



**TECNOLOGIA E INOVAÇÃO  
EM PROL DA INDÚSTRIA**



**Técnico em Informática**

# Programação de aplicativos – 140h

Prof<sup>a</sup>: Francisleide Almeida

# Estrutura de Decisão



# Instrução if

- A instrução "if" é a estrutura de teste mais básica, ela é encontrada em todas as linguagens (com sintaxes diferentes...).
- Ela executa uma série de instruções caso uma condição se realize. A sintaxe desta expressão é a seguinte:

```
if (condição realizada) {  
    lista de instruções;  
}
```

**Observações:** A condição deve ficar entre parênteses É possível definir várias condições a serem cumpridas com os operadores **E OU** (&& e ||) **Por exemplo, a seguinte instrução testa se as duas condições são verdadeiras:**

```
if ((condição1)&&(condição2))
```

# If .. Else

- A instrução *if* em sua forma básica só testa uma condição ou, na maioria das vezes, gostaríamos de poder escolher as instruções a serem executadas **em caso do não cumprimento da condição...**
- A expressão *if ... else* executa outra série de instruções, em caso do não cumprimento da condição.
- A sintaxe desta expressão é a seguinte:

```
if (condição realizada) {  
    lista de instruções  
} else {  
    outra série de instruções  
}
```

# If.. Else

- **Uma maneira mais rápida de fazer o teste** É possível fazer um teste com uma estrutura muito mais leve, graças à seguinte estrutura:

(condição) ? instrução se verdadeira : instrução se falsa

- Observações:
  - A condição deve ficar entre parênteses
  - Quando a condição for verdadeira, a Instrução da esquerda será executada
  - Quando a condição for falsa, a instrução da direita será executada
- ***Assim, esta forma forma: é frequentemente utilizada como segue:***

posição = ((Antes == 1) ? contador+1 : contador-1);

# Switch

- A Instrução *switch* efetua vários testes de valores sobre o conteúdo de uma mesma variável. Esta conexão condicional simplifica muito o teste de vários valores de uma variável, pois esta operação teria sido complicada (mas possível) com *ifs aninhados*. *Sua sintaxe é a seguinte:*

```
switch (variável) {  
    case Valor1 :  
        Lista de instruções;  
        break;  
    case Valor2 :  
        Lista de instruções;  
        break;  
    case Valores... :  
        Lista de instruções;  
        break;  
    default:  
        Lista de instruções;  
}
```



# Loops

- Os loops são estruturas que executam várias vezes a mesma série de instruções até que uma condição não seja mais realizada...

- **O loop for**

A instrução for executa várias vezes a mesma série de instruções: é um loop!

A sintaxe desta expressão é a seguinte:

```
for (contador; condição; modificação do contador) {  
    lista de instruções;  
}
```

**Por exemplo:**

```
for (i=1; i<6; i++) { printf("%d", i); }
```

**OBS.:** É preciso contar o número de vezes que você deseja executar o loop:

- `for(i=0;i<10;i++)` executa 10 vezes o loop (i de 0 a 9)
- `for(i=0;i<=10;i++)` executa 11 vezes o loop (i de 0 a 10)
- `for(i=1;i<10;i++)` executa 9 vezes o loop (i de 1 a 9)

# Loops

- **while e do-while:**  
**aplicações em C**
- Em **C**, o enquanto e substituído por ***while*** e o faça-enquanto por ***do-while***

```
while ( condição ) {  
    comando_1;  
    comando_2;  
    . . .  
    comando_n;  
}
```

```
do {  
    comando_1;  
    comando_2;  
    . . .  
    comando_n;  
} while ( condição );
```

# Loops

- **A instrução While** A instrução While representa outro meio de executar várias vezes a mesma série de instruções. A sintaxe desta expressão é a seguinte:  

```
while (condição realizada) {  
    lista de instruções;  
}
```

# Loops

- Pulo incondicional
  - A sintaxe desta expressão é "*continue*" (esta instrução é colocada em um loop!), e é geralmente associada a uma estrutura condicional, caso contrário, as linhas entre esta instrução e o final do loop seriam obsoletas.

```
x = 1;
while (x<=10) {
    if (x == 7) {
        printf("Division par 0");
        x++;
        continue;
    }
    a = 1/(x-7);
    printf("%f", a);
    x++;
}
```

# Atividades

- Elabore um algoritmo que calcule o valor do fatorial de um número informado pelo usuário
- Elabore um algoritmo que receba um valor inteiro e informe se o valor recebido é um número primo.
- Escreva um algoritmo que mostre o menor e o maior dos números reais digitados pelo usuário. Termine a leitura se o usuário digitar zero.