

# Estrutura de Dados 1

---

Prof. Igor Calebe Zadi  
igor.zadi@ifsp.edu.br



**INSTITUTO FEDERAL**  
São Paulo  
Campus Catanduva



# I. Fundamentos de Estruturas de Dados



# I. Fundamentos de Estruturas de Dados

1. Definições
2. Classificação das Estruturas de Dados
3. Programação Orientada a Procedimentos
4. Tipos de dados primitivos
5. Cadeias de caracteres
6. Registros
7. Ponteiros

# 1. Definições



# 1. Definições

- Algoritmo
  - Conjunto **finito de passos**, processos ou instruções lógicas necessárias **à resolução de um problema**
- Estrutura de Dados (ED)
  - Categoria de dados que se caracteriza por sua **organização** e pela sua **álgebra**, isto é, pelo conjunto de **operações** definidas sobre ela.



# 1. Definições

- De modo geral

**Programas = Algoritmos + ED**

- As ED são importantes para a manipulação de dados dentro dos programas



# 1. Definições

- Variável
  - área de memória que armazena valores utilizados durante a execução do programa
  - valores podem sofrer alteração ao longo do tempo
- Toda variável pertence a alguma ED
  - isso determina o conjunto de operações definidas sobre a variável



# 1. Definições

- Conjunto de valores possíveis para uma variável
  - depende da ED à qual ela pertence
  - depende também da implementação da **linguagem de programação** utilizada
    - C para Windows: **int** tem 2 bytes
    - C para Unix: **int** tem 4 bytes



## 2. Classificação das ED

## 2. Classificação das ED

- Classificação geral

PRIMITIVAS	COMPOSTAS		
	SIMPLES	LINEARES	NÃO-LINEARES
Inteiro	Cadeia	Lista	Árvore
Real	Registro	Pilha	Grafo
Lógico	Arranjo	Fila	
Caracter			



## 2. Classificação das ED

- Quanto à homogeneidade dos dados

Tipo	Característica	Exemplo
<b>Homogênea</b>	Os elementos armazenados são todos do mesmo tipo	Cadeia de caracteres Arranjo
<b>Heterogênea</b>	Os elementos armazenados não são obrigatoriamente todos do mesmo tipo	Registro



## 2. Classificação das ED

- Quanto à linearidade

Tipo	Característica	Exemplo
Linear	ED tem como regra preservar uma relação de ordem qualquer entre os dados	Arranjo  Lista linear
Não-linear	ED não tem como regra preservar uma relação de ordem qualquer entre os dados	Árvore  Grafo



## 2. Classificação das ED

- Quanto à forma de alocação

Tipo	Característica	Exemplo
<b>Estática</b>	ED tem tamanho fixo (ocupa sempre a mesma porção de memória)	Registro
<b>Dinâmica</b>	ED (quantidade de dados) aumenta ou diminui durante a execução do programa	Lista encadeada

# 3. Programação Orientada a Procedimentos



## 2. Classificação das ED

- Quanto à forma de alocação

Tipo	Característica	Exemplo
<b>Estática</b>	ED tem tamanho fixo (ocupa sempre a mesma porção de memória)	Registro
<b>Dinâmica</b>	ED (quantidade de dados) aumenta ou diminui durante a execução do programa	Lista encadeada



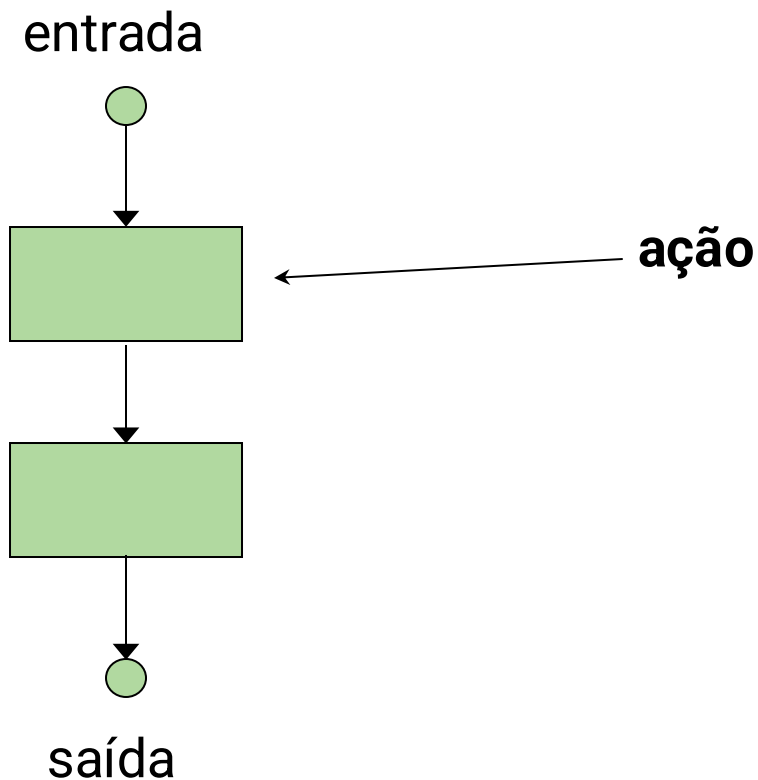
# 3. Programação Orientada a Procedimentos

- Programação estruturada
  - apenas 3 tipos de estruturas de controle são necessárias em um programa:
    - sequência
    - Seleção
    - repetição
- Orientação a procedimentos
  - o programa é dividido em programas menores, cada um com finalidade bem determinada
  - “dividir para conquistar”



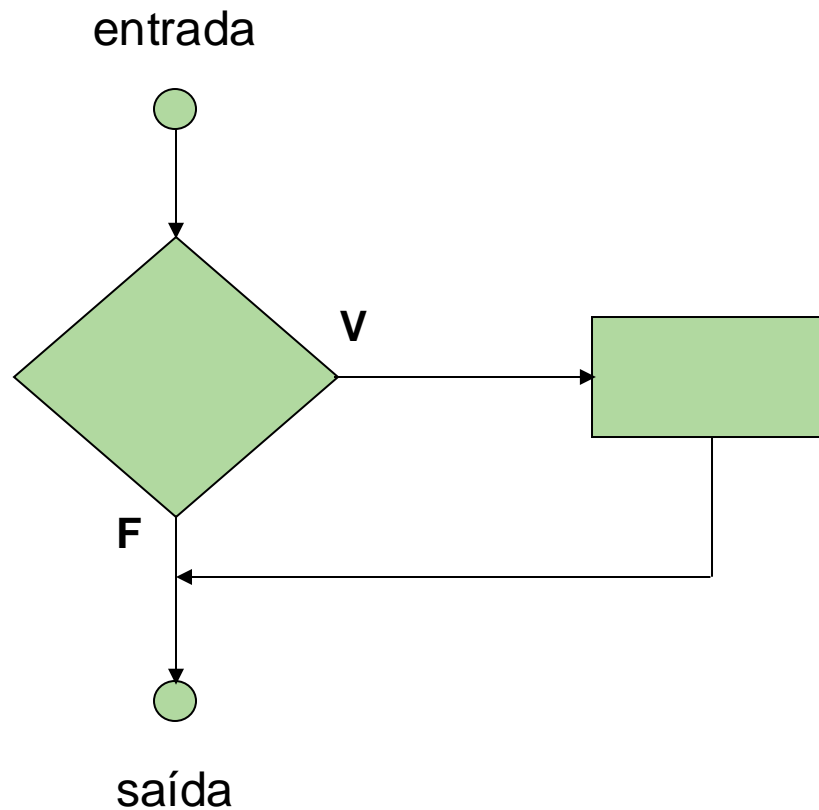
### 3. Programação Orientada a Procedimentos

- Estrutura de controle: sequência



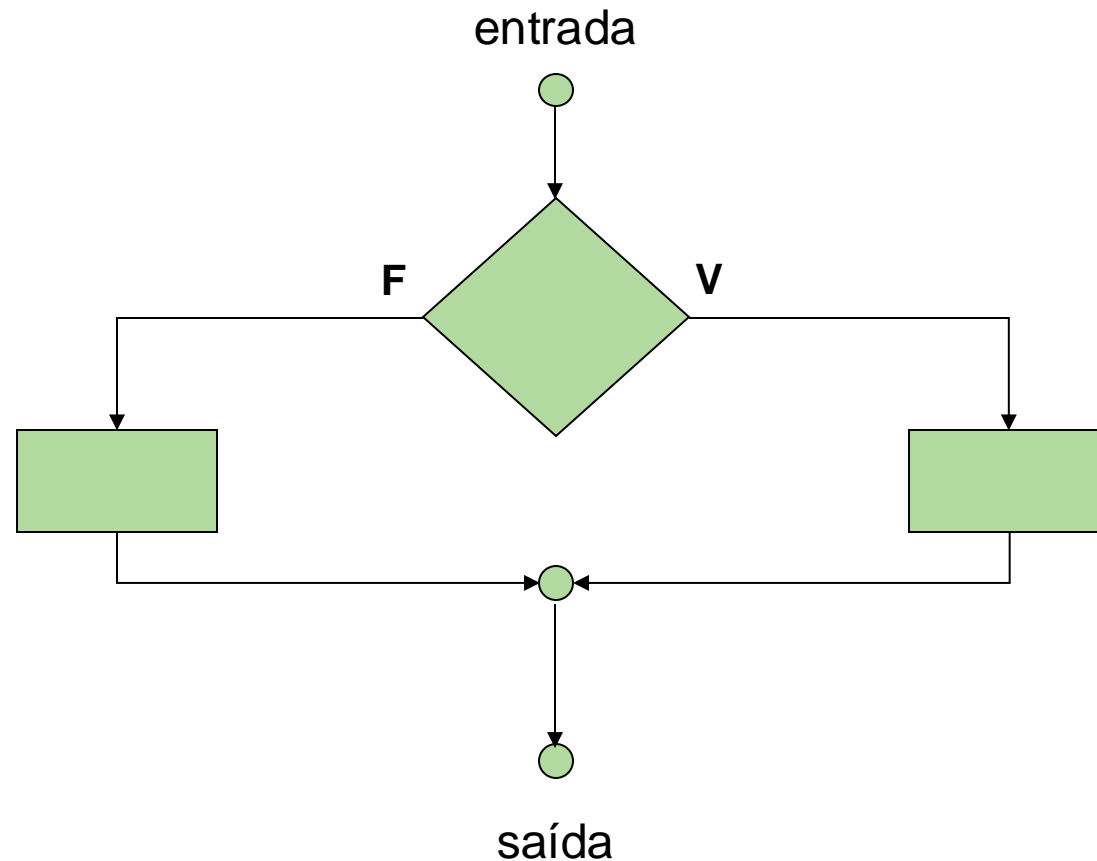
### 3. Programação Orientada a Procedimentos

- Estrutura de controle: seleção única



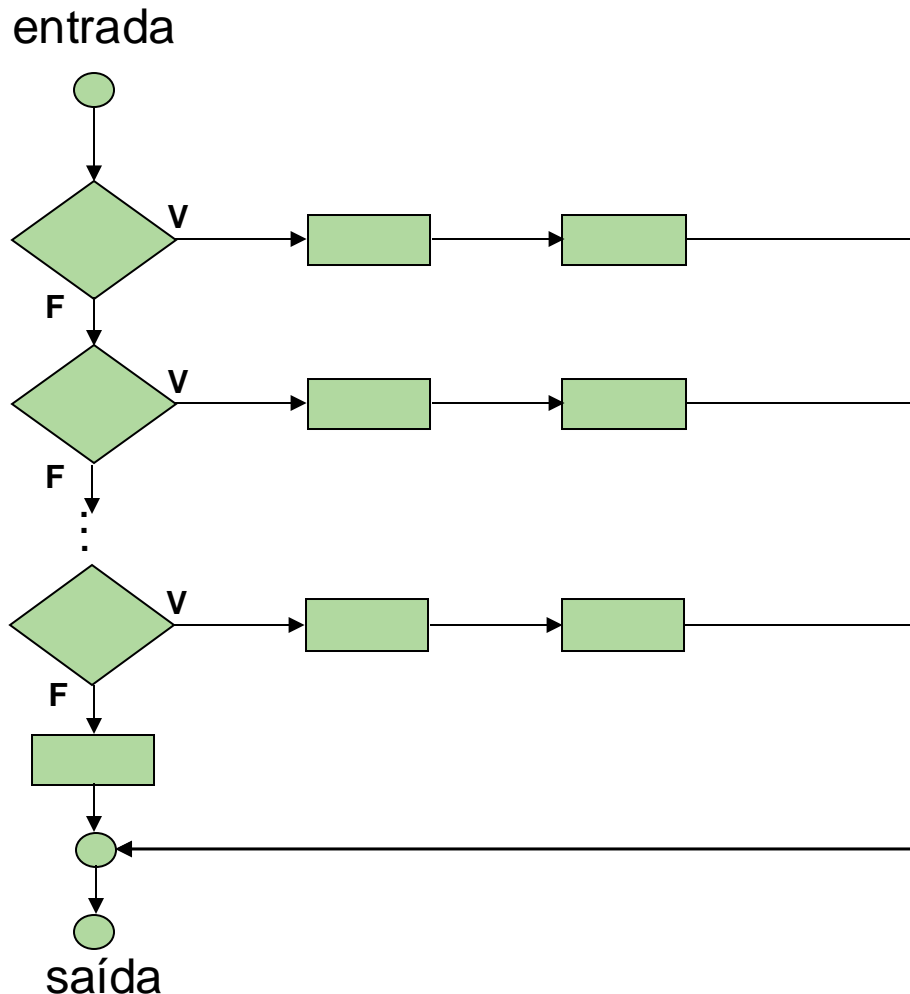
### 3. Programação Orientada a Procedimentos

- Estrutura de controle: seleção dupla



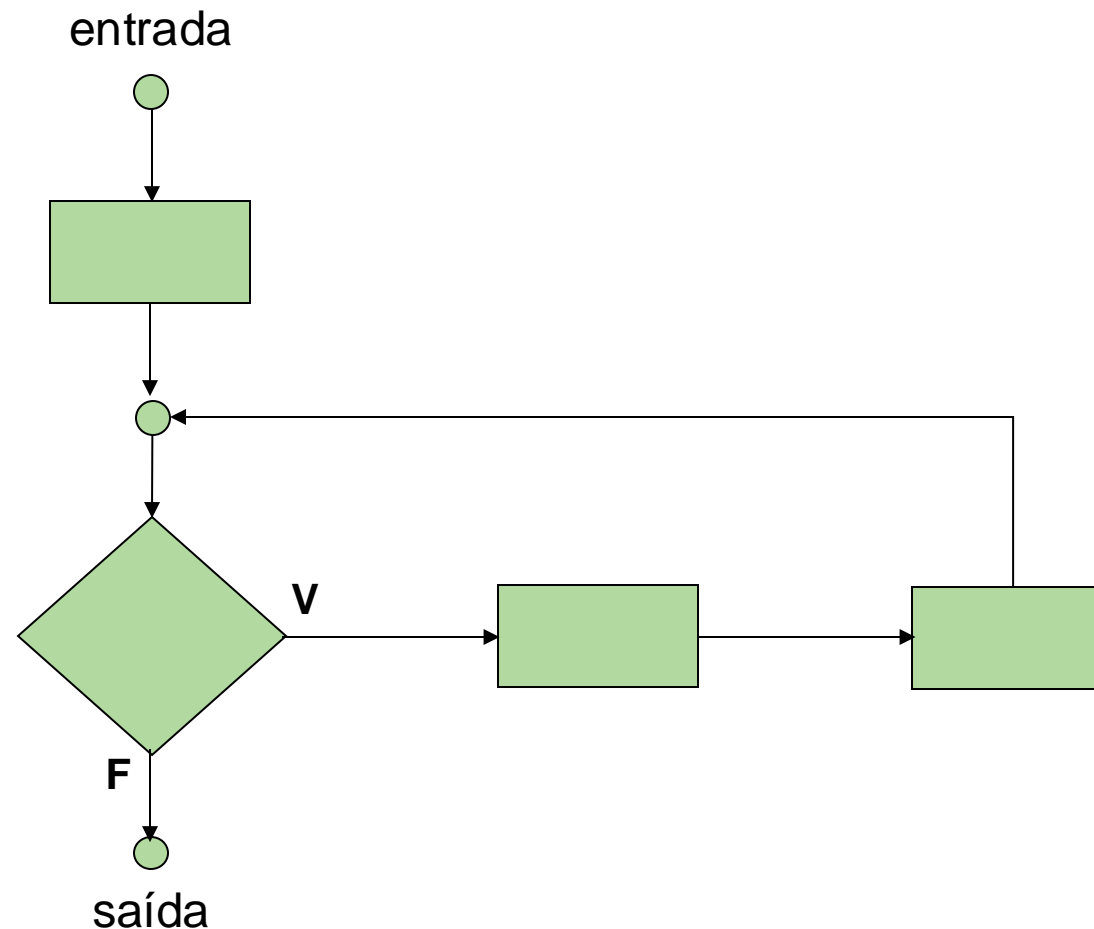
### 3. Programação Orientada a Procedimentos

- Estrutura de controle: seleção múltipla



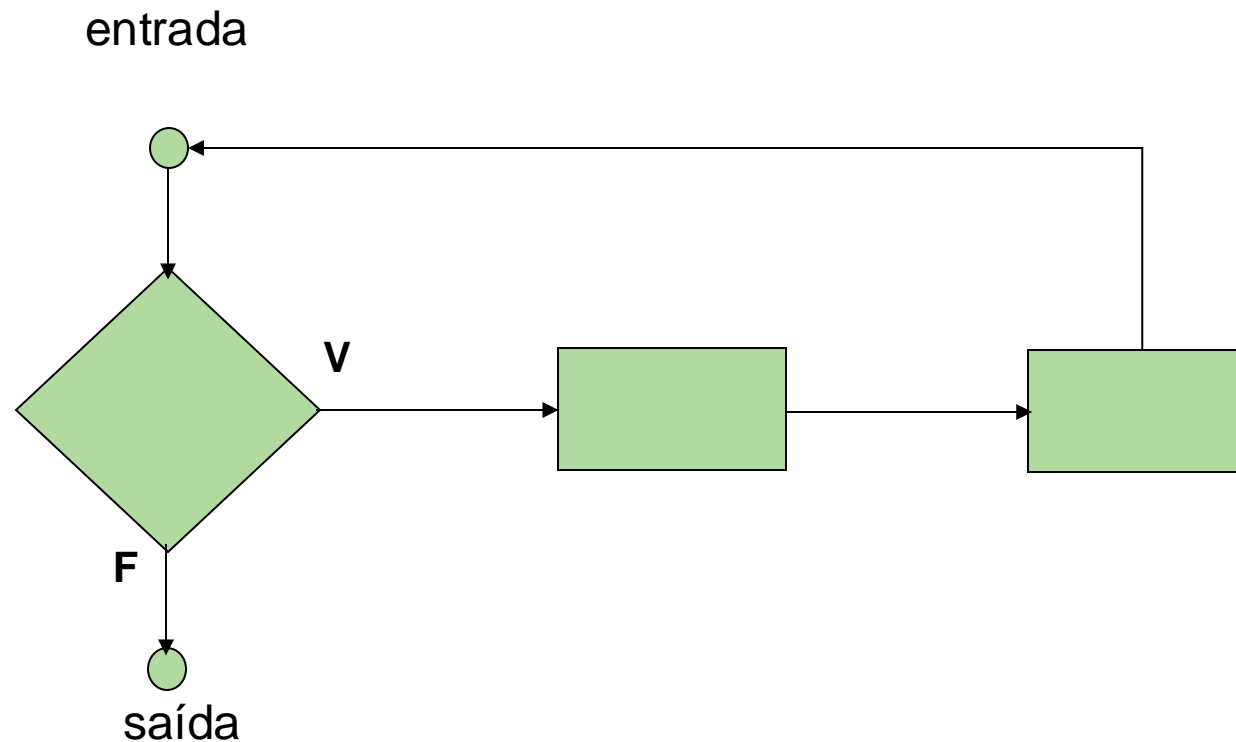
### 3. Programação Orientada a Procedimentos

- Estrutura de repetição do tipo **for**



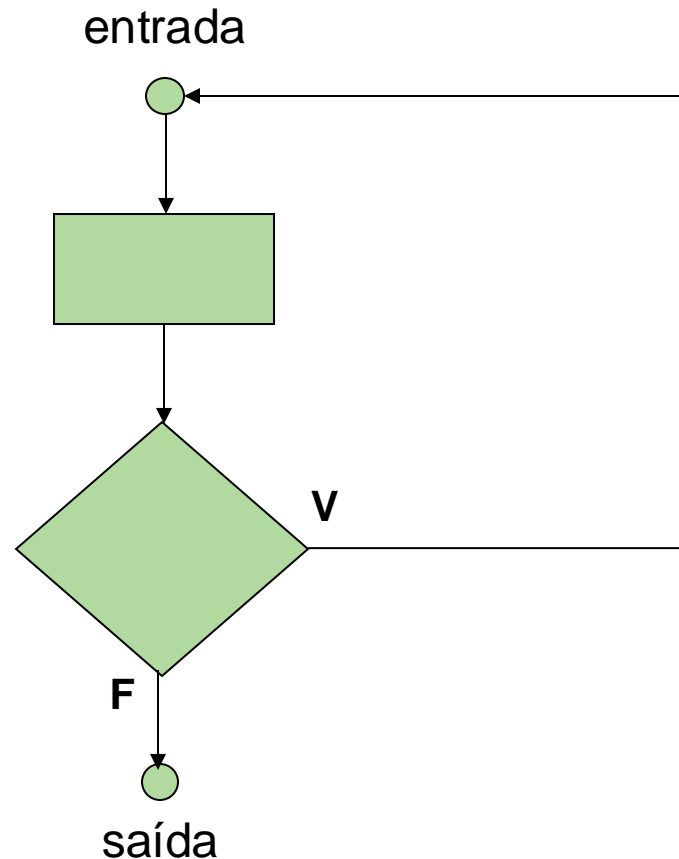
### 3. Programação Orientada a Procedimentos

- Estrutura de repetição do tipo **while**



### 3. Programação Orientada a Procedimentos

- Estrutura de repetição do tipo **do.. while**





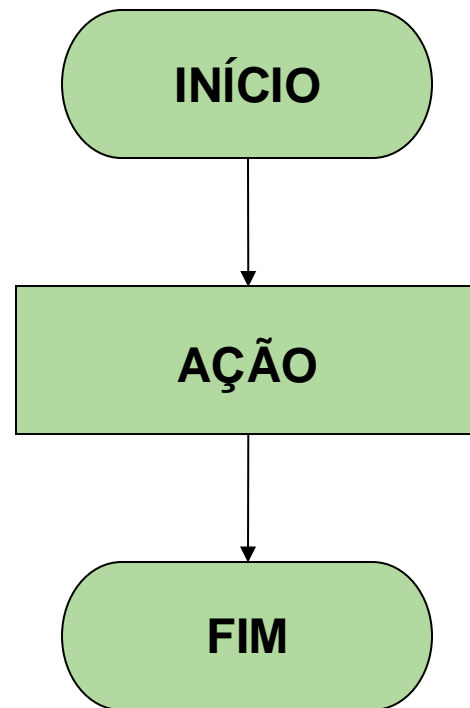
### 3. Programação Orientada a Procedimentos

- Regra para obter programas estruturados
  1. Começar com o fluxograma mais simples
  2. Qualquer retângulo pode ser substituído por dois retângulos em sequência (**empilhamento**)
  3. Qualquer retângulo pode ser substituído por qualquer estrutura de controle (**aninhamento**)
  4. As regras 2 e 3 podem ser aplicadas a qualquer momento e em qualquer ordem



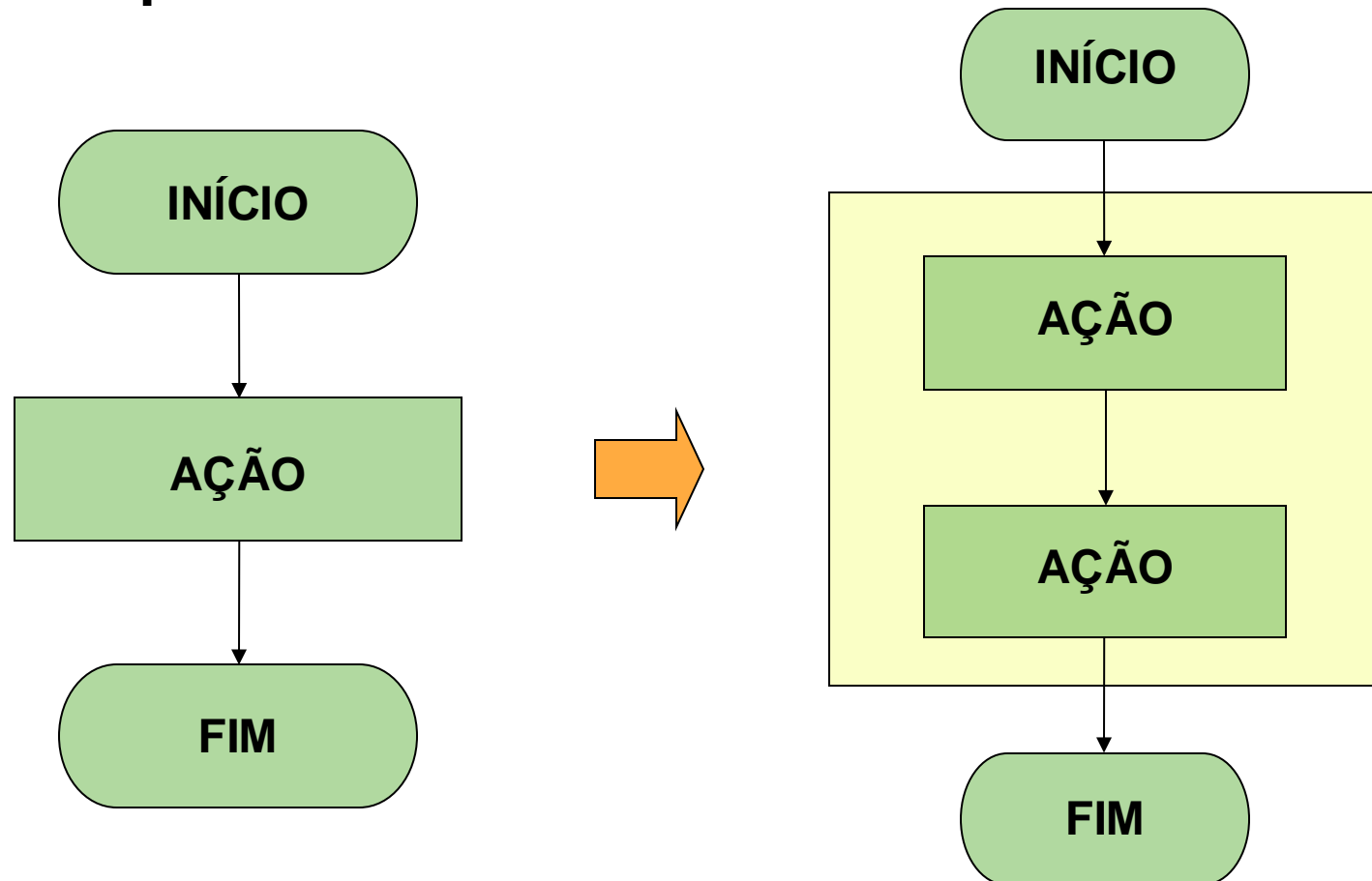
### 3. Programação Orientada a Procedimentos

- **Regra 1:** fluxograma mais simples



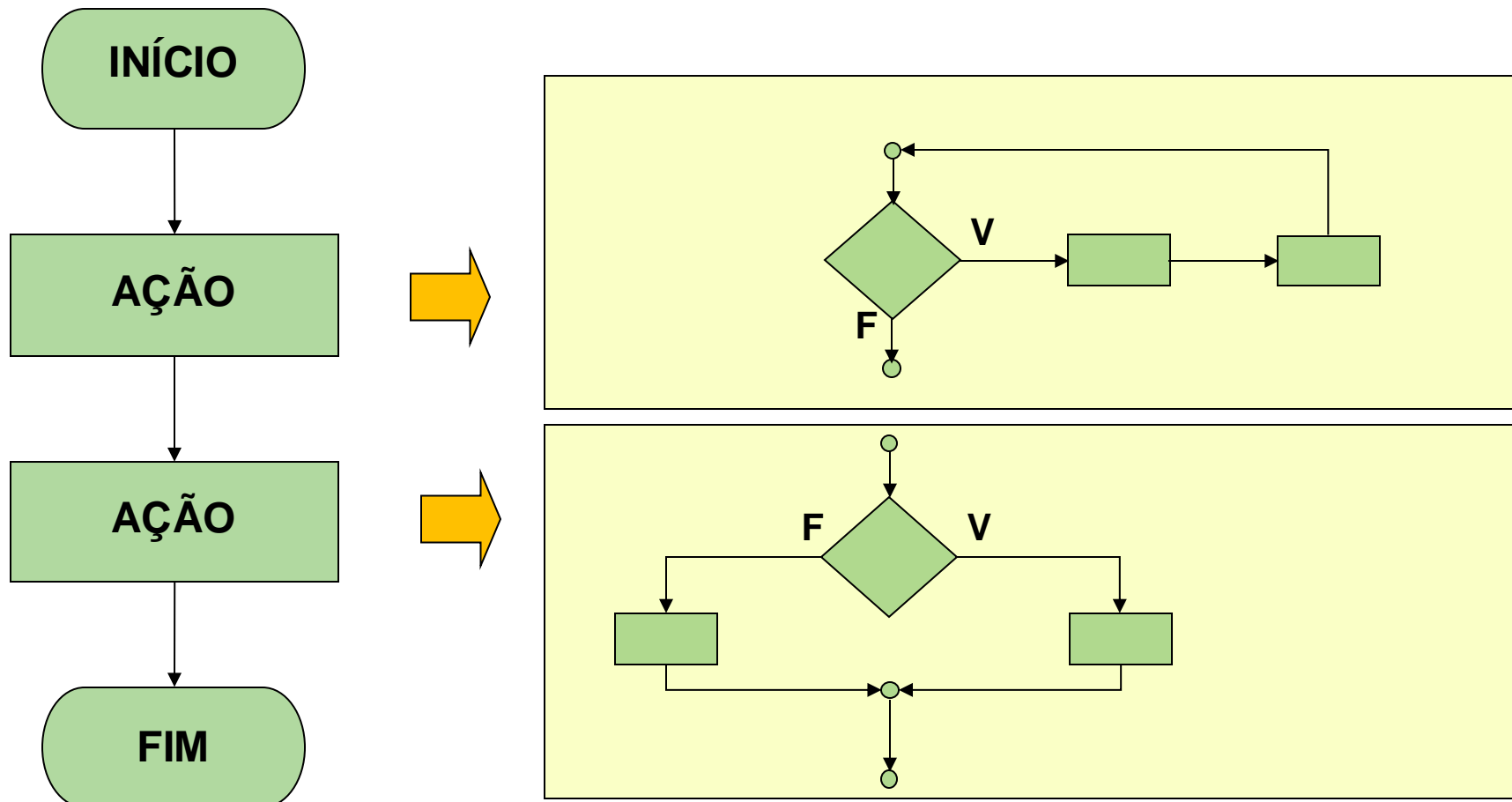
### 3. Programação Orientada a Procedimentos

- **Regra 2:** empilhamento



# 3. Programação Orientada a Procedimentos

- **Regra 3:** aninhamento





# Observações sobre o material eletrônico

- O material ficará disponível na pasta compartilhada que é acessada sob convite
- O material foi elaborado a partir de diversas fontes (livros, internet, colegas, alunos etc.)
- Alguns trechos podem ter sido inteiramente transcritos a partir dessas fontes
- Outros trechos são de autoria própria
- Esta observação deve estar presente em qualquer utilização do material fora do ambiente de aulas do IFSP - Catanduva