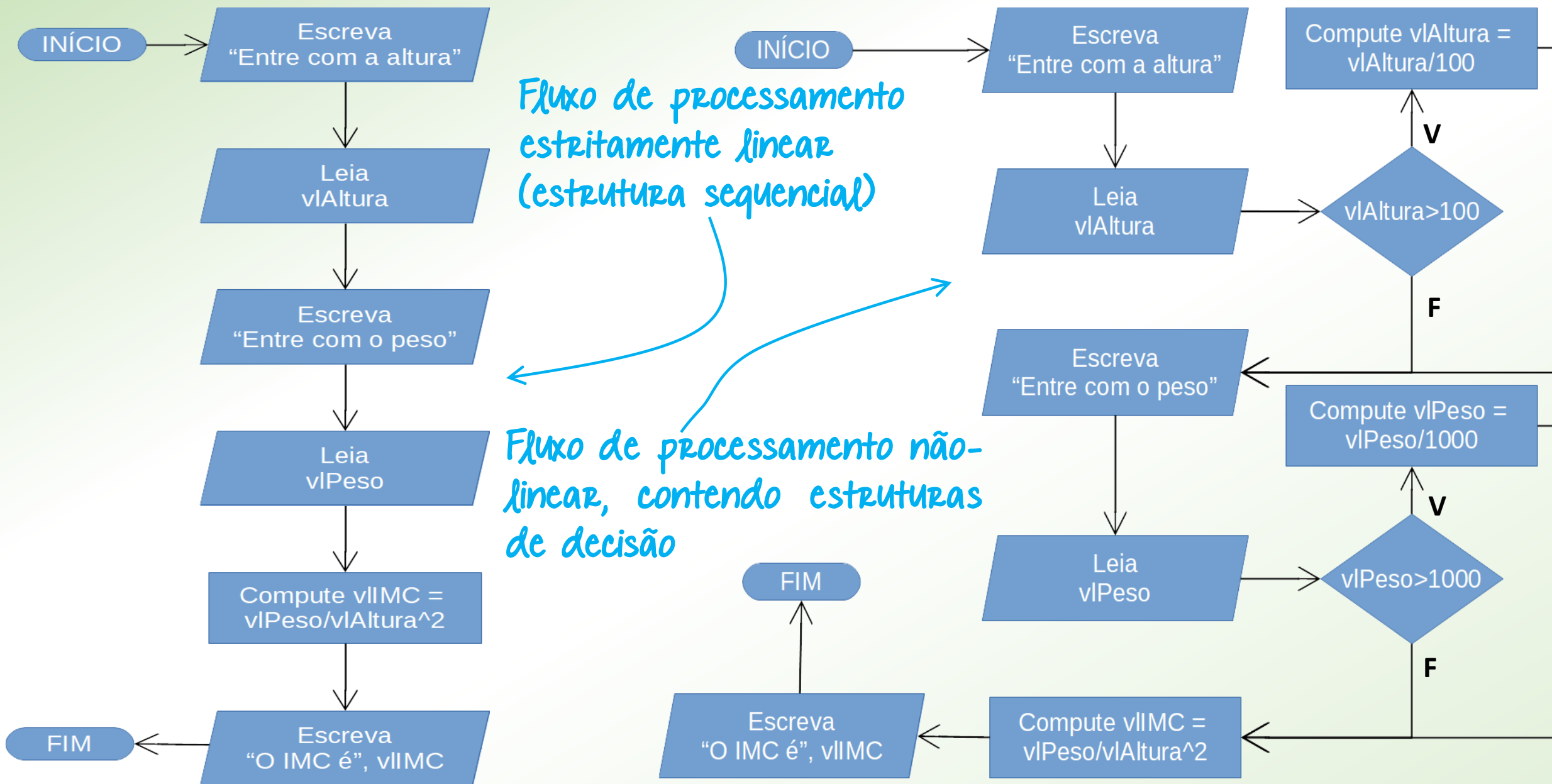
vlPeso, vlAltura, vlIMC: **real**;

**Início do algoritmo**

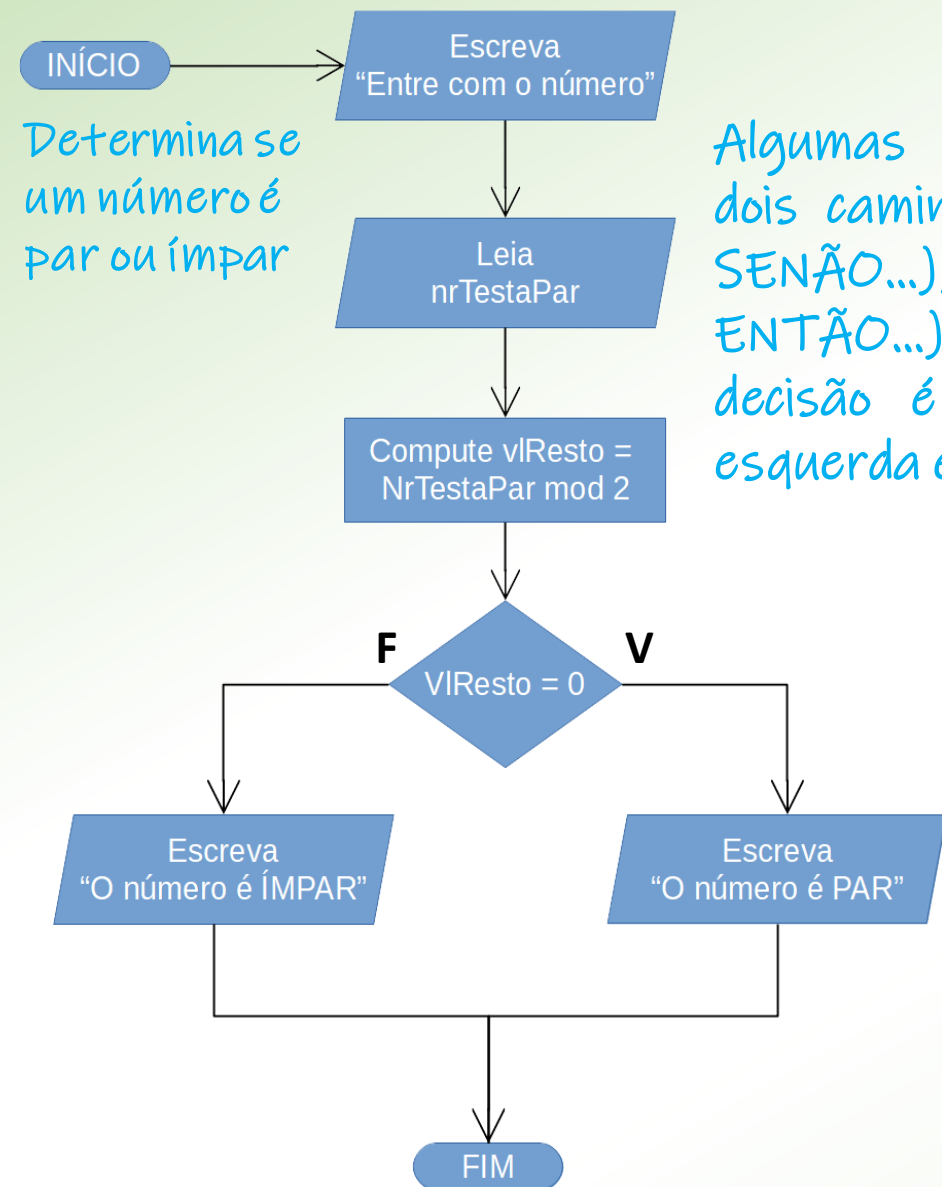
```
1. Escreva ("Entre com a altura");
2. Leia (vlAltura);
3. Se vlAltura > 100 então
4.     vlAltura ← vlAltura/100;
5. Escreva ("Entre com o peso");
6. Leia (vlPeso);
7. Se vlPeso > 1000 então
8.     vlPeso ← vlPeso/1000;
9. vlIMC ← vlPeso/vlAltura^2;
10. Escreva ("O IMC é", vlIMC);
Fim do algoritmo.
```



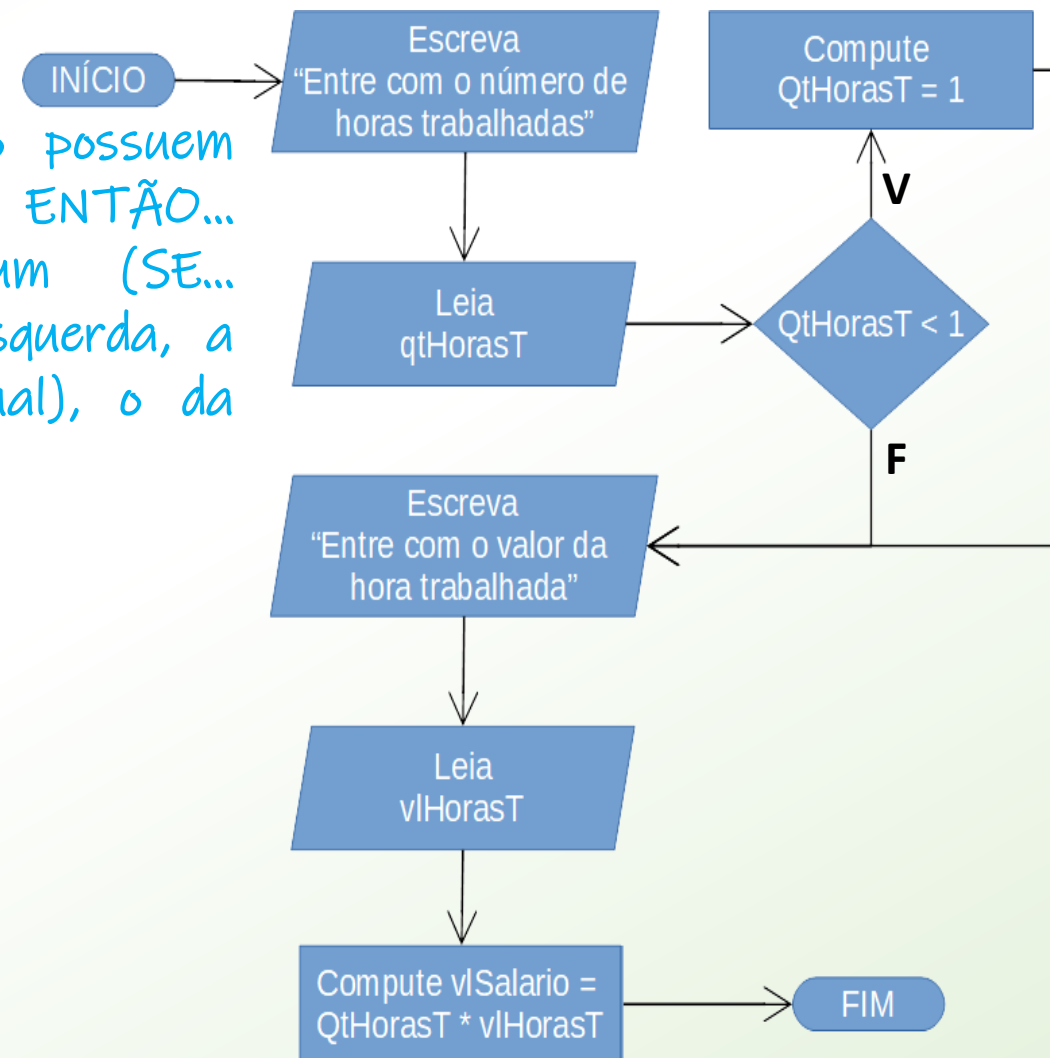
# Testes Lógicos e Estruturas de Decisão (8)



# Estruturas de Decisão Simples e Duais (1)



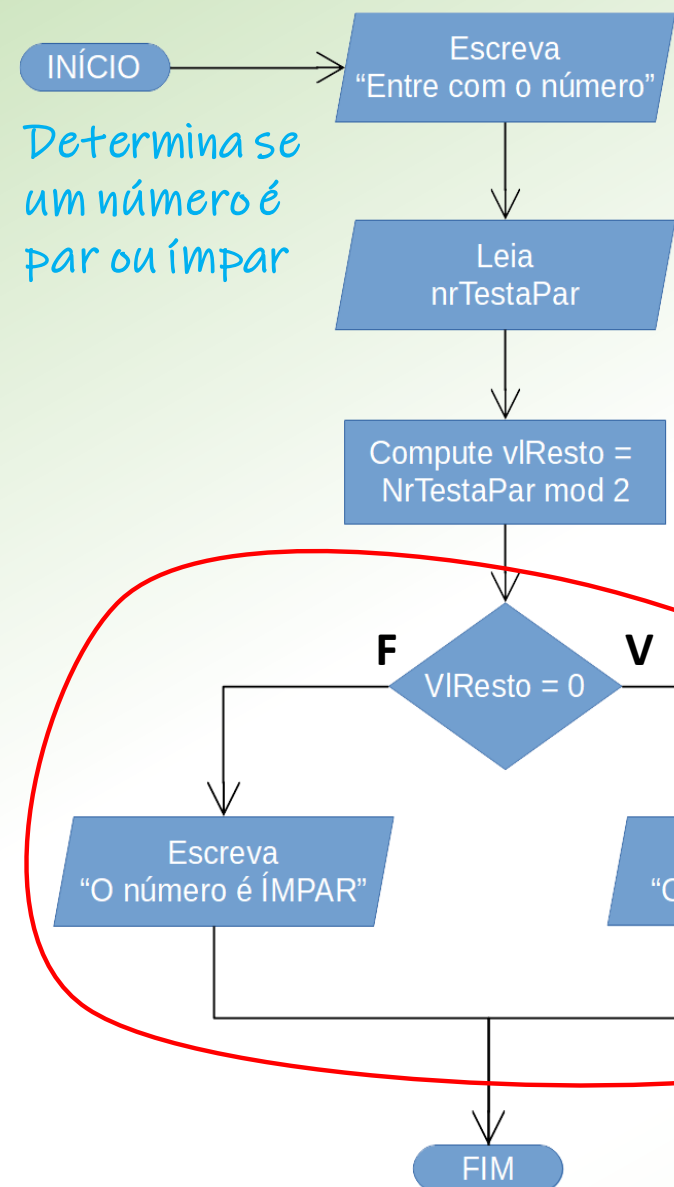
*Algumas estruturas de decisão possuem dois caminhos de execução (SE... ENTÃO... SENÃO...), outras apenas um (SE... ENTÃO...). No fluxograma à esquerda, a decisão é do primeiro tipo (dual), o da esquerda é do segundo (simples).*



*Calcula o salário de um profissional que trabalha por hora (paga-se, no mínimo, uma hora de trabalho).*



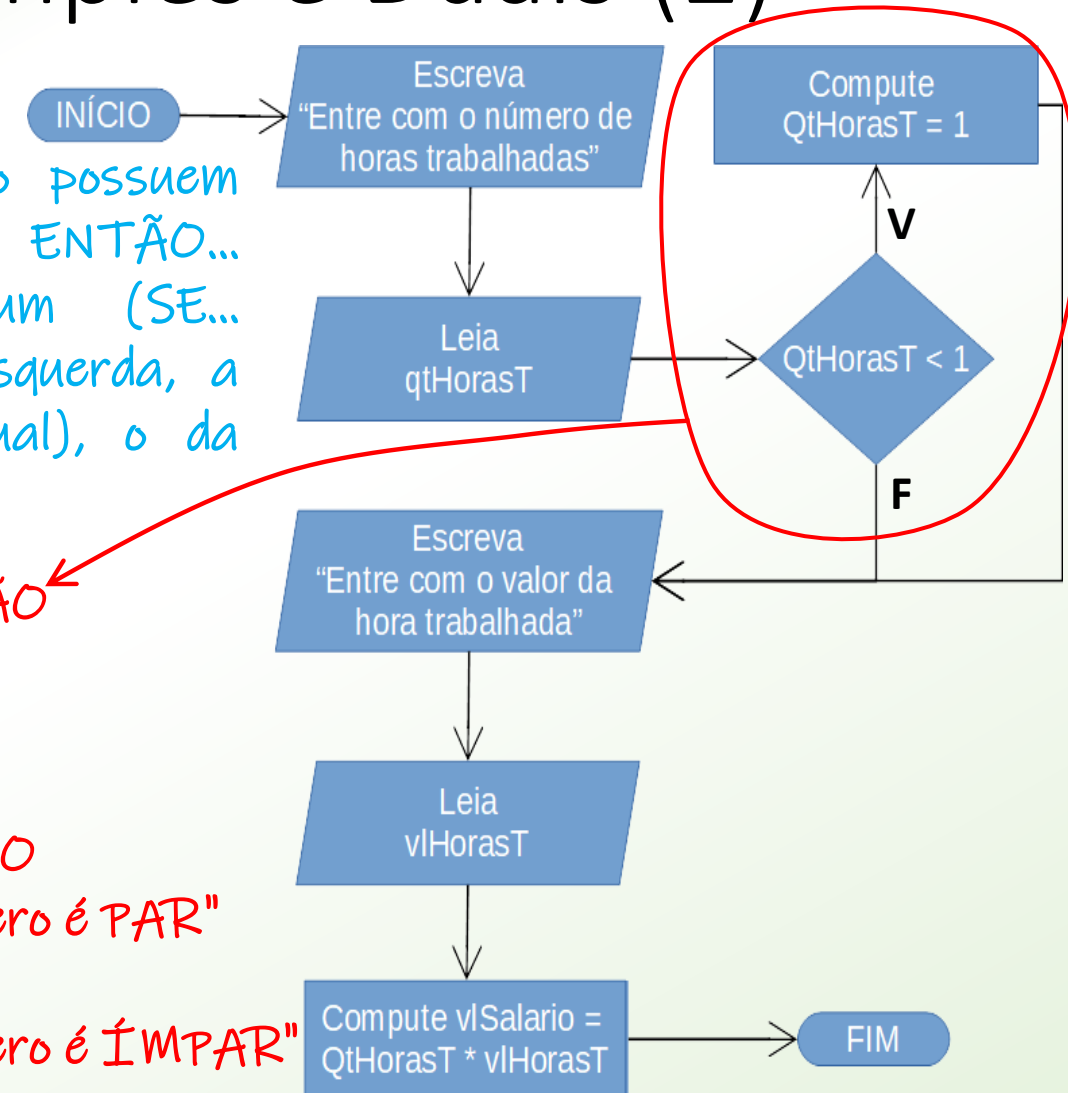
# Estruturas de Decisão Simples e Duais (2)



Algumas estruturas de decisão possuem dois caminhos de execução (SE... ENTÃO... SENÃO...), outras apenas um (SE... ENTÃO...). No fluxograma à esquerda, a decisão é do primeiro tipo (dual), o da esquerda é do segundo (simples).

SE  $qtHoras < 1$  ENTÃO  
 $qtHoras \leftarrow 1$

SE  $vlResto = 0$  ENTÃO  
ESCREVA "O número é PAR"  
SENÃO  
ESCREVA "O número é ÍMPAR"



Calcula o salário de um profissional que trabalha por hora (paga-se, no mínimo, uma hora de trabalho).