

Etapa 4 — Estruturas Condicionais (Conditional Structures)

Guia Completo de Lógica de Programação

1. Introdução

As estruturas condicionais são blocos de decisão que permitem que um algoritmo **reaja a diferentes situações**.

Elas determinam **quais instruções devem ser executadas** com base em uma **condição lógica**, que pode ser **verdadeira (true)** ou **falsa (false)**.

Sem decisões, os programas seriam apenas **sequências lineares de comandos**. Com elas, o algoritmo pode **avaliar opções e escolher o caminho adequado**.

Exemplo prático:

Se o usuário tiver mais de 18 anos, pode entrar no sistema.
Caso contrário, a entrada é negada.

2. Tipos de Estruturas Condicionais

As estruturas condicionais podem ser classificadas em três tipos principais:

1. Simples (Simple Condition)

- Executa uma ação **apenas se** a condição for verdadeira.
- Caso contrário, o programa **segue em frente** sem executar nada.

2. Composta (Composite Condition)

- Executa uma ação **se a condição for verdadeira, e outra ação** se for falsa.

3. Encadeada (Nested Condition)

- Usa **várias condições dentro de outras**, formando uma cadeia de decisões.

3. Estrutura Condicional Simples (Simple If)

Sintaxe:

```
se (condicao) entao  
    // comandos executados se for verdadeiro  
fimse
```

Exemplo:

```
inicio  
    leia idade  
    se idade >= 18 entao  
        escreva "Maior de idade"  
    fimse  
fim
```

Saída:

“Maior de idade” (apenas se a condição for verdadeira)

4. Estrutura Condicional Composta (If-Else)

Sintaxe:

```
se (condicao) entao  
    // comandos se for verdadeiro  
senao  
    // comandos se for falso  
fimse
```

Exemplo:

```
inicio  
leia nota  
se nota >= 6 entao  
    escreva "Aprovado"  
senao  
    escreva "Reprovado"  
fimse  
fim
```

Saída:

Mostra “Aprovado” se a nota for 6 ou maior, caso contrário, “Reprovado”.

5. Estrutura Condicional Encadeada (Nested If)

Permite testar **várias condições em sequência**.

Cada condição é avaliada **somente se as anteriores forem falsas**.

Exemplo:

```
inicio  
leia nota  
se nota >= 9 entao  
    escreva "Excelente"  
senao  
    se nota >= 6 entao  
        escreva "Aprovado"  
    senao  
        escreva "Reprovado"  
fimse
```

fimse

fim

Saída:

Classifica o desempenho do aluno em três categorias: Excelente, Aprovado ou Reprovado.

6. Condições Compostas e Operadores Lógicos

As condições podem ser combinadas com **operadores lógicos (logical operators)**:

Operador	Nome	Resultado
e	AND	Verdadeiro se ambas forem verdadeiras
ou	OR	Verdadeiro se pelo menos uma for verdadeira
nao	NOT	Inverte o valor lógico

Exemplo:

inicio

leia idade, carteira

se idade >= 18 e carteira = verdadeiro entao

 escreva "Pode dirigir"

senao

 escreva "Não pode dirigir"

fimse

fim

Saída:

“Pode dirigir” se a pessoa for maior de idade **e** tiver carteira.

7. Estruturas Condicionais Aninhadas (Nested Conditions)

Usadas quando há **duas verificações dependentes**.

Exemplo:

inicio

leia idade, nacionalidade

se idade >= 18 entao

 se nacionalidade = "Brasileiro" entao

 escreva "Pode votar"

 fimse

fimse

fim

Saída:

“Pode votar” somente se as duas condições forem verdadeiras.

8. Exemplo Prático Completo

algoritmo "sistema_desconto"

inicio

leia valor

se valor >= 100 entao

 desconto ← valor * 0.1

senao

 desconto ← 0

fimse

total ← valor - desconto

escreva "Total a pagar:", total

fim

Saída:

Se o valor for 150, o total será 135 (10% de desconto).

Se for 80, o total será 80 (sem desconto).

9. Cuidados e Boas Práticas

- Use **indentação (indentation)** para deixar o código legível.
Sempre feche os blocos com `fimse`.
Teste **todas as possibilidades** (verdadeiro e falso).
Use **parênteses** em expressões complexas.
Evite condições redundantes.

Erros comuns:

- Esquecer o `fimse` → causa confusão no bloco.
- Usar `=` ao invés de `==` para comparar valores.
- Misturar tipos de dados (ex: texto com número).

10. Exercício Extra

Problema:

Escreva um algoritmo que leia a idade de uma pessoa e informe:

- “Criança” se idade < 12
- “Adolescente” se idade entre 12 e 17
- “Adulto” se idade ≥ 18

Solução:

inicio

```
leia idade  
se idade < 12 entao  
    escreva "Criança"  
senao  
    se idade < 18 entao  
        escreva "Adolescente"  
    senao  
        escreva "Adulto"  
fimse  
fimse  
fim
```

11. Conclusão

Nesta etapa, aprendemos que:

- As estruturas condicionais tornam o algoritmo **inteligente**.
- Permitem **escolhas lógicas e decisões automáticas**.
- São a base para o controle de fluxo em qualquer programa.
- Usar **se**, **senao** e operadores lógicos é essencial para o domínio da lógica de programação.

Na próxima etapa, estudaremos as **Estruturas de Repetição (Loops)**, que permitem **automatizar tarefas repetitivas**.

12. Exercícios de Fixação (Multiple Choice)

1. Qual estrutura executa uma ação apenas se a condição for verdadeira?
 - a) composta
 - b) simples
 - c) encadeada
 - d) aninhada

2. Qual operador lógico retorna verdadeiro se **ambas** as condições forem verdadeiras?

- a) ou (OR)
- b) nao (NOT)
- c) e (AND)
- d) senao

3. Em um pseudocódigo, o que acontece se a condição for falsa em um **se...senao**?

- a) Nada acontece
- b) O bloco **senao** é executado
- c) O programa é encerrado
- d) O código entra em loop

4. O que significa uma estrutura condicional **encadeada**?

- a) Uma estrutura com várias condições internas
- b) Um comando que repete instruções
- c) Um tipo de operador aritmético
- d) Um erro de sintaxe

Gabarito

1. **b)** simples
2. **c)** e (AND)
3. **b)** executa o bloco **senao**
4. **a)** várias condições internas