

[Clique aqui para ler pelo PDF e melhorar a experiência](#)

## Semana 4: Estruturas Condicionais (If-Else)

```
int x = 1;

if (x > 3) {
    printf("x is greater than 3\n");
} else {
    printf("x is less than or equal to 3\n");
}
```

Nesta aula, vamos abordar as **estruturas condicionais** na linguagem C. As estruturas condicionais são usadas para tomar decisões no código, ou seja, fazer com que o programa execute ações diferentes dependendo de certas condições.

### Operadores Relacionais e Lógicos

Antes de entender como usar as estruturas condicionais, é essencial conhecer os operadores relacionais e lógicos, pois eles são usados nas condições das instruções `if`.

#### Operadores Relacionais

Os operadores relacionais comparam dois valores e retornam `true` (1) ou `false` (0). Aqui estão os operadores mais comuns:

- `==`: igual a
- `!=`: diferente de
- `>`: maior que
- `<`: menor que
- `>=`: maior ou igual a
- `<=`: menor ou igual a

#### Operadores Lógicos

Os operadores lógicos são usados para combinar várias condições. Os operadores lógicos mais comuns em C são:

- `&&`: E lógico (true se ambas as condições forem verdadeiras)
- `||`: OU lógico (true se pelo menos uma condição for verdadeira)
- `!`: NÃO lógico (inverte o valor da condição)

### Introdução à Lógica Condicional

#### Estrutura Condicional Simples: `if`

A estrutura `if` é usada para verificar uma condição. Se a condição for verdadeira (`true`), o bloco de código dentro do `if` será executado.

Exemplo de `if` simples:

```
#include <stdio.h>

int main() {
    int numero = 5;

    if (numero > 0) {
        printf("O numero eh positivo.\n");
    }

    return 0;
}
```

Exemplo para comparar uma variável a um valor:

```
#include <stdio.h>

int main() {
    int numero = 5;

    if (numero == 2) {
        printf("O numero eh igual a 2.\n");
    }

    return 0;
}
```

Exemplo de condicional com operador lógico:

```
#include <stdio.h>

int main() {
    int numero = 5;

    if (numero > 0 && numero == 5) { // Se o número é positivo e igual a 5
        printf("O numero eh positivo e igual a 5.\n");
    }

    return 0;
}
```

### Estrutura Condicional Composta: `if-else`

O `if-else` permite executar um bloco de código se a condição for verdadeira e outro bloco se for falsa.

Exemplo de `if-else`:

```
#include <stdio.h>

int main() {
    int numero = -3;

    if (numero > 0) {
        printf("O numero eh positivo.\n");
    } else {
        printf("O numero nao eh positivo.\n");
    }

    return 0;
}
```

No código acima, caso o número não seja positivo (maior do que 0) o programa entrará direto no corpo de execução do else. Resultando no print "O numero nao eh positivo".

### Estrutura Condicional Aninhada: **if-else if**

Você pode usar o **else if** para verificar múltiplas condições. O programa avalia cada condição na ordem e executa o bloco de código da primeira condição verdadeira.

Exemplo de else-if:

```
#include <stdio.h>

int main() {
    int numero = 0;

    if (numero > 0) {
        printf("O número é positivo.\n");
    } else if (numero < 0) {
        printf("O número é negativo.\n");
    } else {
        printf("O número é zero.\n");
    }

    return 0;
}
```

O programa primeiro verificará se a variável número é maior que 0, se for, ele executará o print e não executará nem o else-if nem o else. Caso ele não seja maior que 0 será verificado agora se é menor que 0, se for executará o print e não executará o else. Caso o número não seja maior ou menor que 0, então ele só pode ser 0.

### Aplicações Práticas de Condicionais

Agora, vamos ver alguns exemplos práticos de como usar estruturas condicionais em programas do dia a dia.

## 1) Verificar se um número é positivo, negativo ou zero

Este programa pede ao usuário um número e verifica se ele é positivo, negativo ou zero.

Exemplos de execução:

```
Digite um numero: 5
O numero eh positivo.
```

```
Digite um numero: -5
O numero eh negativo.
```

```
Digite um numero: 0
O numero eh zero.
```

## 2) Cálculo de Descontos com Base em Faixas de Valores

Neste exemplo, vamos calcular o desconto de um produto com base em seu valor. O desconto será aplicado de acordo com a faixa de preço:

- Valores acima de 100 têm 10% de desconto
- Valores entre 50 e 100 têm 5% de desconto
- Valores abaixo de 50 não têm desconto

Exemplos de execução:

Valores acima de 100

```
Digite o preco do produto: 110.5
O valor final com desconto eh: 99.45
```

Valores entre 50 e 100

```
Digite um numero: 100.0
O valor final com desconto eh: 95.00
```

Valores abaixo de 50

```
Digite um numero: 40.75
O valor final com desconto eh: 40.75
```

Atenção: atente-se pois a variável que receberá o preço inserido pelo usuário é do tipo float. Além disso, os prints são formatados com 2 casas decimais.

### 3) Cálculo de IMC 2.0

Na aula anterior, criamos um algoritmo simples para calcular o IMC de um usuário com base em sua altura e peso. Agora, vamos dar mais utilidade a essa aplicação. Com base no valor do IMC calculado, você vai usar estruturas condicionais para exibir a classificação correspondente à condição do usuário, como 'abaixo do peso', 'peso normal', 'sobrepeso', ou 'obesidade'.

Fórmula do IMC:  $IMC = \text{peso (em kg)} / (\text{altura} * \text{altura})$

IMC (kg/m <sup>2</sup> )	Diagnóstico
Menos de 18.5	Abaixo do peso
18.5 - 24.9	Peso normal
25.0 - 29.9	Sobrepeso
30.0 - 34.9	Obesidade Grau 1
35.0 - 39.9	Obesidade Grau 2
40.0 ou mais	Obesidade Grau 3 (mórbida)

Exemplos de execução:

```
Digite seu peso em kg: 55
Digite sua altura em metros: 1.75
Seu IMC eh: 17.96
Abaixo do peso
```

```
Digite seu peso em kg: 85.5
Digite sua altura em metros: 1.91
Seu IMC eh: 23.43
Peso normal
```

```
Digite seu peso em kg: 85
Digite sua altura em metros: 1.80
Seu IMC eh: 26.23
Sobrepeso
```

```
Digite seu peso em kg: 90
Digite sua altura em metros: 1.65
```

```
Seu IMC eh: 33.06  
Obesidade Grau 1
```

```
Digite seu peso em kg: 95  
Digite sua altura em metros: 1.60  
Seu IMC eh: 37.11  
Obesidade Grau 2
```

```
Digite seu peso em kg: 120  
Digite sua altura em metros: 1.70  
Seu IMC eh: 41.52  
Obesidade Grau 3 (morbida)
```

## Noção de Blocos e Escopo

Em C, um **bloco** de código é definido por chaves {}. Quando usamos estruturas condicionais, é importante entender que variáveis declaradas dentro de um bloco só são visíveis dentro desse bloco. Esse conceito é chamado de **escopo**.

Exemplo de escopo:

```
#include <stdio.h>

int main() {
    int x = 10;

    if (x > 0) {
        int y = 20; // 'y' só existe dentro deste bloco if
        printf("Dentro do if, y = %d\n", y);
    }

    // A variável 'y' não pode ser acessada aqui
    // printf("Fora do if, y = %d\n", y); // Isso resultaria em um erro

    return 0;
}
```

## Conclusão

Com essas estruturas condicionais, você pode controlar o fluxo de execução dos seus programas e fazer com que eles tomem decisões com base em condições. Pratique criando programas que verificam diferentes situações e explore a lógica condicional no seu código.

Na próxima aula, vamos aprender sobre laços de repetição (loops) e como eles podem ser usados para repetir blocos de código.

