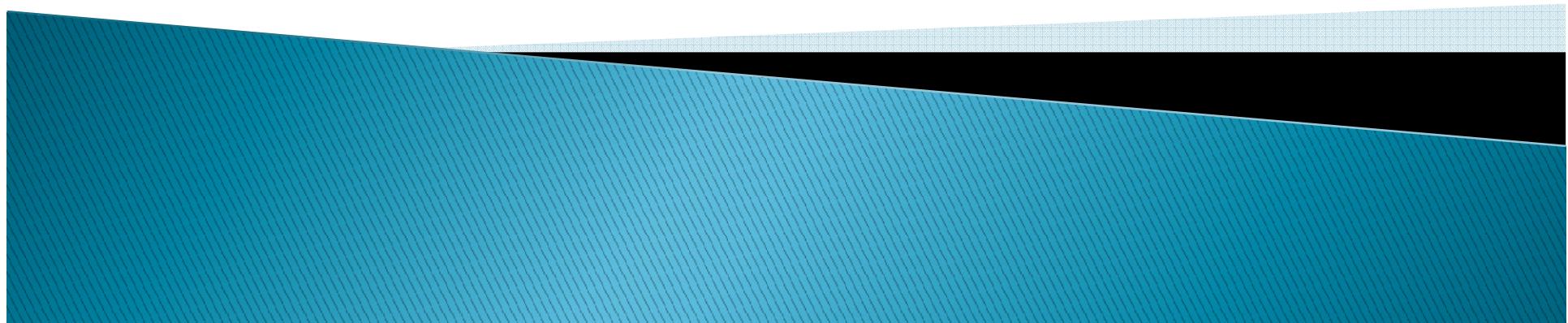


Instituto Federal da Bahia-IFBa
Curso: Engenharia Elétrica
Curso da Linguagem C

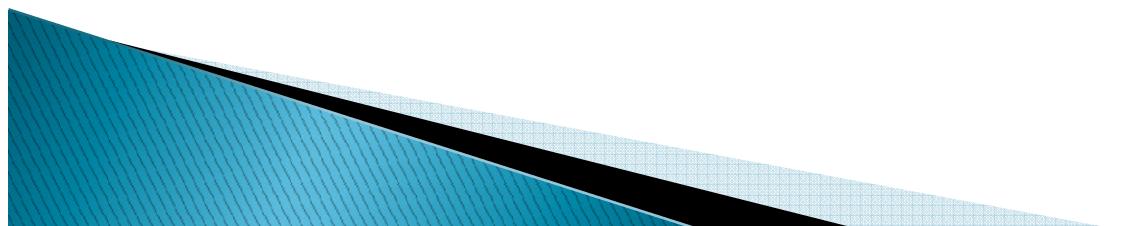
Estruturas Condicionais

Prof. Luiz Cláudio Machado



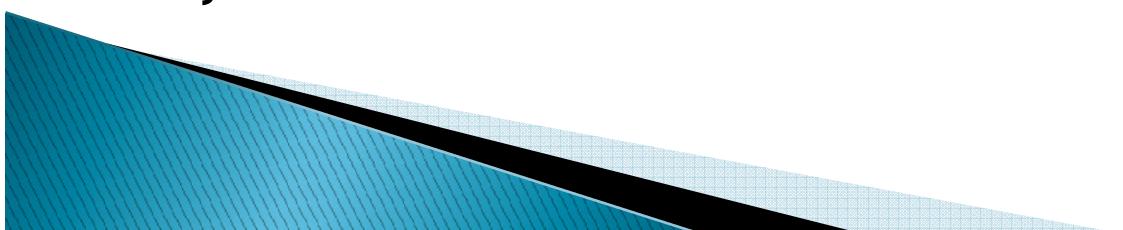
Estruturas Condicionais

- ▶ É possível que o programa execute diferentes tipos de procedimentos baseados em uma determinada decisão.
- ▶ Basicamente, existem dois tipos de estruturas condicionais: alternativa simples e alternativa composta.



Estruturas Condicionais Simples

```
#include "stdio.h"
#include "conio.h"
main ( )
{
    int a;
    clrscr( );
    printf ("Entre com o valor de A:");
    scanf ("%d", &a);
    if ( a > 0 )
        printf ("A e maior que ZERO\n");
}
```



Estruturas Condicionais Composta

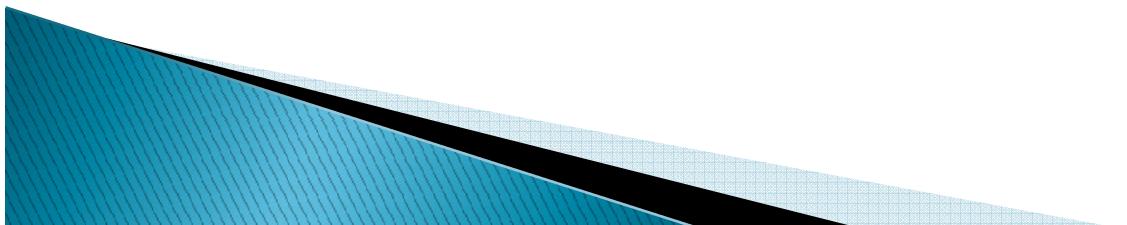
Essa estrutura permite ao usuário retornar dois valores possíveis. O primeiro verdadeiro, se a condição estipulada for satisfeita e o segundo falso, caso a condição não seja devidamente atendida.

```
#include <stdio.h>
#include <conio.h>
main ( )
{
    float salario;
    clrscr ( );
    printf ("Entre com seu salario:");
    scanf ("%f", &salario);
    if ( salario > 1500 )
        printf ("Voce ganha bem\n");
    else
        printf ("Voce precisa ganhar mais um pouquinho\n");
}
```



ENCADEAMENTO DE if's

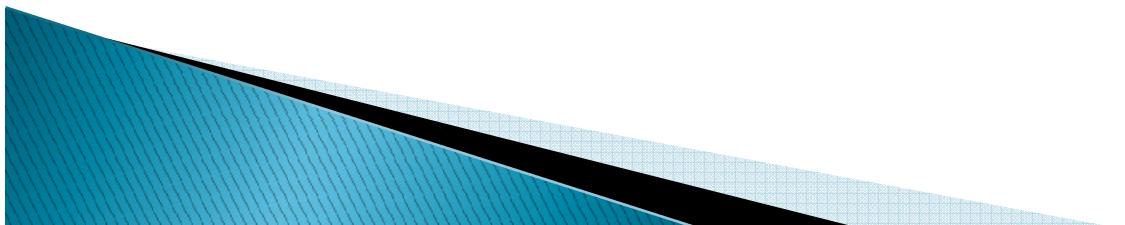
- ▶ Trata-se de um recurso que permite ao usuário utilizar uma estrutura if dentro de outra obtendo, assim, diversas respostas possíveis.



ENCADEAMENTO DE if's

```
/* Uso do ninho de if's */
#include "stdio.h"

void main ()
{
    int num;
    printf ("Entre com um numero:");
    scanf ("%d", &num);
    if ( num = 0 )
        printf ("Numero ZERO");
    else
        if ( num < 0 )
            printf ("Numero Negativo");
        else
            printf ("Numero Positivo");
}
```

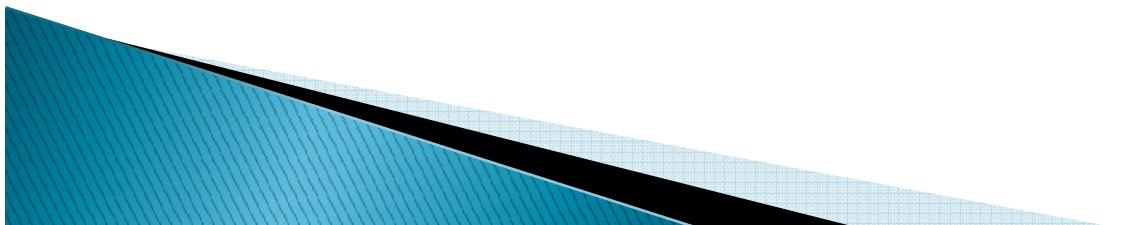


Elself

```
#include <stdio.h>
int main ()
{
    int num;
    printf ("Digite um numero: ");
    scanf ("%d",&num);
    if (num>10)
        printf ("\n\nO numero e maior que 10");
    else if (num==10)
    {
        printf ("\n\nVoce acertou!\n");
        printf ("O numero e igual a 10.");
    }
    else if (num<10)
        printf ("\n\nO numero e menor que 10");
    return(0);
}
```

Operador Condicional Ternário

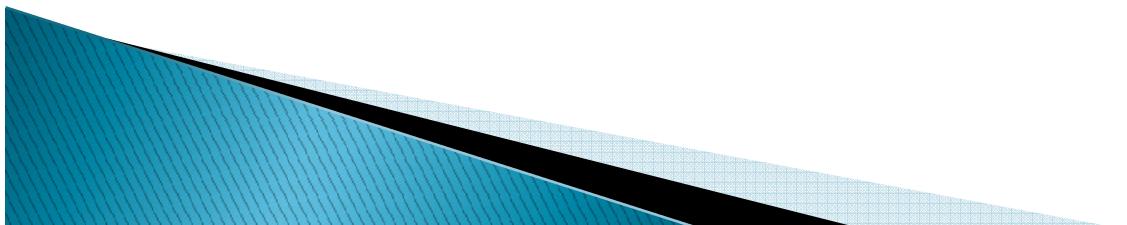
- ▶ Funciona muito bem para situações de decisões do tipo IF... THEN... ELSE, ou seja, quando em uma determinada condição tem-se que obter duas alternativas possíveis, uma verdadeira e outra falsa.



Operador Condicional Ternário

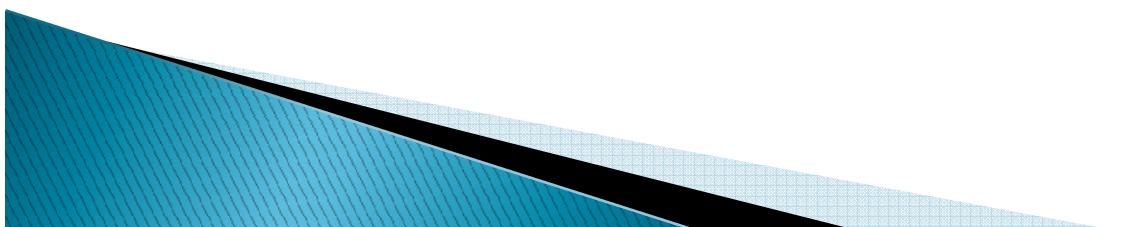
```
#include <stdio.h>
#include <conio.h>

main ()
{
    int a,b,max;
    clrscr();
    printf ("Digite dois numeros:");
    scanf ("%d %d", &a, &b);
    max = ( a > b ) ? a : b;
    printf ("O maior deles e %d\n", max);
}
```



Operador Condicional Ternário

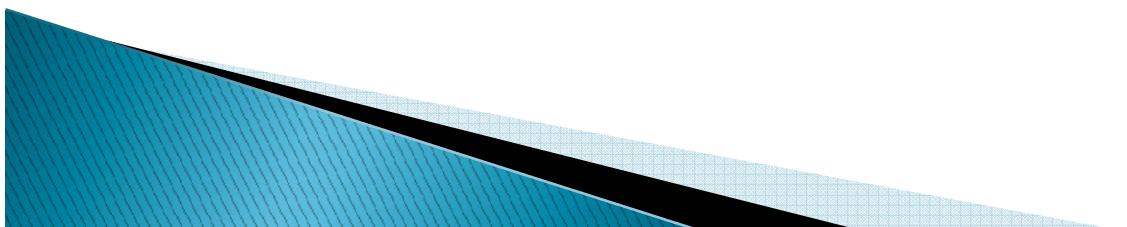
```
#include "stdio.h"
#include "conio.h"
main ()
{
    int a,b;
    clrscr ();
    printf ("Digite dois numeros:");
    scanf ("%d %d", &a, &b);
    printf ("O maior deles e %d\n", (a > b) ? a : b);
}
```



MÚLTIPLA ESCOLHA

switch... case...

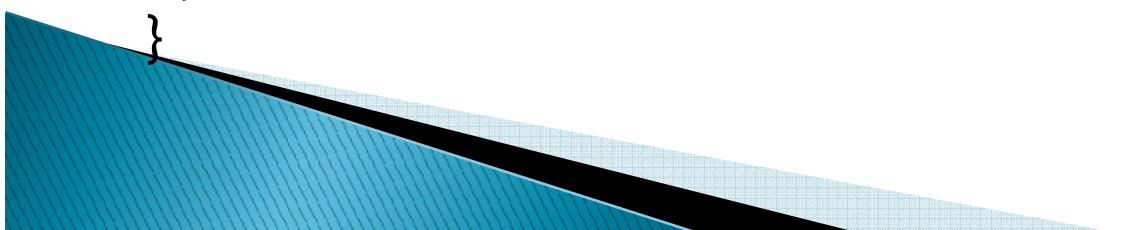
- ▶ A utilização do switch/case oferece inúmeras vantagens em relação à utilização da estrutura ninho de if's.
- Um exemplo prático é a facilidade de escrita de uma estrutura composta por múltiplas escolhas, que requer diversas alternativas, a partir de um certo programa de computador.



MÚLTIPLA ESCOLHA

switch... case...

```
▶ #include <stdio.h>
main ( )
{
    char opcao;
    puts ("Entre com uma letra:");
    opcao = getch();
    switch ( opcao )
    {
        case 'A' : printf ("Letra A\n");
        case 'B' : printf ("Letra B\n");
        case 'C' : printf ("Letra C\n");
        case 'D' : printf ("Letra D\n");
        default : printf ("Não e A, B, C nem D\n");
    }
}
```



MÚLTIPLA ESCOLHA

switch....case....

```
#include <stdio.h>
#include <conio.h>
main ()
{
    char
    opcao;
    clrscr( );
    puts ("Entre com uma letra:");
    opcao = getch( );
    switch ( opcao )
    {
        case 'A' :
            printf ("Letra A\n");
            break;
        case 'B' :
            printf ("Letra B\n");
            break;
        case 'C' :
            printf ("Letra C\n");
            break;
        default :
            printf ("Não e A, B, C nem D\n");
    }
}
```



Operação Curto Circuito

- ▶ Operadores Lógicos de Curto Circuito
- ▶ Ao encontrarem um valor que determine seu resultado, não testam mais as outras condições, isto é, a expressão só será analisada até que seja possível determinar o resultado
- ▶ Exemplo: considere que os métodos a e b retornem booleano
- ▶ a() & b() ! executa os dois métodos para obter o resultado
- ▶ a() | b() ! executa os dois métodos para obter o resultado
- ▶ a() && b() ! se o método a retornar false, o resultado será false e o método b nem será executado
- ▶ a() || b() ! se o método a retornar true, o resultado será true e o método b nem será executado

