



# Estruturas de Repetição

## Introdução

Na programação de computadores, uma estrutura de repetição é uma sequência de instruções que é repetida continuamente até que uma determinada condição seja alcançada. Normalmente, um determinado processo é feito, como obter um item de dados e alterá-lo, e então alguma condição é verificada, como se um contador atingiu um número prescrito. Se não tiver, a próxima instrução na sequência é uma instrução para retornar à primeira instrução na sequência e repetir a sequência. Se a condição foi alcançada, o fluxo do programa será direcionado para a próxima instrução sequencial ou desvia para fora do loop. Uma estrutura de repetição é uma ideia fundamental de programação que é comumente usada na escrita de programas.

Deste modo, nesta aula estudaremos os principais conceitos relacionados a estrutura de repetição, bem como as diferentes estruturas e seus impactos no controle de fluxo de código.

## Objetivos da aula

- Compreender o conceito de controle de fluxo de código.
- Reconhecer os tipos de estruturas de repetição.
- Compreender cada tipo de estrutura de controle de fluxo.

- Diferenciar as estruturas de repetição for, while e aninhada.




## Resumo

Considere o seguinte caso: você deseja somar os três primeiros números inteiros maiores que zero. Bem, isso é simples, basta fazer soma = 1 + 2 + 3 (lê-se: soma recebe um mais dois mais três). Até certo ponto sim, é simples, porém, e se você for solicitado a somar os mil primeiros números inteiros positivos? Agora, considere que seu chefe pediu para você criar um programa que verifique qual o maior número primo existente (com o objetivo de usar este primo em uma chave criptográfica, por exemplo). Como você faria? Certamente a pura programação sequencial não é a melhor resposta, pois imagine você somando 1000 números manualmente! Isso levaria muito tempo.

Para resolver este problema, podemos construir um algoritmo que, tendo que repetir uma determinada instrução (ou conjunto de instruções), podemos iterá-la, ou seja, executar uma quantidade fixa de vezes até que uma determinada condição seja alcançada. Para tanto, temos duas estruturas de repetição que podem ser consideradas: while e for.

A estrutura de repetição while diz que: enquanto uma dada condição for verdadeira (condição de parada), o conteúdo de seu bloco (aquele que está em recuo entre chaves) será executado (MANZANO, 2016). Veja um exemplo de um laço while no código 1 Java a seguir.

1	contador = 1
2	
3	while (contador < 10){

4	<code>System.out.println("Valor do contador é " + contador);</code> 
5	<code>contador++;</code>
6	<code>}</code>
7	<code>System.out.println("Fim");</code>

Código 1 – Exemplo de aplicação da estrutura de repetição while

Fonte: Elaborado pelo autor

Observe no código que a variável *contador* é inicializada na linha 1 com o valor 1. Esta inicialização ocorre antes de iniciar o laço de repetição. Na linha 3 temos a estrutura de repetição while com a condição de parada *contador < 10*, ou seja, enquanto o valor da variável contador for menor do que 10, o conteúdo interno (linhas 4 e 5) será executado. Deste modo, será apresentado o valor da variável contador (linha 4) e, em seguida, é somado mais 1 no valor atual da variável contador, a esta ação de somar mais 1 nós chamamos de incremento.

Observe também que o conteúdo interno do while está afastado para a direita, ou seja, está indentado e organizado entre chaves. Como na estrutura condicional if, no while é necessária esta chave de abertura e de fechamento para indicar que o conteúdo está interno, ou seja, pertence ao escopo da função.

Ainda analisando o código 1, observe que a instrução da linha 7 só será executada após o laço de repetição concluir a execução. Mas, e se a variável *contador* na linha 1 inicia-se em 10? Neste caso, 10 é menor do que 10? Não, então as linhas 3, 4 e 5 não seriam executadas e o fluxo do código continuaria na linha 7.

Outra estrutura de repetição é o laço for, o qual segue o mesmo princípio do laço anterior, realizando um loop até que se atinja uma condição de parada. O código 2 a seguir apresenta um exemplo do laço for, em seguida sua sintaxe é detalhada.


1	System.out.println("Laço for");
2	
3	//início do laço for
4	for(int i=0; i<10; i++)
5	{
6	System.out.println("Valor : " + i);
7	}

Código 2 – Exemplo de aplicação da estrutura de repetição for

Fonte: Elaborado pelo autor

Observe pelo código 2 que a partir da linha 4 foi definido o laço de repetição for. Sua estrutura, assim como em outras linguagens é: a palavra reservada for, com os atributos:

- **inicialização do laço** (neste caso, uma variável inteira i começando com o valor zero). Esta variável pode ter qualquer nome, respeitando sempre os critérios de nomenclatura;

- **condição de parada** (neste caso, enquanto o laço for menor que 10). A condição de parada pode utilizar os operadores relacionais , de maior, menor, maior ou igual a ou menor ou igual a;
- **incremento ou decremento** (neste caso, estamos incrementando o laço de 1 em um, afinal,  $i++$  é o mesmo que  $i = i + 1$ ).


Note então que toda estrutura de repetição, seja ela for ou while, necessita de ter uma condição de parada, caso contrário, aquele escopo de código ficará em execução por tempo indeterminado, logo, se você esquecer de colocar uma condição de parada, certamente terá grandes problemas com seu programa.

Mas afinal, qual a diferença ou quando usar o while ou o for? Tanto o loop for quanto o while são usados para executar as instruções repetidamente enquanto o programa é executado. A principal diferença entre o loop for e o loop while é que, no caso do loop for, o número de **iterações é conhecido**, enquanto no caso do loop while, o número de **iterações é desconhecido** e a instrução será executada até que a condição seja comprovada como falsa (FORBELLONE, 2005).

Sendo assim, tanto o loop for quanto o while carregam diferentes métodos de execução e podem ser úteis se forem usados de acordo com seu mérito.

Também podemos utilizar o conceito de estrutura de repetição aninhada, que é uma estrutura lógica usada na programação de computadores onde uma estrutura de repetição (loop) está situada dentro do corpo da outra estrutura de repetição.

Em um loop aninhado, a primeira iteração do loop externo faz com que o loop interno seja executado. O loop interno se repete em função do loop externo, logo, se o loop externo for executado 3 vezes, o loop interno será


repetido 3 vezes também (FORBELLONE, 2005). Aqui cabe um detalhe, quando o loop interno é concluído, o loop externo é executado  sua segunda iteração, acionando o loop interno novamente, o qual inicia tudo novamente até que seja concluído, até que os requisitos para o loop externo sejam completos (MANZANO, 2016). O código 3 a seguir apresenta um exemplo de estrutura de repetição aninhada, neste caso, temos um loop for externo e um loop for interno, porém, podemos ter um loop while com um loop while interno, um loop while com um loop for interno, entre outras combinações.

1	for(int i=0; i<3; i++)
2	{
3	for(int j=0; j<=3; j++)
4	{
5	System.out.print(j);
6	}
7	System.out.println("\n");
8	}

Código 3 – Exemplo de aplicação da estrutura de repetição aninhada

Fonte: Elaborado pelo autor

Observe no código 3 que, quando o laço da linha 1 inicia, o valor da variável i passa a ser zero. Em seguida, na linha três inicia a execução do

laço for, onde a variável `j` recebe o valor zero. O próximo passo é mostrar o valor de `j` por meio da instrução da linha 5. 

A instrução da linha 5 será executada 4 vezes (para `j` igual a zero, igual a 1, igual a 2 e igual a 3). Assim que a variável `j` valer 4, o laço for interno será finalizado e o fluxo do código volta para a linha 1. Lembra que o valor da variável `i` era 0? Agora ele será incrementado pelo `i++`, passando a ser 1. Novamente tudo aquilo que foi descrito anteriormente sobre a linha 5 será repetido, ou seja, esta linha será executada 4 vezes. Este processo se repete enquanto o valor da variável `i` for menor que 3.

Por fim, cabe ressaltar que um loop ou estrutura de repetição é um comando que tende a se repetir para obter o resultado desejado. Em outras palavras, um comando de programação que se repete em um número conhecido de vezes (quando temos o loop for) ou o número desconhecido de vezes (quando temos o laço while) para atender a certas condições.

## Tópicos avançados

As estruturas de repetição são utilizadas em diferentes cenários e certamente você se deparará com muitas delas em sua jornada como programador.

Considere criar um programa que seja responsável por receber o nome dos cinco melhores filmes ou séries que você já assistiu. Observe que para cada filme você necessitará de uma variável diferente para armazenar os valores na memória do computador. Após digitar o nome dos filmes ou séries, experimente mostrá-los na tela. Importante: somente com a prática você obterá a proficiência em programação.



## Referência Bibliográfica

MANZANO, J. A. N. G. **Algoritmos: lógica para desenvolvimento de programação de computadores**. 28. ed. São Paulo: Érica, 2016.