



**INSTITUTO FEDERAL**  
Sul-rio-grandense

Campus  
Gravataí

# Estruturas de Controle em C - Seleção

Prof. Hunder Evertto Correa Junior



Curso Superior de Tecnologia em  
**Análise e Desenvolvimento**  
de Sistemas

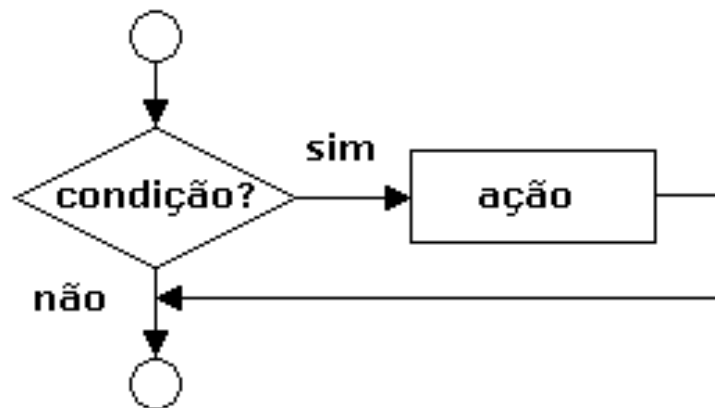
# Introdução

- Nem sempre é possível resolver um problema somente com instruções sequenciais.
- Nesta aula iremos entender o funcionamento das estruturas de Seleção.



# Estrutura de Seleção Simples

- Os comandos de decisão ou Seleção fazem parte das técnicas de programação que conduzem a estruturas de programas que não são totalmente sequenciais. Com as instruções de SALTO ou DESVIO pode-se fazer com que o programa proceda de uma ou outra maneira, de acordo com as decisões lógicas tomadas em função dos dados ou resultados anteriores.



# Seleção Simples

- COMANDO *IF-ELSE*

- 

- O comando ***if*** instrui o computador a tomar uma decisão simples. O comando ***else*** é acionado quando a condição for falsa.

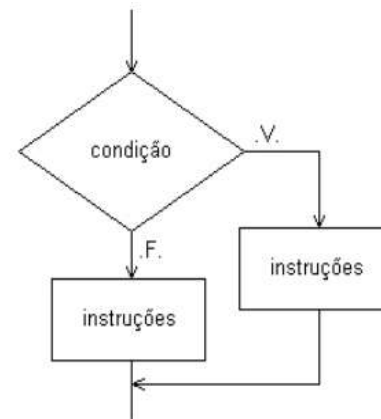
- Forma geral:

- 

- If (condição) comando ;

- Ou

- **if** ( condição ) comando;
- **else** comando;



## Pseudocódigo

```
se (<condição> )  
  então      <instruções>  
  senão      <instruções>  
fimse
```

# Exemplo

- `/* programa do numero magico */`
- `#include <stdio.h>`
- `#include <conio.h>`
- `void main( )`
- `{`
- `int magico, entrada;`
- `magico = rand() %20;    //gera um numero entre 0 e 20`
- `system("CLS");`
- `printf( "Adivinhe o numero: ");`
- `scanf("%d",&entrada);`
- `if (entrada == magico) printf (" Você acertou");`
- `else printf (" Você não acertou pressione qualquer tecla" );`
- `getch( );`
- `system("PAUSE");}`

# Importante

- Quando precisamos utilizar mais de uma expressão no comando IF, devemos utilizar a estrutura de bloco: `{ }`
- *Ex.*
- `if (num==10)`
- `{`     `printf ("\n\nVoce acertou!\n");`
- `printf ("O numero e igual a 10.\n");`
- `};`
- `else`
- `{`     `printf ("\n\nVoce errou!\n");`
- `printf ("O numero e diferente de 10.\n");`
- `};`

# EXERCÍCIO

- Escreva um programa onde o usuário entra com um número qualquer e o programa responda se o número é par ou ímpar.
- Se for par emite a mensagem “ O número é par ” ou caso contrário “O número é ímpar ”.

# Estrutura de Seleção com Múltipla escolha

- Muitas vezes nos deparamos com múltiplas alternativas, neste caso podemos utilizar a estrutura de Múltipla escolha:

```
escolha (X)
  caso E1:
    /*bloco de comandos*/
  caso E2:
    /*bloco de comandos*/
  caso E3:
    /*bloco de comandos*/
  caso Contrário: //pode ser omitido
    /*bloco de comandos*/
fim escolha;
```



```
switch (X)
{
  case E1:
    /*bloco de comandos*/
    break;
  case E2:
    /*bloco de comandos*/
    break;
  case E3:
    /*bloco de comandos*/
    break;
  default: //pode ser omitido
    /*bloco de comandos*/
    break;
}
```



# Seleção com Múltipla escolha

- **COMANDO *SWITCH CASE***

- 

- O comando switch pode ser usado no caso de alternativas múltiplas.

- Forma geral

- **switch**( variável )

- {

- **case** constante1: seqüência de comandos; **break;**

- **case** constante2: seqüência de comandos; **break;**

- **case** constante3: seqüência de comandos; **break;**

- .

- .

- **default:** seqüência de comandos;

- }

# Seleção com Múltipla Escolha

- O comando switch ao avaliar a expressão entre parênteses, desviamos para o rótulo case cujo valor coincida com o valor da expressão. O comando break serve para sairmos do bloco mais interno ao qual o break aparece. O comando break garante a execução de apenas uma chamada dentro do switch.

# Exemplo

- Exemplo:
- 
- `#include <stdio.h>`
- `void main ( )`
- `{`
- 
- `char opção;`
- `clrscr( );`
- `printf("A - imprime a letra f");`
- `printf("B - imprime a letra g");`
- `printf("C - imprime a letra h");`
- `opção = getch( );`
- `switch(opção)`
- `{`
- `case 'a': printf("f");break;`
- `case 'b': printf("g");break;`
- `case 'c': printf("h");break;`
- `}`
- `}`

# Exercícios

- Faça um programa contendo um menu com as seguintes opções:
- S - soma
- P - produto
- U - subtração
- D - divisão
- Q - sair
- O programa deve mostrar na tela a função escolhida : soma, subtração etc. Quando o usuário teclar **Q** o programa deve terminar.

# if-else-if

- A estrutura **if-else-if** é apenas uma extensão da estrutura **if-else**. Sua forma geral pode ser escrita como sendo:
- *if (condição\_1) declaração\_1;*  
*else if (condição\_2) declaração\_2;*  
*else if (condição\_3) declaração\_3;*  
*.*  
*.*  
*.*  
*else if (condição\_n) declaração\_n;*  
*else declaração\_default;*

# if-else-if

- O programa começa a testar as condições começando pela 1 e continua a testar até que ele ache uma expressão cujo resultado dê diferente de zero. Neste caso ele executa a declaração correspondente.
- Só uma declaração será executada, ou seja, só será executada a declaração equivalente à *primeira* condição que der diferente de zero.
- A última declaração (default) é a que será executada no caso de todas as condições darem zero e é opcional.

# if-else-if

- Exemplo:

```
int numero;  
printf("Digite um valor Inteiro!");  
scanf("%d",&numero);  
if (numero<3)  
{  
    printf("Numero Menor do que 3");  
}  
else if (numero==3)  
{  
    printf("Numero Igual a 3");  
}  
else  
{  
    printf("Numero Maior do que 3");  
}
```

# Faça os Exercícios...



Praticar, importante é !