



Instituto Federal de Educação, Ciência e Tecnologia do Ceará – IFCE  
*Campus* Tianguá  
Bacharelado em Ciência da Computação

# Introdução à Programação

Prof. Anderson Passos de Aragão

[anderson.aragao@ifce.edu.br](mailto:anderson.aragao@ifce.edu.br)

# Agenda

- Estrutura de Repetição em C

# Estrutura de Repetição

- Uma estrutura de repetição é utilizada quando um trecho do algoritmo, ou até mesmo o algoritmo inteiro, precisa ser repetido
- O número de repetições pode ser fixo ou estar atrelado a uma condição
- Assim, existem estruturas para tais situações e veremos como utilizá-las



# Estrutura de Repetição FOR

- Essa estrutura de repetição é utilizada quando **se sabe o número de vezes** que um trecho do programa deve ser repetido.
- O formato geral do comando for é composto por três partes:

```
for (i = valor_inicial; condição; incremento de i)  
comando;
```

# Estrutura de Repetição FOR

```
for (i = valor_inicial; condição; incremento de i)  
comando;
```

- A **primeira parte** atribui um **valor\_inicial** à variável **i**, que tem como função **controlar o número necessário de repetições**
- A **segunda parte** corresponde a uma **expressão relacional** que, **quando assumir o valor falso**, determinará o **fim da repetição**
- A **terceira parte** é responsável por **alterar o valor da variável i** (incremento ou decremento) com o objetivo de, em algum momento, **fazer a condição assumir o valor falso**

# Estrutura de Repetição FOR

- Exemplo:

```
for (a = 1; a <= 20; a++)  
    printf("\no valor de a é: %d", a);
```

# Estrutura de Repetição FOR

- Se for necessária a repetição de mais de um comando, o compilador entenderá que a estrutura de repetição começará quando for encontrado o símbolo **{** e terminará quando for encontrado o símbolo **}**
- Exemplo:

```
for (a = 15; a >= 1; a = a-2) {  
    printf("Digite um número: ");  
    scanf("%d", &x);  
}
```

# Estrutura de Repetição FOR

- Exemplo:

```
for (i = 1; i <= 5; i++)  
printf("%d", i);  
ou  
for (i = 1; i <= 5; i = i+1)  
printf("%d", i);
```



# Estrutura de Repetição FOR

- Existem duas instruções comumente usadas nos comandos internos das estruturas de repetição
- São as instruções denominadas **acumuladores** e **contadores**
- Os acumuladores devem ser usados quando a realização de um cálculo precisa de valores obtidos a cada iteração, ou seja, o cálculo só estará pronto com a conclusão da repetição



# Estrutura de Repetição FOR

- Acumuladores:

```
SOMA = 0; // inicialização da variável SOMA com o valor zero
for(I = 1; I <= 5; I++){
    printf("Digite um número: ");
    scanf("%d",&NUM);
    SOMA = SOMA + NUM;
}
printf("Soma = %d",SOMA);
```

# Estrutura de Repetição FOR

- Contadores:

```
CONT = 0;          // inicialização da variável CONT com o valor zero
for(I = 1; I<= 5; I++) {
    printf("Digite um número: ");
    scanf("%d ", &NUM);
    if (NUM > 5)
        CONT = CONT + 1; // contando mais 1 na variável CONT
}
printf("Quantidade de número maiores que 5 = %d", CONT);
```

# Estrutura de Repetição WHILE

- Trata-se de uma estrutura de repetição que pode ser utilizada quando o número de **REPETIÇÕES NECESSÁRIAS NÃO FOR FIXO**.
- Os comandos serão repetidos até a **CONDIÇÃO** assumir o valor **FALSO**.

```
while(condição)
{
    comando1;
    comando2;
    comando3;
    ...
}
```

# Estrutura de Repetição WHILE

- Nesse tipo de estrutura, o **teste condicional** ocorre no **INÍCIO**
- Logo existe a possibilidade da repetição **NÃO SER EXECUTADA** quando a condição assumir o valor **FALSO** logo na primeira verificação

```
while(condição)
{ comando1;
  comando2;
  comando3;
  ...
}
```

# Estrutura de Repetição WHILE

- Exemplo

```
X = 0;           // inicialização da variável x com o valor 0
while (X != 5)
    printf("Valor de X = %d",X);
    X = X + 1;    // contador incrementado em 1 unidade
}
printf("Valor de X depois que sair da estrutura = %d",X);
```

# Estrutura de Repetição WHILE

- Exemplo

```
X = 1;           // inicialização da variável X com o valor 1
Y = 10;          // inicialização da variável Y com o valor 10
while (Y > X){
    printf("Valor de Y = %d",Y);
    Y = Y - 2;    // contador decrementado em 2 unidades
}
printf("Valor de Y depois que sair da estrutura = %d",Y);
```

# Estrutura de Repetição DO-WHILE

- Trata-se de uma estrutura de repetição que pode ser utilizada quando o número de **REPETIÇÕES NECESSÁRIAS NÃO FOR FIXO**
- Os comandos serão repetidos até a condição assumir o valor **FALSO**
- Nesse tipo de estrutura, o **teste condicional** ocorre no **FIM**
- Logo a **REPETIÇÃO** será executada, **no mínimo, UMA VEZ**, quando todo o bloco for executado uma vez, e, ao final, a condição assumir o valor **FALSO**.



# Estrutura de Repetição DO-WHILE

- Sintaxe:

```
do {  
    comandos;  
} while (condição);
```

- Os comandos serão repetidos até que a condição assuma valor **FALSO**

# Estrutura de Repetição DO-WHILE

- Exemplo:

```
X = 0;           // inicialização da variável X com o valor 0
do {
    printf("Valor de X = %d",X);
    X = X + 1;   // contador incrementado em 1 unidade
} while (X != 5);
printf("Valor de X depois que sair da estrutura = %d",X);
```

# Estrutura de Repetição DO-WHILE

- Exemplo:

```
X = 1;           // inicialização da variável X com o valor 1
Y = 10;          // inicialização da variável Y com o valor 10
do {
    printf("Valor de Y = %d",Y);
    Y = Y - 2;    // decrementando o contador em 2 unidades
} while (Y > X);
printf("Valor de Y depois que sair da estrutura = %d",Y);
```

# Estrutura de Repetição Aninhadas

- Tal qual as Estruturas Condicionais as Estruturas de Repetição podem ser aninhadas
- Os mesmos cuidados devem ser tomados



# Dúvidas?



# Por hoje é só pessoal

Obrigado pela Atenção

