

Fundamentos de Programação de Computadores

Instruções de Decisão

Docente: Fátima Leal

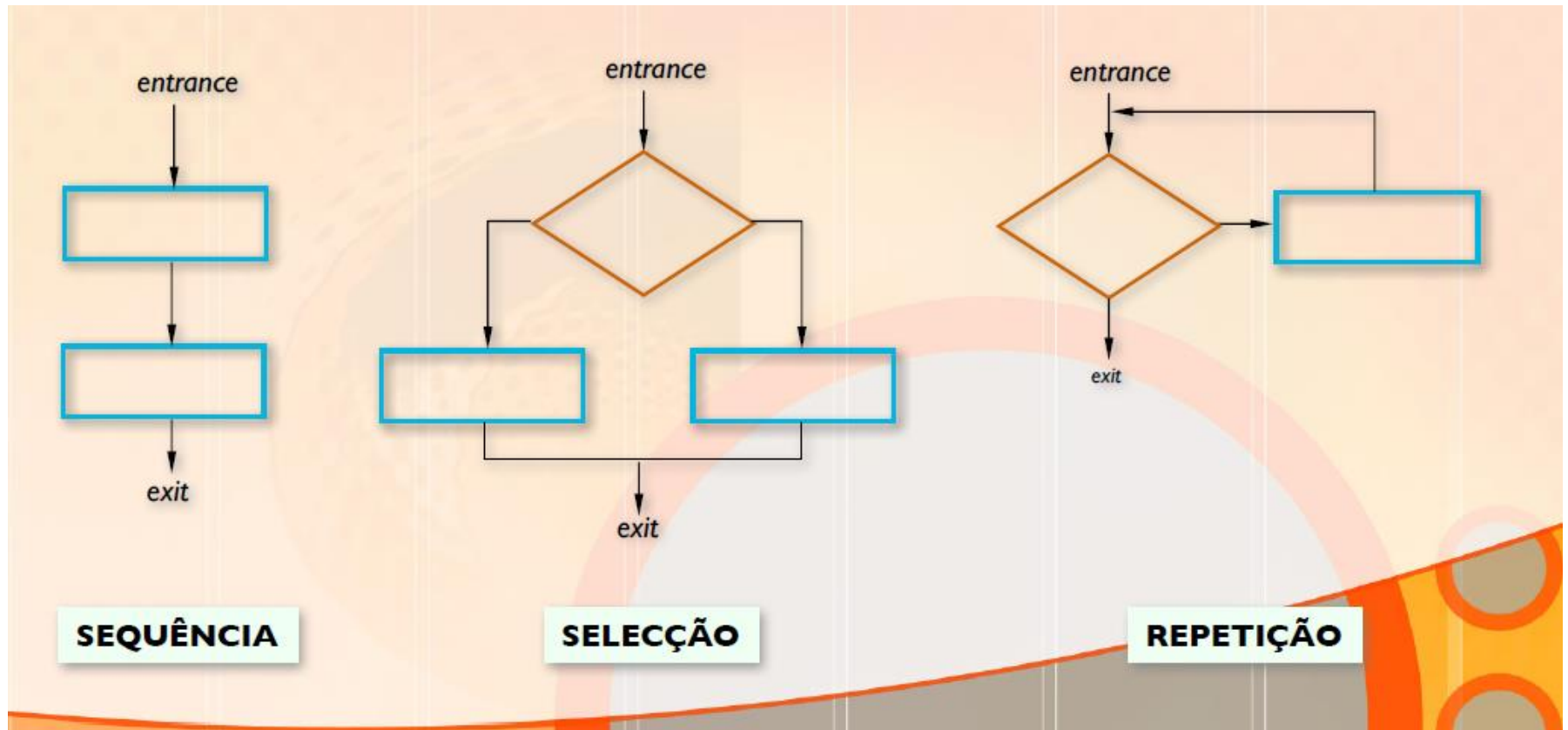
DCT DEPARTAMENTO CIÊNCIA
E TECNOLOGIA

Algoritmos

- **Relembrando...**
- Sequência de passos que nos permitem encontrar o caminho para a solução de um problema
- Dados de entrada, processamento, dados de saída
- Tipos de dados
 - Primitivos: inteiros, reais, strings, etc.
 - Não primitivos: vetores e matrizes
- Algoritmos representados utilizando pseudocódigo e fluxogramas

Estruturas de Programação

■ Relembrando...

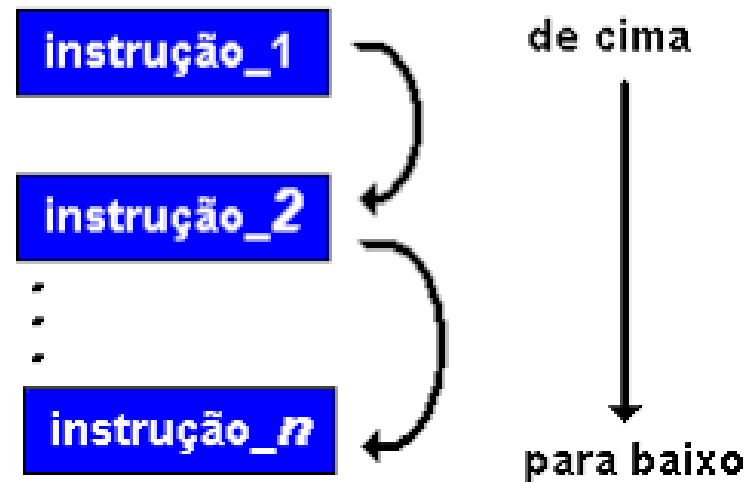


Estruturas de Programação

- **Instruções de Sequência** - as instruções de sequência são instruções atômicas (simples) permitem a leitura/escrita de dados, bem como o cálculo e atribuição de valores;
- **Instruções de Decisão** - as instruções de decisão, ou seleção, permitem a seleção em alternância de um ou outro conjunto de ações após a avaliação lógica de uma condição;
- **Instruções de Repetição** - as instruções de repetição, ou ciclos, permitem a execução, de forma repetitiva, de um conjunto de instruções. Esta execução depende do valor lógico de uma condição que é testada em cada iteração para decidir se a execução do ciclo continua ou termina.

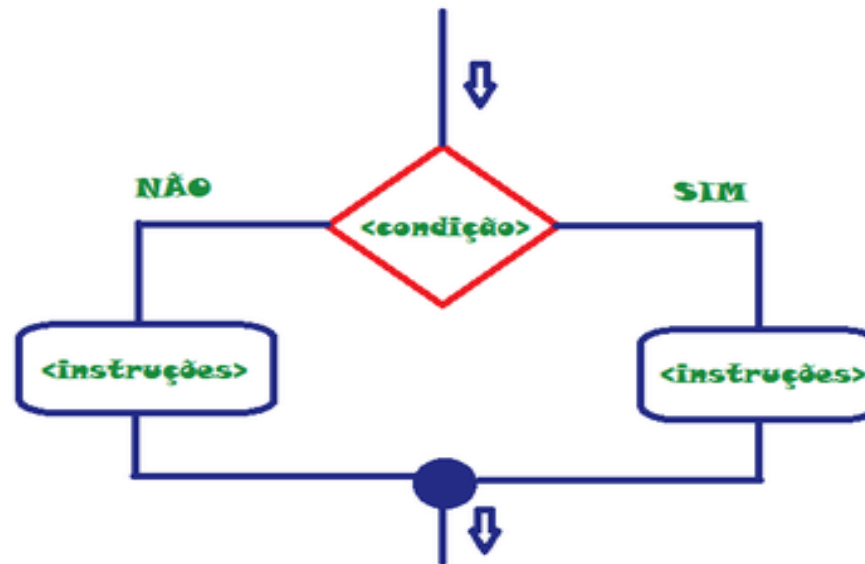
Instruções de Sequência

- Os **passos ou instruções** são executadas **uma de cada vez e não em paralelo**
- **Cada passo** é executado apenas **uma vez**
- A **execução dos passos** é efetuada **na ordem** em que estão escritos
- A execução do **último passo termina** o algoritmo
- As instruções sequenciais **não são flexíveis** uma vez que não podem ser adaptados para responder a diferentes circunstâncias



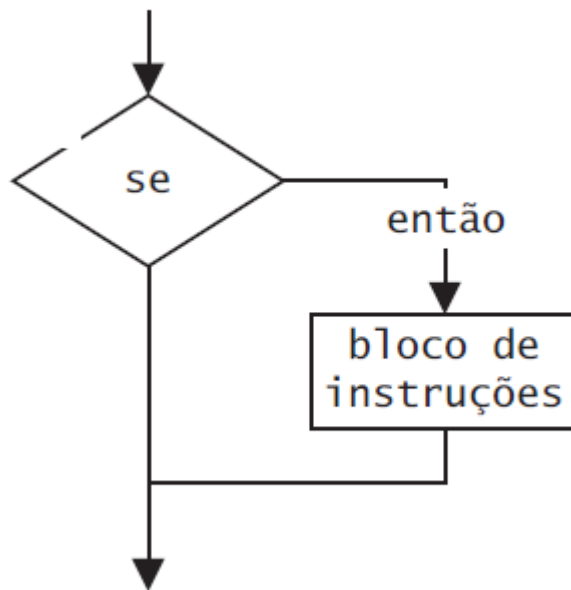
Instruções de Condição

- **Instruções de condição** são estruturas mais avançadas que proporcionam maior flexibilidade
- Possibilitam a **execução de comandos/instruções** de acordo com o **resultado da condição**
- **Simple e composta**



Instruções de Condição Binária

- **Condições simples** permitem dividir a execução do algoritmo em dois fluxos distintos. Esta instrução pode ser utilizada de duas formas *se-então* e *se-então-senão*.



```
inicio
...
se <condição>
então <bloco-instruções>
fim-se
...
fim
```

Instruções de Condição Binária

- Exemplo:

Pedir ao utilizador que introduza a sua classificação em Fundamentos de programação.

O algoritmo deverá ser capaz de mostrar se o utilizador está aprovado!

Instruções de Condição Binária

Algoritmo *Instrução Binária*

Var nota : real

Início

Escrever (“Introduza nota:”)

Ler nota;

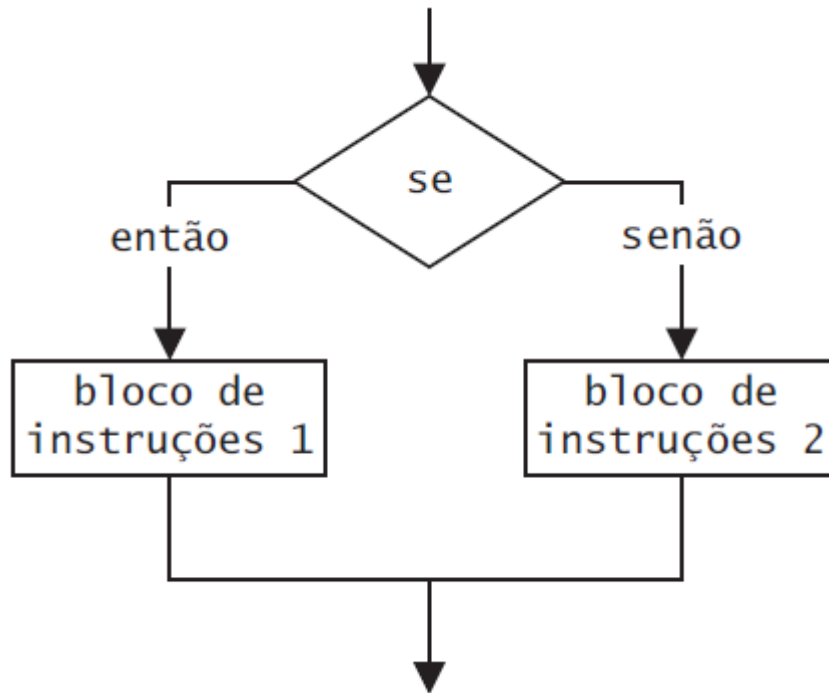
Se (nota \geq 9,5)

Então

Escrever (“O Aluno foi aprovado.”)

Fim

Instruções de Condição Binária



```
inicio  
...  
se <condição>  
então <bloco-instruções1>  
senão <bloco-instruções2>  
fim-se  
...  
fim
```

Instruções de Condição Binária

- Mais um exemplo ...
- **Crie um algoritmo que classifica uma pessoa como maior ou menor de idade**

Algoritmo *idade*

Var idade: inteiro

Início

Escrever (“Introduza a sua idade:”)

Ler idade

Se (idade \geq 18)

Então

Escrever (“Encontra-se na idade adulta.”)

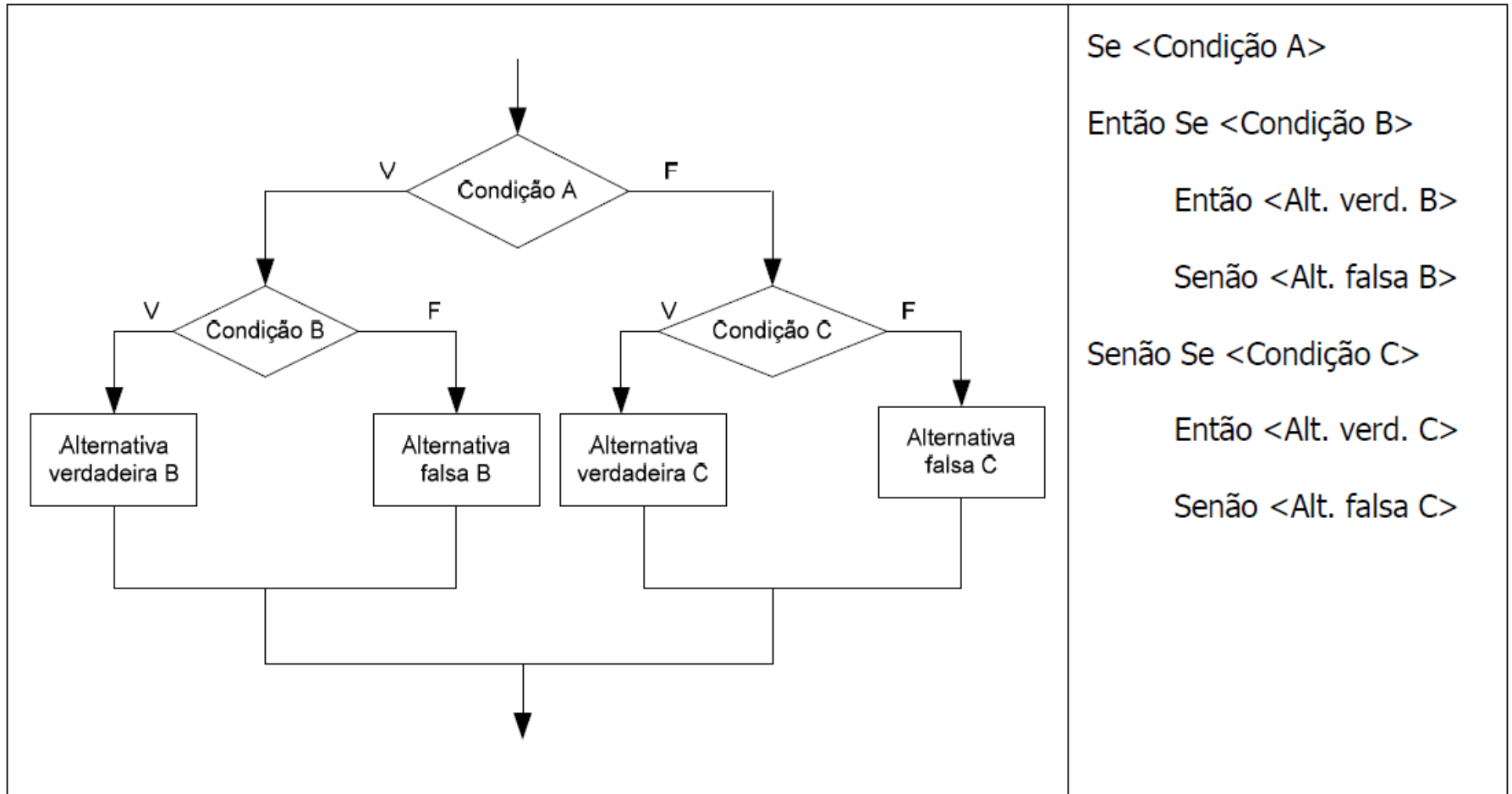
Senão

Escrever (“Menor idade.”)

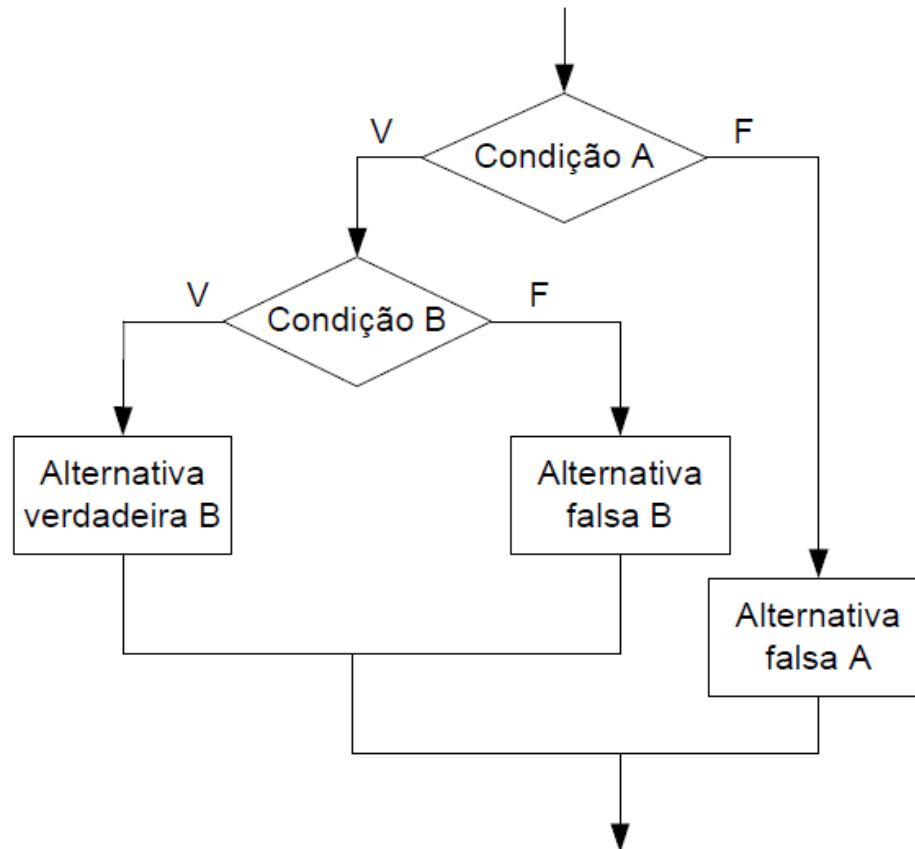
Fim_se

Fim

Instruções de Condição Múltipla



Instruções de Condição Múltipla



Se <Condição A>

Então Se <Condição B>

Então <Alt. verd. B>

Senão <Alt. falsa B>

Senão <Alt. falsa. A>

Instruções de Condição Múltipla

- Vamos tomar como exemplo a **leitura da altura de uma pessoa e a sua correspondente classificação**:
 - **Altura < 160 cm** a pessoa é **baixa**
 - **$160 \leq$ Altura < 175 cm** a pessoa é de **média estatura**
 - **Altura ≥ 175 cm** a pessoa é considerada **alta**

Apenas um
Exemplo

Instruções de Condição Múltipla

Algoritmo *Altura.* Este algoritmo lê a altura de uma pessoa e imprime uma mensagem. O input é um valor inteiro positivo em cm.

Var altura: inteiro

Início

Escrever (“Introduza a altura em cm:”)

Ler altura

Se (altura < 160)

Então Escrever (“Estatura Baixa.”)

Se (altura >= 160 && altura < 175)

Então Escrever (“Estatura Média”)

Se (altura >= 175)

Então Escrever (“Estatura alta”)

Fim

Todos os ses são testados independentemente da condição de cada ser verdadeira ou falsa.

Se embutido ou encadeado

Instruções de Condição Múltipla

Algoritmo *Altura*.

Var altura: inteiro

Início

Escrever (“Introduza a altura em cm:”)

Ler altura;

Se (altura < 160)

Então Escrever (“Estatura Baixa.”)

Senão Se (altura < 175)

Então Escrever (“Estatura Média”)

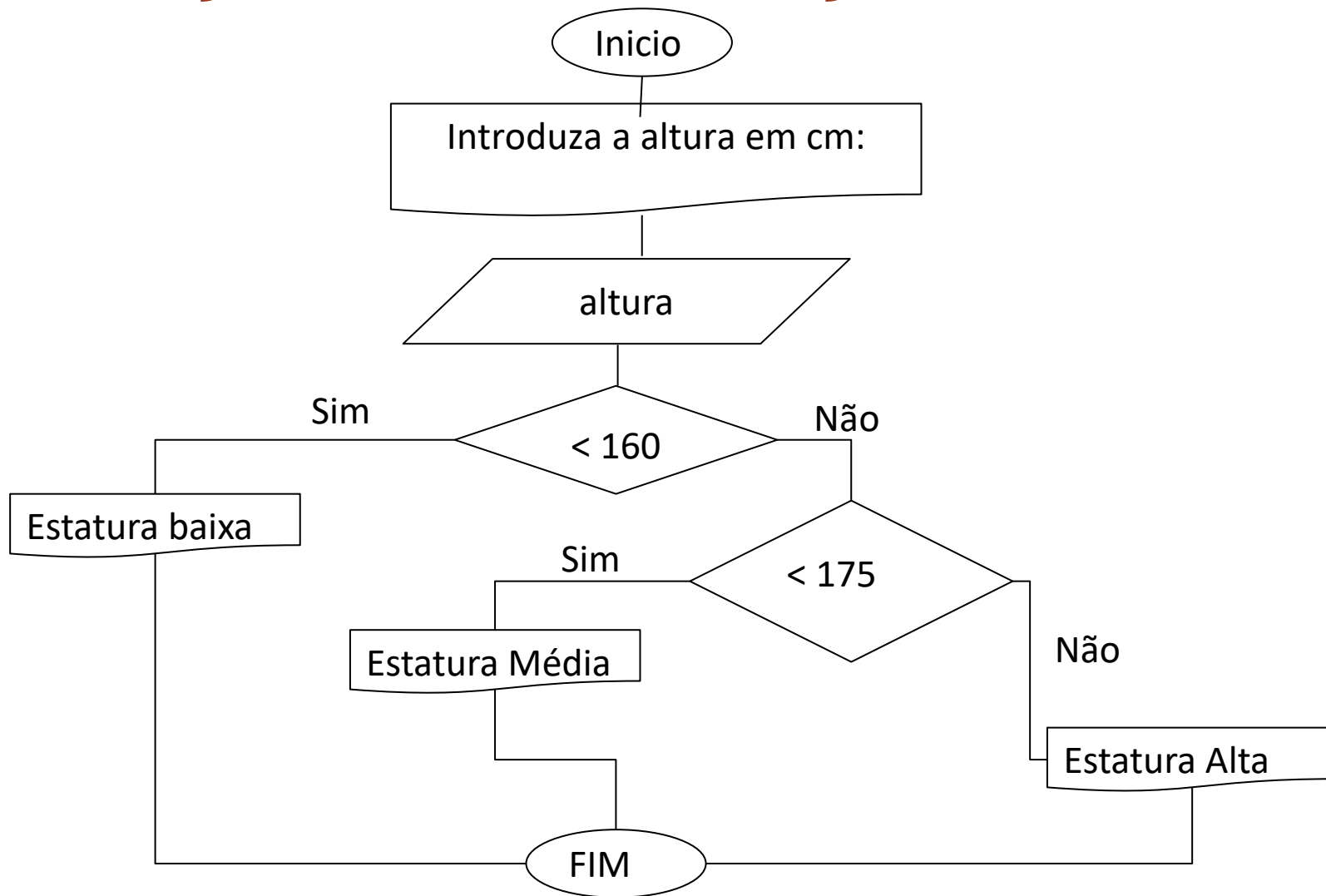
Senão Escrever (“Estatura alta”)

Fim_se

Fim_se

Fim

Instruções de Condição Múltipla



Instruções de Condição Múltipla

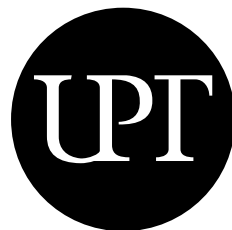
- Mais um exemplo:
- Considere uma máquina que permite apenas três operações, **ligar**, **desligar** e **furar**. Construa um algoritmo que permita modelar o funcionamento da respectiva máquina.
- Quando são digitadas as letras: '**L**', '**D**' e '**F**', são apresentadas, respectivamente, as mensagens: **Ligar**, **Desligar** e **Furar**. No caso da letra digitada ser outra é apresentada uma mensagem de erro.

Instruções de Condição Múltipla

Algoritmo: Máquina

```
início
  # Ler letra;
  escrever "Introduza letra (L/D/F):";
  ler letra;
  # Testar casos e escrever mensagem respectiva;
  se letra='L' então
    | escrever "Ligar";
  senão
    se letra='D' então
      | escrever "Desligar";
    senão
      se letra='F' então
        | escrever "Furar";
      senão
        | escrever "Operação inválida";
      fim-se
    fim-se
  fim-se
fim
```

Solução com a abordagem do se encadeado



UNIVERSIDADE
PORTUCALENSE

Do conhecimento à prática.

Instruções de Condição

Desenvolva um algoritmo que peça ao utilizador dois números.
O algoritmo deverá verificar qual o maior valor.

Algoritmo *Exemplo2***Var** num1, num2, maior : inteiro**Início****Ler** (num1, num2)**Se** num1==num2 **então**

Escrever("Iguais")

Senão Se (num1 > num2) **então**

maior <- num1

Senão se (num1 < num2)

maior <- num2

fim_se **fim_se****Fim_se****Escrever** (maior)**Fim**

Fluxogramas: Exemplo 2

