

# Estruturas Condicionais

# O que são?

Estruturas utilizadas para desviar o fluxo  
de execução de um algoritmo  
dependendo do resultado de uma  
comparação lógica.

# Operadores de Comparação Lógica

(>) Maior

(<) Menor

(>=) Maior ou Igual

(<=) Menor ou Igual

(==) Igual

(!=) Diferente

(&&) E

(||) Ou

# If/else

```
int x = 1;  
int y = 2;  
  
if(x > y) {  
    printf("X é maior do que Y");  
} else {  
    printf("Y é maior do que X");  
}
```

Saída:

Y é maior do que X

```
int x = 5;  
int y = 5;  
  
if(x == y) {  
    printf("X é igual a Y");  
} else if (x < y) {  
    printf("Y é maior do que X");  
} else {  
    printf("X é maior do que Y");  
}
```

Saída:

X é igual a Y

# Switch/case

```
double x = 8;  
double restoDivisao = x % 2;  
  
switch(restoDivisao) {  
    case 0:  
        printf("X é par");  
        break;  
    case 1:  
        printf("X é ímpar");  
        break;  
}
```

Saída:

X é par

# Exercício de fixação

Criar um algoritmo que recebe um número inteiro e verifica, por meio de uma única estrutura condicional, se este é divisível por 2 e 3.

Dicas:

Utilize a estrutura if/else;

Combine expressões com &&. Exemplo:

```
if(x == 1 && y > x) {  
    printf("Y é maior que 1");  
}
```

# Resolução do Exercício

```
int n;

printf("Insira um número inteiro: ");
scanf("%d", &n);

int restoDivisaoDois = n % 2;
int restoDivisaoTres = n % 3;

if(restoDivisaoDois == 1 && restoDivisaoTres == 1) {
    printf("%d não é divisível por 2 e nem por 3", n);
} else if (restoDivisaoDois == 0 && restoDivisaoTres == 1) {
    printf("%d é divisível por 2, mas não por 3", n);
} else if (restoDivisaoDois == 1 && restoDivisaoTres == 0) {
    printf("%d é divisível por 3, mas não por 2", n);
} else {
    printf("%d é divisível por 2 e por 3", n);
}
```