

# Algoritmo e Programação Estruturada

**Aula 6**

**Prof. Osmam Brás de Souto**

# Estrutura de Repetição (laço)

Em C existem três comandos de repetição que são empregados de maneira mais apropriada a determinadas situações, onde as características de cada um possam ser melhor aproveitadas.

*for*(<inicialização>;<condição>;<incremento>)  
instrução;

- ➔ A **inicialização** é executada uma única vez antes do laço ser iniciado
- ➔ A sequência de comandos será repetida enquanto a **condição** for verdadeira. Quando ela for falsa os comandos após o laço são executados
- ➔ O **incremento** define como será alterada a variável de controle do laço, cada vez que ele for repetido, sendo executado logo após o fim do corpo do laço

# Estrutura de Repetição

→ A *<inicialização>*, *<condição>* e *<incremento>*, podem compostas por qualquer instrução válida em C

→ A *<inicialização>*, *<condição>* e *<incremento>*, podem conter várias instruções separadas por ‘,’ e qualquer uma delas pode ser omitida, mantendo sempre os ‘;’ que as separam

→ A *<inicialização>*, *<condição>* e *<incremento>*, podem chamar funções

## Exemplos:

```
int main(void)
{
    int x,y;
    for(x=0,y=0;x+y<10;x=x+1,y=y+1)
        printf(“%d”,x+y);
}
```

```
int main(void)
{
    int c;
    for(c='a'; c<='z'; c++)
        printf(“%c ASCII= %d\n”,c,c);
}
```

```
int main(void)
{
    int c;
    for(c=getch();c!='X';c=getch())
        printf(“%c”,c+1);
}
```

```
int main(void)
{
    int c;
    for( ; (c=getch())!='X'; )
        printf(“%c”, c);
}
```

# Estrutura de Repetição(Exemplo-for)

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
    int count;

    for(count=1 ; count <= 5 ; count++)
    {
        printf("%d\n", count);
    }
    return 0;
}
```

# Estrutura de Repetição(Desafio)

Implementa uma contagem regressiva de 5 até 1, usando o laço FOR



Fonte: [br.pinterest.com](https://br.pinterest.com)

# Estrutura de Repetição(Resultado-Desafio)

Implementa uma contagem regressiva de 5 até 1, usando o laço FOR

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
    int count;

    for(count=5 ; count >= 1 ; count--)
    {
        printf("%d\n", count);
    }
    return 0;
}
```

# Estrutura de Repetição

## Instruções Múltiplas no Corpo do for

Duas ou mais instruções podem fazer parte do corpo do laço 'for'. Quando isso for necessário abra um bloco de instrução ({), coloque as instruções desejadas e feche o bloco (}). Não se esqueça do ';' para encerrar cada uma destas instruções.

## Laço for Aninhados

Quando tem-se um laço dentro de outro, diz-se que o laço interior esta aninhado

```
for (<inicialização>;<condição>;<incremento>)  
{  
    for (<inicialização>;<condição>;<incremento>)  
        instrução 1;  
        instrução 2;  
}
```

# Estrutura de Repetição

A segunda estrutura de repetição usa os mesmos elementos do laço 'for', mas eles estão distribuídos de forma diferente.

while (<condição>)

instrução;

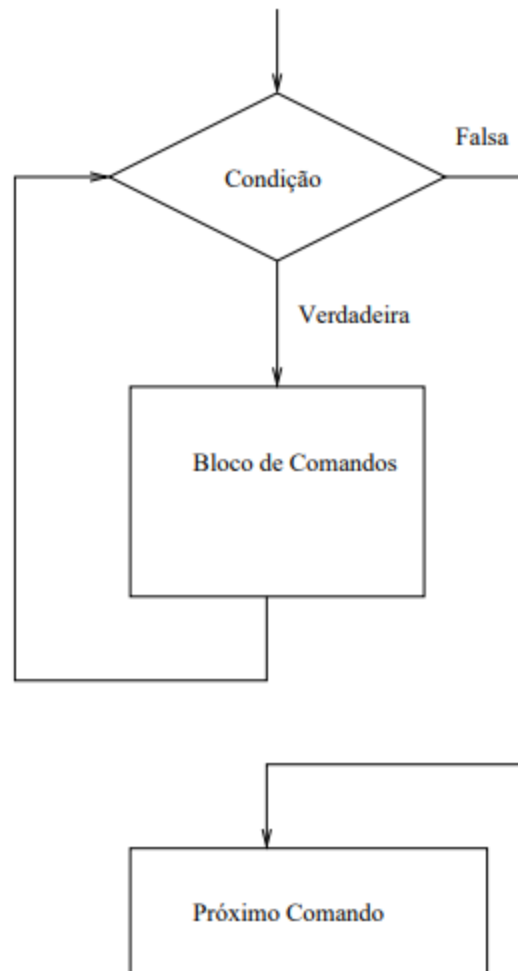
→ Se a condição for verdadeira ( $\neq 0$ ) o corpo do laço é executado e a condição é novamente avaliada. Esta operação se repete até que a condição se torne falsa ( $= 0$ ), encerrando o laço e continuando a executar o programa após o corpo do laço.

- O laço while é mais apropriado para situações que a repetição possa ser **encerrada inesperadamente**, enquanto que o 'for' é mais empregado em quantidades de repetições mais conhecidas.

- O while também pode ser aninhado, ou seja, possuir um while dentro de outro while.



# Estrutura de Repetição(while)



# Estrutura de Repetição(Exemplo-while)

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int n;
    int soma = 0;
    printf("Digite um numero positivo para ser somado ou negativo para sair: ");
    scanf("%d", &n);
    while( n >= 0 )
    {
        soma = soma + n;
        printf("\nDigite um numero positivo para ser somado ou negativo para sair: ");
        scanf("%d", &n);
    }
    printf("A soma e %d\n", soma);
    return 0;
}
```

# Estrutura de Repetição

A última estrutura de repetição cria um ciclo repetitivo até a condição ser falsa ( $= 0$ ). A condição neste laço é avaliada depois do laço ser executado, ou seja, seu bloco é executado ao menos uma vez para que a condição seja verificada.

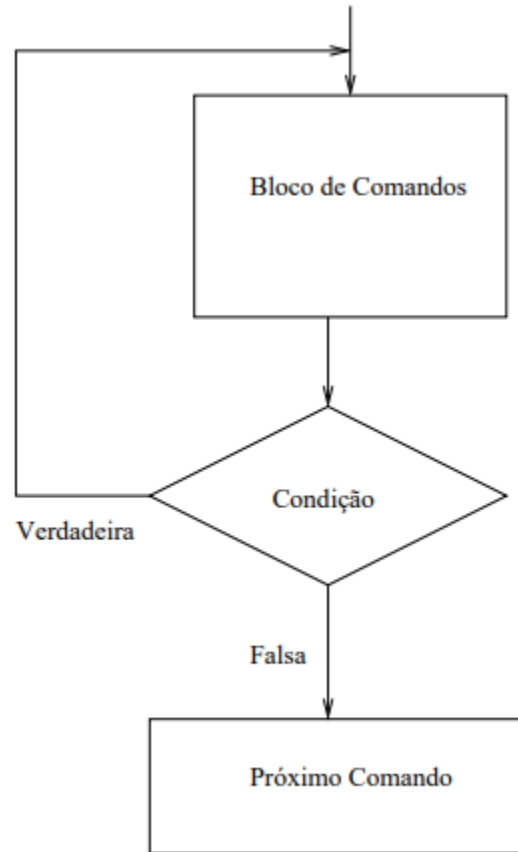
```
do {  
    instruções;  
} while (<condição>);
```

→ As chaves não são sempre necessárias, somente quando existem mais que uma instrução no bloco, mas elas favorecem a legibilidade do programa.

Exemplo: :

```
do {  
    total += valor;  
    printf("O total seria %5.2f",total);  
} while (total < 8000);  
:
```

# Estrutura de Repetição(do-while)



Fonte: <https://www.ic.unicamp.br/~jpaulo/cursos/mc102/2s2007/aula6.pdf>

# Estrutura de Repetição(Exemplo-do-while)

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int n;
    int soma = 0;
    do
    {
        printf("\nDigite um numero positivo para ser somado ou negativo para sair: ");
        scanf("%d", &n);
        if( n >= 0 )
            soma = soma + n;
    }while( n >= 0 );
    printf("A soma e %d\n", soma);
    return 0;
}
```

# Referências

- EVARISTO, J., Aprendendo a programar programando em C, Book Express, 205 p., 2001.
  - Capítulo 1 e 2
- MIZRAHI, V. V., Treinamento em Linguagem C, Curso Completo, Módulo 1, Makron Books do Brasil Editora Ltda, 1990.
  - Capítulo 1