

Agenda

- Estruturas de Repetição
 - Teste no Início (Sintaxe **while**)

E.R - Teste no Início

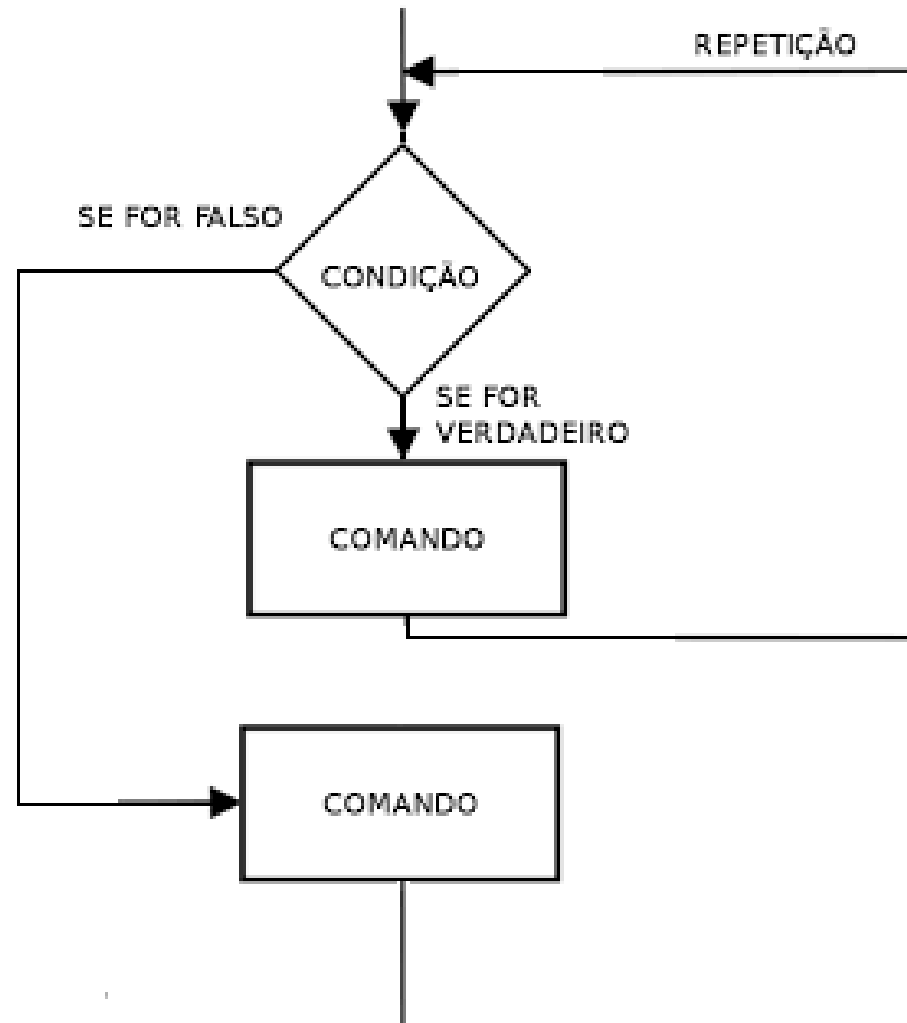
Estrutura de repetição com teste no início verifica antes de cada execução se é permitido repetir o fluxo identificado.

E.R - Teste no Início

Se o resultado for **falso**, o comando que representa a estrutura de repetição é abandonado.

Pode não ser executado **nenhuma** vez.

E.R - Teste no Início



E.R - Teste no Início

Sintaxe:

```
while (condição) {  
    Comando1;  
    Comando2;  
    ComandoN;  
}
```

E.R - Teste no Início

Exercício Exemplo:

Crie um programa para solicitar um número de 1 a 4 (incluindo 1 e 4). Se a pessoa digitar um número diferente, mostrar a mensagem "entrada inválida" e solicitar o número novamente. Se digitar correto mostrar o número digitado.

E.R - Teste no Início

```
#include <stdio.h>
main() {
    int num = 0;
    while (num < 1 || num > 4) {
        printf("\n\n Informe um numero entre 1 e 4: ");
        scanf("%d", &num);
        if (num < 1 || num > 4) {
            printf("\nVoce nao digitou um numero entre 1 e 4");
            printf("\nEntrada Invalida.");
        }
    }
    printf("\n\n Numero digitado: %d", num);
}
```

E.R - Teste no Início

Exercício Exemplo:

Crie um programa para solicitar a idade de várias pessoas e imprimir o total de pessoas com menos de 21 anos e o total de pessoas com mais de 50 anos. O programa termina quando a idade digitada for igual a -99.

E.R - Teste no Início

```
#include<stdio.h>
main()
{
    int idade, contador21, contador50;
    contador21=0;
    contador50=0;
    while (idade!=-99)
    {
        printf("\n DIGITE A IDADE (PARA ENCERRAR DIGITE -99): ");
        scanf("%d", &idade);
        if(idade<21) contador21++;
        if(idade>50) contador50++;
    }
    printf("\n O TOTAL DE PESSOAS COM MENOS DE 21 ANOS E:%d", contador21);
    printf("\n O TOTAL DE PESSOAS COM MAIS DE 50 ANOS E:%d", contador50);
}
```

Exercícios comando WHILE

1- Crie um programa que leia um valor inteiro **n** e imprima todos os números **pares** de 0 até n (incluindo 0 e n).

2- Fazer um programa que receba vários números naturais do teclado (um de cada vez) e determine o maior. A condição de término do programa é quando o usuário digitar um número negativo.

Bibliografia

FORBELLONE, A. L. V. e EBERSPACHER, H. F. **Lógica de Programação – A Construção de Algoritmos e Estrutura de Dados.** 3ª Edição. Prentice Hall. 2005. Editora, 1999

FARRER, H. **Algoritmos Estruturados.** 3 ed. Rio de Janeiro: LTC

CORMEN, T. H. e LEISERSON, C. E. **Algoritmos - teoria e prática.** 1. ed. Rio de Janeiro: Campus, 2002