

Instituto Federal Sudeste de Minas Gerais

Campus Barbacena

Curso de Tecnologia em Sistemas para Internet

Disciplina: Lógica de Programação

Prof. Wender Magno Cota

Assunto: Estruturas Condicionais

Estrutura Condicional

Usada quando um comando ou conjunto de comandos tiver sua execução condicionada a veracidade de um condição.

Iremos trabalhar com as estruturas condicionais:

- if ... else
- switch
- Operador ternário

Comando if

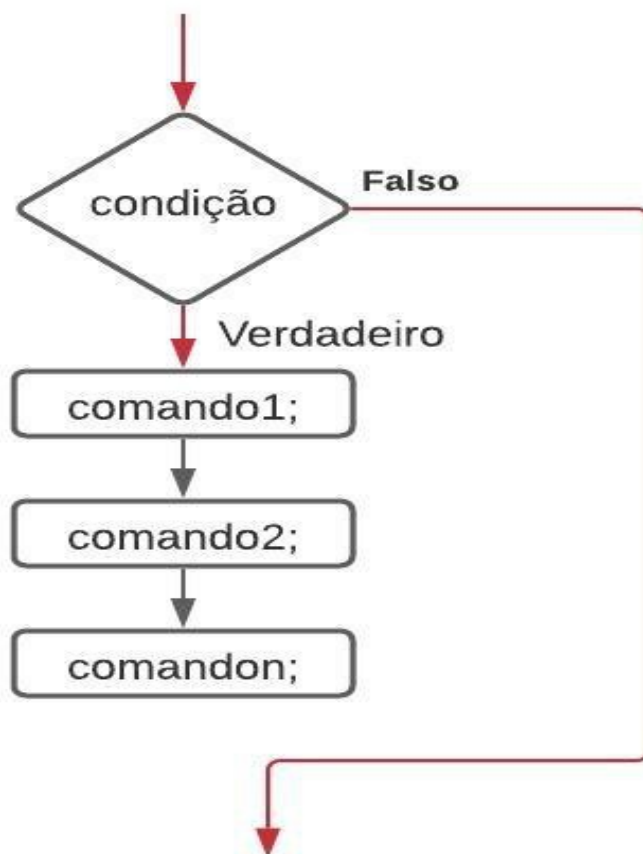
Comando if simples

```
if (condição) entao  
    Comando1;
```

Quando mais de um comando tiver sua execução condicionada a veracidade de **condição** deve envolver tais comandos pelos caracteres de { }(usados para delimitar um bloco de comandos).

```
if(condição){  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```

Comando if-fluxograma



Comando if

Onde:

- Condição: Tem que ser uma expressão Lógica(valor verdadeiro ou Falso);
- Comando1: pode ser um comando ou um conjunto de comandos válidos da linguagem.

Semântica:

Os comandos do comando **if** só serão executados quando **condição** for verdadeiro. No caso dela ter o valor falso os comandos da cláusula **else** serão executados. Os comandos do **if** e do **else** são mutualmente exclusivos.

Nota:

Em C os inteiros também são usados como valores **booleanos**: qualquer valor não nulo (1) representa **true(verdadeiro)** e 0 representa **false(falso)**.

Comando if

```
/*programa que calcula/mostra a área de uma superfície circular*/  
#include <stdio.h>
```

```
int main(){  
    const double PI=3.1415;//declaração da constante PI  
    double raio,area;  
    printf("Forneça a medida do raio:");  
    scanf("%lf",&raio);  
  
    //verificando se o valor fornecido para o raio é válido(>0)  
    if(raio>0){  
        area =PI*raio*raio*raio;  
        printf("Área da Superfície circular = %lf",area);  
    }  
  
    return 0;  
}
```

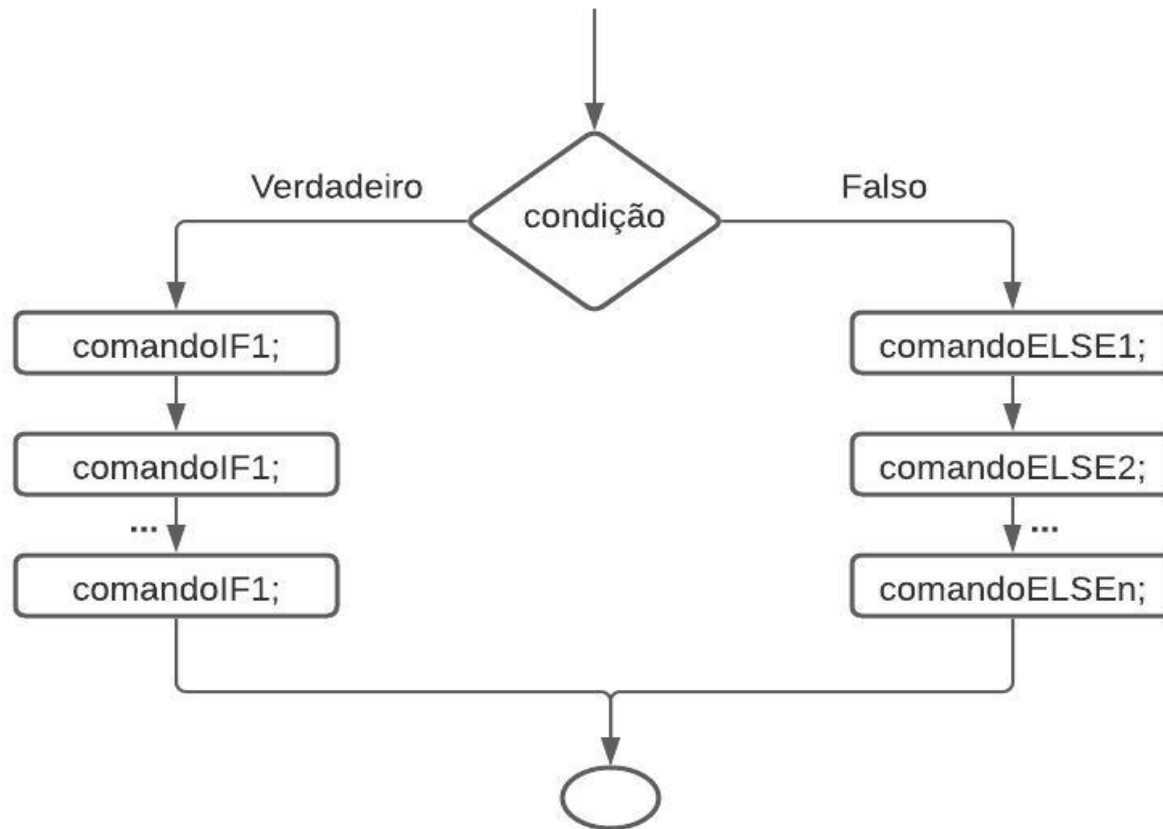
Comando if composto

- O comando **if** permite executar um ou mais comandos se a expressão relacional resultar em **verdadeiro**. Se desejarmos que algo seja executado se a expressão relacional resultar em **falso**, então devemos utilizar o comando **if...else**.
- Dessa forma, podemos executar um ou mais comandos se o teste for **verdadeiro**; ou executar um ou mais comandos distintos se o teste for **falso**.
- O comando **if-else** consiste no comando **if** seguido de uma instrução (ou um bloco de instruções), seguido da palavra-chave **else**, que por sua vez é seguida de uma instrução (ou um bloco de instruções).

Comando if composto

```
if( condição){  
    ComandoIF1;  
    ComandoIF2;  
    ...  
    ComandoIFn;  
}  
else{  
    ComandoELSE1;  
    ComandoELSE2;  
    ...  
    ComandoELSEn;  
}
```


Comando if..else



Comando if..else

```
/*programa que calcula/mostra a área de uma superfície circular.  
Mostrar uma mensagem de erro caso o valor fornecido para o raio não seja >0*/  
#include <stdio.h>
```

```
int main(){  
    const double PI=3.1415;//declaração da constante PI  
    double raio,area;  
    printf("Forneça a medida do raio:");  
    scanf("%lf",&raio);  
    //verificando se o valor fornecido é válido(>0)  
    if(raio>0){  
        area =PI*raio*raio*raio;  
        printf("Área da Superfície circular = %lf",area);  
    }  
    //verificando se o valor é incorreto  
    if(raio<=0)  
        printf("Valor do raio tem que ser maior que zero!!");  
    return 0;  
}
```

Nota: observe que ou consigo calcular a área ou mostro a mensagem de erro(situações mutuamente exclusivas). Assim sendo, poderíamos melhorar o programa, uma vez que o segundo teste só se justifica no caso do primeiro resultar em um valor falso.

Comando if..else

```
/*programa que calcula/mostra a área de uma superfície circular.  
Mostrar uma mensagem de erro caso o valor fornecido para o raio não seja >0*/  
#include <stdio.h>
```

```
int main(){  
    const double PI=3.1415;//declaração da constante PI  
    double raio,area;  
    printf("Forneça a medida do raio:");  
    scanf("%lf",&raio);  
    //verificando se o valor fornecido é válido(>0)  
    if(raio>0){  
        area =PI*raio*raio*raio;  
        printf("Área da Superfície circular = %lf",area);  
    }  
    else  
        printf("Valor do raio tem que ser maior que zero!!");  
    return 0;  
}
```

Comando if..else

```
/*programa para calcular as raízes de uma equação do segundo grau*/
```

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

```
#include <math.h>
```

```
int main(){
```

```
    float a,b,c,delta,x1,x2;
```

```
    printf("Forneça os coeficientes da equação:");
```

```
    scanf("%f%f%f",&a,&b,&c);
```

```
    if(a==0)
```

```
        printf("Não é uma equação do segundo grau");
```

```
    else{
```

```
        delta = b*b-4*a*c;
```

```
        if(delta<0)
```

```
            printf("A equação não possui solução real");
```

```
        else{
```

```
            x1=(-b+pow(delta,0.5))/(2*a);
```

```
            x2=(-b-pow(delta,0.5))/(2*a);
```

```
            printf("Raízes da equação %.2f e %.2f",x1,x2);
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

Comando switch

Usado para testar(apenas a igualdade) uma expressão(de um tipo ordenado) em relação a um conjunto de valores(constantes) pré-estabelecidos.

Sintaxe:

```
switch(<expressão>)  
{  
    case <valor1>: comandos1;  
        break;  
    case <valor2>: comandos2;  
        break;  
    case <valorn>: comandosn;  
        break;  
    default: comandos_default;  
}
```

Nota:

Se não usarmos o comando break em cada case o programa continuará até o fim do bloco.

Comando Switch

- A cláusula default é opcional
- O break faz com que a execução do switch seja interrompida.
- Expressão deve ser de um tipo ordinal.

Estrutura de Seleção - Switch

/*programa que leia uma operação(+, -, / ou *) e dois números. Calcular/Mostrar o resultado Da expressão formada pelo operador lido e os dois números.*/

```
#include <stdio.h>
int main(){
    float n1,n2;
    char op;
    printf("Forneça dois números:");
    scanf("%f%f",&n1,&n2);
    printf("Forneça um Operador + - * /:");
    scanf("%c",&op);
    switch (op)
    {
        case '+': printf("Resultado = %f",n1+n2); break;
        case '-': printf("Resultado = %f",n1-n2); break;
        case '*': printf("Resultado = %f",n1*n2); break;
        case '/': if(n2!=0)
                    printf("Resultado = %f",n1/n2);
                else
                    printf("Divisão por zero"); break;
        default: printf("Operador Incorreto");
    }

    return 0;
}
```

Comando switch

Usado para testar(apenas a igualdade) uma expressão(de um tipo ordenado) em relação a um conjunto de valores(constantes) pré-estabelecidos.

Sintaxe:

```
switch(<expressão>)  
{  
    case <valor1>: comandos1;  
        break;  
    case <valor2>: comandos2;  
        break;  
    case <valorn>: comandosn;  
        break;  
    default: comandos_default;  
}
```

Nota:

Se não usarmos o comando break em cada case o programa continuará até o fim do bloco.

Operador Ternário

Usado em expressões do tipo

```
if(condição)
```

```
    comando1;
```

```
else
```

```
    comando2;
```

Pode ser escrita como

```
condicao ? comando1 : comando2;
```

Operador Ternário

/*Programa que tem como entrada dois números inteiros e mostra o maior deles.*/

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

```
int main() {
```

```
    int n1,n2;
```

```
    printf("Forneça dois números inteiros:");
```

```
    scanf("%d%d",&n1,&n2);
```

```
    (n1>n2) ? printf("%d",n1) : printf("%d",n2);
```

```
    return 0;
```

```
}
```