Etapa 4 — Estruturas Condicionais (Conditional Structures)

Trilha: Lógica de Programação

O que são Estruturas Condicionais?

- Permitem tomar decisões dentro de um algoritmo.
- Fazem com que o programa **escolha caminhos diferentes** conforme condições sejam **verdadeiras (true)** ou **falsas (false)**.
- São essenciais para que o programa **“pense” e reaja** a diferentes situações.

Exemplo prático:

  - Se o usuário digitar uma senha incorreta, o sistema exibe “Acesso negado”.

Tipos de Estruturas Condicionais

1. **Simples (Simple Condition)** — executa algo **apenas se** a condição for verdadeira.
2. **Composta (Composite Condition)** — executa uma ação **se for verdadeiro** e outra **se for falso.**
3. **Encadeada (Nested Condition)** — combina **várias condições**, uma dentro da outra.

Estrutura Condicional Simples (Simple If)

- **Sintaxe em pseudocódigo:**

```
se (condicao) entao
  // ações executadas se for verdadeiro
fimse
```

- **Exemplo:**

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

- **Saída:**

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

Estrutura Condicional Composta (If-Else)

- **Sintaxe:**

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

- **Exemplo:**

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

- **Saída:**

“Aprovado” ou “Reprovado”, dependendo da nota.

Estrutura Condicional Encadeada (Nested If)

- **Sintaxe:**

```
se (condicao1) entao
    ...
senao
    se (condicao2) entao
        ...
    senao
        ...
    fimse
fimse
```

- **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 conforme a nota.

Estruturas Condicionais Aninhadas (Nested Conditions)

- Uma **condição dentro de outra**.
- Usada para **verificações duplas** (ex: idade e nacionalidade).
- Requer **atenção com a identação** (indentation) para manter o código legível.
- **Exemplo:**

```
inicio
    leia idade, nacionalidade
    se idade >= 18 entao
        se nacionalidade = "Brasileiro" entao
            escreva "Pode votar"
        fimse
    fimse
fim
```

Estrutura Condicional com Operadores Lógicos

- Permite **combinar condições** usando **e (AND)**, **ou (OR)** e **nao (NOT)**.
- **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” apenas se ambas as condições forem verdadeiras.

Dica Prática

- Sempre use **parênteses** para deixar clara a ordem das condições.
- Use **comentários** (//) para explicar blocos de decisão.
- Evite **condições duplicadas** ou desnecessárias.
- **Exemplo:**

```
inicio
    // Verifica se número é par
    leia num
    se (num % 2 = 0) entao
        escreva "Número par"
    senao
        escreva "Número ímpar"
    fimse
fim
```

Exemplo Completo: Sistema de Desconto

```
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:**

Exibe o valor com ou sem desconto, conforme o caso.

Erros Comuns

- Usar = ao invés de == para comparar.
- Esquecer o fimse ao encerrar o bloco.
- Condições mal indentadas dificultam leitura.
- Usar operadores lógicos de forma confusa.
- Dica: Sempre **teste as condições** com diferentes valores de entrada.

Conclusão

- As estruturas condicionais tornam o algoritmo **inteligente e adaptável**.
- O comando se...entao...senao é a base de qualquer **decisão lógica**.
- Entender bem essa etapa é essencial antes de passar para **estruturas de repetição (loops)**.