

[◀ VOLTAR](#)

# Implementação de estruturas de repetição em C

Apresentar os conceitos de estruturas de repetição e como elas são utilizadas na linguagem C. Serão apresentados exemplos de cada desvio em C.

## NESTE TÓPICO

- > Introdução
- > Estrutura de Repetição Enquanto em C
- > Exemplo 1
- > Exemplo 2
- > Estrutura de Repetição Repita em C

Marcar  
tópico



## Introdução

Em determinadas situações, temos de repetir o programa ou parte dele várias vezes, como no cálculo das médias das notas de um grupo de alunos. Uma forma de solucionar este problema, seria escrever o algoritmo em questão uma vez para cada aluno, ou seja, se forem 10 alunos, teríamos de escrevê-lo 10 vezes, o que é inviável.

Outro modo de resolver essa questão seria utilizar a mesma sequência de comandos novamente, ou seja, teríamos de realizar um retrocesso (ao início dos comandos) para cada aluno, fazendo, portanto, com que o fluxo de execução repetisse certo trecho do algoritmo, o que nessa aplicação corresponderia a repetir o mesmo trecho 30 vezes, sem, no entanto, ter de escrevê-lo 10 vezes.

A esses trechos do algoritmo que são repetidos damos o nome de estruturas de repetição. O número de repetições pode ser indeterminado, porém necessariamente finito.

O conceito de repetição (*looping* ou *laço*) é utilizado quando se deseja repetir um certo trecho de instruções por um número de vezes.

As estruturas de repetição se dividem em ENQUANTO, REPITA, PARA. Para determinamos qual é a estrutura mais adequada para determinado programa, devemos saber qual o número de vezes que o trecho do programa

vai ser executado (laços contados) ou a condição para que ele aconteça (laços condicionais). Neste tópico serão abordadas todas as estruturas de repetição na linguagem C.

## Estrutura de Repetição Enquanto em C

Consiste de uma estrutura de controle do fluxo de execução que permite repetir diversas vezes um mesmo trecho do algoritmo, porém, sempre verificando **antes** de cada execução se é permitido executar o mesmo trecho.

Desse modo, na estrutura enquanto, a condição de repetição é verificada antes de entrar no laço, isto é, uma condição é testada inicialmente e, se o resultado for verdadeiro, o bloco de instrução será executado.

Na Linguagem C, a utilização do enquanto é feita a partir da instrução *while*, sendo seu funcionamento controlado por decisão (condição).

Assim, enquanto o valor da condição for verdadeiro, as ações dos comandos são executadas. No momento em que a condição se torna falsa, o processamento da rotina é desviado para fora do laço. Se a condição for falsa logo de início, os comandos não são executados nenhuma vez, ou seja, as instruções contidas no laço são ignoradas.

A estrutura enquanto, na linguagem C, deve ser utilizada da seguinte forma:

```
1. while ( <condicao> ){  
2.  
3.     instruções;  
4.  
5. }
```

O enquanto é utilizado quando:

- Não sabemos o número de repetições.
- Quando possuímos uma expressão que deve ser avaliada para que os comandos da estrutura sejam executados.

Quando utilizamos os laços contados, conhecemos previamente quantas vezes o conjunto de instruções será repetido; então, precisamos do auxílio de um contador, que auxiliará no *looping* (laço) da estrutura. A utilização da variável (cont) possibilita a repetição da estrutura até que o contador atinja o limite estipulado na condição da estrutura. Assim, ele deixa de executar as instruções seguindo o fluxo do programa.

## Exemplo 1

Um exemplo da utilização de um laço enquanto com contador é o cálculo da média para 10 alunos. O processo de contagem ocorre por meio da expressão aritmética que obtém o valor da variável *cont* e adiciona 1, armazenando esse resultado na própria variável *cont*. Repetindo esse comando várias vezes, perceberemos que a variável vai aumentando gradativamente de valor (de 1 em 1), simulando a contagem das execuções. Este exemplo pode ser observado a seguir:

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. main () {
4.     int cont;
5.     float n1, n2, n3, m;
6.     cont = 1;
7.
8.     while (cont <= 10) {
9.
10.        printf ("Digite 3 notas \n");
11.        scanf ("%f%f%f", &n1, &n2, &n3);
12.        m = (n1 + n2 + n3) /3;
13.        printf ("Media: %.2f\n", m);
14.        cont = cont + 1;
15.
16.    }
17.    system ("PAUSE");
18. }
```

Na animação a seguir, é possível observar a execução passo a passo do algoritmo para o cálculo da média dos 10 alunos.

Fonte:

Outra forma de utilizarmos a estrutura de repetição enquanto é quando não sabemos, ao certo, o número de repetições. Nesse caso, não há a necessidade da utilização de uma variável como contador, em contrapartida, a variável utilizada na condição deve ser controlada dentro do laço. Tal controle é feito, normalmente, por leitura dessa variável.

## Exemplo 2

Para exemplificar esta forma de utilização do enquanto, no exemplo a seguir é feito cálculo do somatório (soma acumulada) de apenas número positivos, os quais deverão ser definidos pelo usuário (por leitura). O somatório é feito enquanto o usuário digitar um número positivo e, quando ele digitar um número negativo, o laço é encerrado. Por fim, o valor do somatório é mostrado no final do algoritmo.

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. main () {
4.     int num, soma;
5.     soma = 0;
6.     printf ("Digite um numero inteiro:");
7.     scanf ("%d", &num);
8.     while (num >= 0) {
9.
10.        soma = soma + num;
11.        printf ("Digite um numero inteiro:");
12.        scanf ("%d", &num);
13.
14.    }
15.
16.    printf ("Somatorio: %d", soma);
17.    system ("PAUSE");
18. }
```

Na animação a seguir, é possível observar a execução passo a passo do algoritmo para o cálculo do somatório dos números positivos.

◀ 13 / 01 ▶

Simulação da execução da estrutura de repetição enquanto em C - Exemplo 2

Fonte:

## Estrutura de Repetição Repita em C

É uma estrutura que realiza um teste lógico no final do laço. Ela é parecida com a estrutura enquanto. Esta estrutura tem o seu funcionamento controlado por decisão, isto é, uma condição, porém executa um conjunto de instruções *pelo menos uma vez* antes de verificar a validade da condição estabelecida

Na linguagem C é utilizada a instrução *do ... while*, que tem seu funcionamento em sentido contrário ao comando *while*, pois sempre processa um conjunto de instruções no mínimo uma vez até que a condição seja falsa. Se a condição for verdadeira, o laço continua; se a condição for falsa, o laço é encerrado.

Vale ressaltar que, na Linguagem C, o resultado da avaliação da condição é interpretado de maneira inversa ao Diagrama de Blocos/Português Estruturado:

- Se a condição for *verdadeira*, o laço continua.
- Se a condição for *falsa*, o laço é encerrado.

O repita é utilizado quando:

- Não sabemos o número de repetições.
- Quando os comandos devem ser executados pelo menos uma vez. Antes da expressão ser avaliada possuímos uma expressão que deve ser avaliada para que os comandos da estrutura sejam executados

Para utilizar o comando *repita*, devemos utilizar o conjunto de instruções *do ... while* da seguinte forma:

```
1. do {  
2.  
3.     instruções;  
4.  
5. } while ( <condicao> );
```

## Exemplo 3

No exemplo a seguir é feita a leitura de uma série de valores, representando os preços dos itens comprados por cliente. Ao final, é apresentado o total a ser pago. Para finalizar a entrada de dados, o usuário deve informar o valor zero. Neste caso, não foi preciso utilizar um contador, pois o término do laço depende do preço digitado pelo usuário.

```
1. #include <stdio.h>  
2. #include <stdlib.h>  
3. int main () {  
4.     float p, s;  
5.     s = 0;  
6.  
7.     do {  
8.  
9.         printf ("Preco: ");  
10.        scanf ("%f", &p);  
11.        s = s + p;  
12.  
13.    } while (p != 0);  
14.  
15.    printf ("Total: %.2f\n", s);  
16.    system ("PAUSE");  
17. }
```

Na animação a seguir, é possível observar a execução passo a passo do algoritmo do Exemplo 1.

« 14 / 01 »

Simulação da execução da estrutura de repetição repita em C - Exemplo 3

Fonte:

## Exemplo 4

No exemplo a seguir, é feita a leitura da variável x por cinco vezes e, em seguida, é feita a multiplicação de x por 3, colocando-a na variável de resposta r, que é exibida a cada vez.

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. int main () {
4.     int x, r, cont;
5.     cont = 1;
6.
7.     do {
8.
9.         printf ("Digite o valor de x: ");
10.        scanf ("%d", &x);
11.        r = x * 3;
12.        printf ("R: %d ", r);
13.        cont = cont + 1;
14.
15.    } while (cont <= 5);
16.
17.    system ("PAUSE");
18. }
```

Na animação a seguir, é possível observar a execução passo a passo do algoritmo do Exemplo 4.

《 26 / 01 》

Simulação da execução da estrutura de repetição repita em C - Exemplo 4

## Estrutura de Repetição Para

Os laços que possuem um número finito de execuções poderão ser processados por meio de estrutura de laço para. A estrutura *for* tem o seu funcionamento controlado por uma variável denominada contador. Sendo assim, pode executar um determinado conjunto de instruções certo número de vezes. Além disso, este tipo de estrutura prevê uma condição e possui limites fixos.

A estrutura *for* é utilizada quando precisamos repetir um conjunto de comandos um número pré-definido de vezes. Ela deve ser utilizada da seguinte forma:

```
1. for (varControle = Inicio; varControle <= Fim; Incremento) {  
2.  
3.     instruções;  
4.  
5. }
```

Em que:

- *varControle*: variável de controle;
- *Inicio*: é o valor inicial da variável *varControle*;
- *Fim*: é o valor final da variável *varControle*, ou seja, o valor até o qual ela vai chegar. Nesse caso, enquanto a condição "*varControle* <= *Fim*" for



verdadeira, o bloco de instruções será executado.

- **Incremento:** é o valor do incremento dado à variável `varControle`. Em geral, o incremento é de 1 unidade na variável de controle e a instrução para que isso ocorra é `varControle++`.

Possuímos, então, um laço com contador de forma compacta, em que sempre temos uma inicialização da variável de controle (`varControle`), um teste para verificar se a variável atingiu o limite e um acréscimo na variável de controle após cada execução do bloco de repetição.

## Exemplo 5

No exemplo a seguir é feito o cálculo da média aritmética para 5 alunos. A cada iteração (passo) da estrutura *for*, é feita a leitura das 3 avaliações (N1, N2 e N3) e, em seguida, o cálculo da média. É possível observar que a variável de controle é definida como *cont*, a qual começa em 1 e termina em 5. A cada iteração do laço é feito o incremento de 1 na variável *cont*.

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. int main () {
4.     int cont;
5.     float n1, n2, n3, m;
6.
7.     for (cont = 1; cont <= 5; cont++) {
8.
9.         printf ("Digite 3 notas \n");
10.        scanf ("%f%f%f", &n1, &n2, &n3);
11.        m = (n1 + n2 + n3) /3;
12.        printf ("Media: %.2f\n", m);
13.
14.    }
15.
16.    system ("PAUSE");
17. }
```

Na animação a seguir, é possível observar a execução passo a passo do algoritmo do Exemplo 5.

« 21 / 01 »

Simulação da execução da estrutura de repetição Para em C - Exemplo 5

## Exemplo 6

No exemplo a seguir é feito o cálculo do somatório dos números inteiros de 1 até N, onde N é um valor definido pelo usuário. É possível observar que a variável de controle é definida como cont, a qual começa em 1 e termina em N. A cada iteração do laço é feito o incremento de 1 na variável cont.

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. int main () {
4.     int cont, N, soma;
5.     soma = 0;
6.     printf ("Digite o valor de N:");
7.     scanf ("%d", &N);
8.
9.     for (cont = 1; cont <= N; cont++) {
10.
11.         soma = soma + cont;
12.
13.     }
14.
15.     printf ("Somatorio: %d", soma);
16.     system ("PAUSE");
17. }
```

Na animação a seguir, é possível observar a execução passo a passo do algoritmo do Exemplo 6.

« 12 / 01 »

Simulação da execução da estrutura de repetição Para em C - Exemplo 6

## Quiz

Exercício Final

Implementação de estruturas de repetição em C

INICIAR ➤

## Referências

FORBELLONE, A. L. V; EBERSPACHER, H. F. Lógica de Programação: a construção de algoritmos e estruturas de dados (3a. edição). São Paulo: Prentice Hall, 2005.

RISSETTI, G.; PUGA, S. Lógica de Programação e estruturas de dados, com aplicações em Java. São Paulo: Prentice Hall, 2003.

SOUZA, M. A. F. e outros. Algoritmos e lógica de programação: um texto introdutório para Engenharia (2a. edição). São Paulo: Cengage Learning, 2011.



Avalie este tópico



ANTERIOR

Estruturas de repetição: para

Biblioteca

([https://www.uninove.br/conheca-](https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/)

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site



Índice

Ajuda?  
PRÓXIMO:  
([https://ava.un](https://ava.uninove.br/ava/estrutura-de-repeticao-encadeada/)

Estruturas de repetição encadeadas

© Todos os direitos reservados

