

Agenda 05: Estruturas de Repetição 1

Conceitos trabalhados:

Sumário

- Momento de Reflexão
- Por que Aprender?
- Para Começar o Assunto
- Mergulhando no Tema
- Ampliando Horizontes
- Resumindo o Estudo

Momento de Reflexão

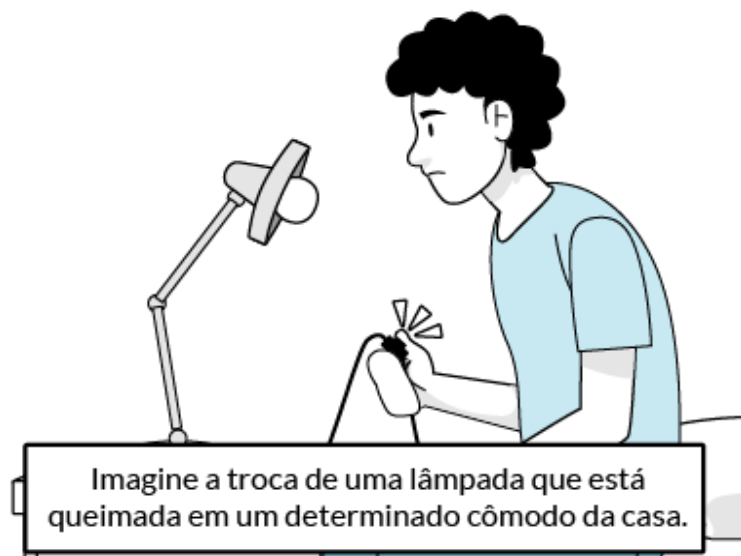
Olá, estudante ;)

Boas-vindas à quinta agenda do Módulo 1, em que você iniciará seus estudos sobre as **Estruturas de Repetição**. Você já ouviu esta frase do jurista e economista alemão, Max Weber?



“O homem não teria alcançado o possível se, repetidas vezes, não tivesse tentado o impossível.”

A frase acima sinaliza o objeto de estudo desta aula: a **Repetição**. Mas antes que você ache estranho falar de repetição na lógica de programação, e para entender melhor o conceito, vamos fazer uma analogia com o cotidiano:





Da mesma forma, na lógica de programação, também temos a necessidade de “experimentar” várias opções na busca pela opção mais acertada. A forma como isso é feito, a estrutura utilizada e em quais momentos, são os temas desta agenda.

Por que Aprender?

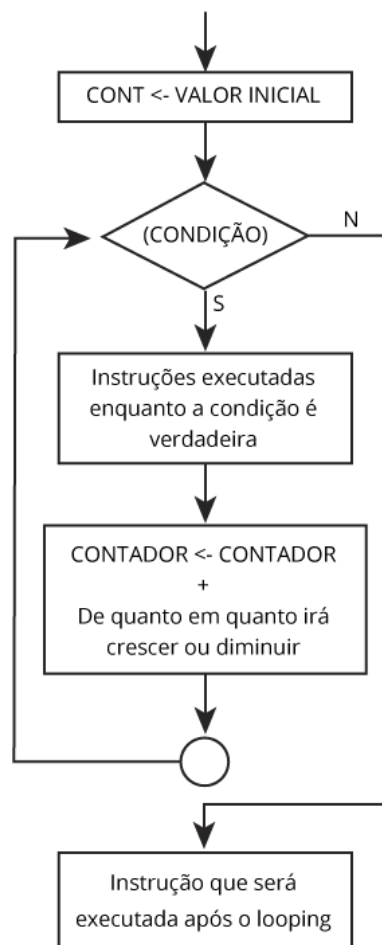
Imagine que você está organizando uma festa e precisa reproduzir uma lista de 10 músicas durante toda a balada. Você pode escolher repetir manualmente essa lista ou deixar programado para que as músicas fiquem tocando até o momento em que você pare a reprodução da mesma.

Os programas computacionais estão por toda parte. Praticamente todos os estabelecimentos, sejam eles comerciais, industriais, acadêmicos ou qualquer outro, fazem uso de programas que **automatizam suas rotinas**. Para ser um indivíduo competitivo no mercado de trabalho e se tornar “empregável” no desenvolvimento de softwares computacionais, é extremamente importante que você conheça o modo como o computador entende e implementa as estruturas de repetição dentro da lógica de programação.

Para Começar o Assunto

Você viu, na última agenda, que é possível “instruir” o computador a prosseguir em um determinado programa a partir de uma condição. Ele pode identificar se a condição é verdadeira e executar determinada(s) instrução(ões); ou verificar se é falsa e seguir pelo caminho contrário. Essa rotina, porém, acontece **uma única vez**. O programa seleciona a opção (falso ou verdadeiro) e executa as instruções uma vez.

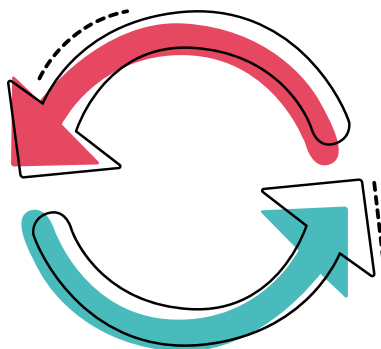
Mas e se for preciso executar essas instruções mais de uma vez? Para responder a essa pergunta, utilizamos as Estruturas de Repetição que, como o próprio nome diz, **repetem um bloco de instruções enquanto uma condição for verdadeira**; ou então, **repetem um bloco de instruções enquanto uma dada condição for falsa**. Veja só o fluxograma a seguir:



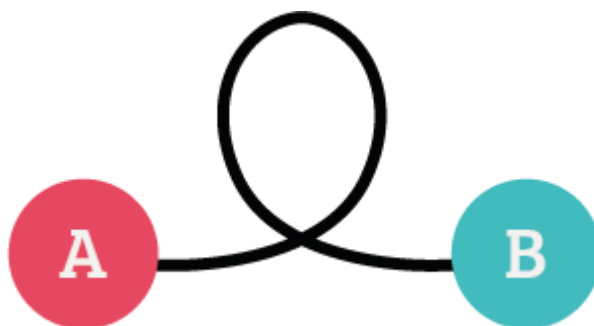
Pareceu complicado? Não! Dê uma olhada nos materiais a seguir para entender direitinho como funciona.

Mergulhando no Tema

Estruturas de repetição



Desvios em Loop



Na agenda anterior, você estudou as Estruturas de Decisão, também conhecidas por **desvios condicionais**.

Além dos desvios condicionais, é possível criar **desvios em loop ou repetição**, ou seja, repetir trechos do algoritmo sob determinada condição e controlar a forma com que serão executados. Nesta agenda você conhecerá a primeira Estrutura de Repetição, o comando para...fim-para. Na próxima unidade você conhecerá outros dois laços. Preparado?

Laço de Repetição “para...fim-para”

Para iniciar seus estudos, assista à videoaula do professor Sandro Valerius, que apresenta esse Laço de Repetição.

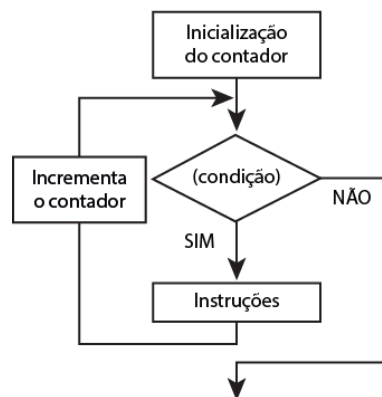


Você viu que o comando “para...fim-para” permite que uma **variável realize a contagem do número de repetição a executar**, conforme a indicação inicial e final dessa contagem e, também, indique o formato em que essa tarefa será realizada. Observe o quadro a seguir e veja as sintaxes em pseudocódigo e em Java, perceba como são parecidas:

PSEUDOCÓDIGO

```
Para {variável} = <valor inicial>
    até <valor final>
    passo <argumento>
    faça {comando(s)}
fim-para
```

FLUXOGRAMA



JAVA

```
for (int i=0; i<10; i++){
    System.out.println(i);
}
```

Observe que no Fluxograma aparece a palavra **contador** e tanto no Pseudocódigo quanto no Java não aparecem. Calma! O contador está lá, porém de outra forma. Veja que no Pseudocódigo há {**variável**} e no Java aparece i. Ambos são os contadores que aparecem no Fluxograma.

Conhecendo um exemplo

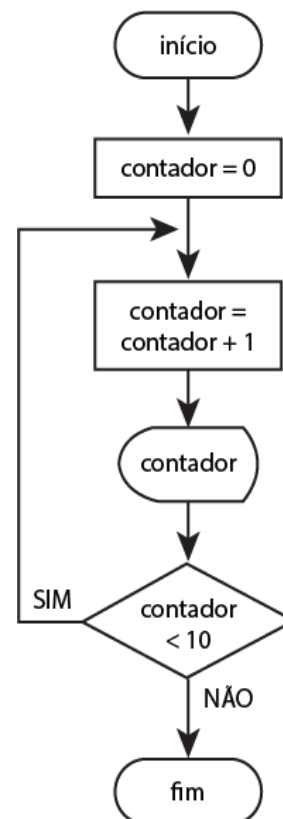
A execução é muito fácil. Para facilitar a compreensão, veja um exemplo de um programa que mostra ao usuário os números de 0 a 9 na tela:



PSEUDOCÓDIGO

```
Programa mostrar números de 0 até 9
Declare
    contador como inteiro
Início
    contador <- 0;
    Para contador de 0 até 9 passo 1 faça
        Escreva (contador)
    Fim para
Fim.
```

FLUXOGRAMA



Apenas para você entender melhor, o pseudocódigo inicia-se com a declaração da variável **contador** com tipo de dados inteiro, ou seja, esta variável irá receber apenas números inteiros. Após a leitura desta variável, iniciada com o valor 0, ocorre o incremento de mais 1 por meio da instrução **passo** até o valor 10. Esse tipo de estrutura de repetição pode ser usado toda vez que houver a necessidade de repetir trechos finitos, em que se conhece os valores inicial e final.

Dessa forma serão impressos os números de 0 a 9, isso porque a condição é que o contador seja menor que 10 ($\text{contador} < 10$).

No exemplo anterior, tanto no Pseudocódigo quanto no Fluxograma, não fazemos a contagem até o número 10. Por que? Você estudou em unidades anteriores os operadores (aritméticos, comparação e lógicos), então é momento de retomar conceitos.

Observe que a condição apresentada no exemplo é:

- Pseudocódigo: Para contador \leftarrow 0 até 09.
- Fluxograma: contador < 10.

Isso porque de 0 até 9, temos **10 contagens**:

Quando o contador está em 0, temos a 1ª contagem (o programa realiza o trecho do algoritmo pela 1ª vez)

Quando o contador está em 1, temos a 2ª contagem.

Quando o contador está em 2, temos a 3ª contagem.

E assim sucessivamente, até que, quando o contador chega em 9, o programa estará contando pela 10ª vez.

Se estivéssemos usando o operador <= (igual) em lugar de < (menor), então o programa totalizaria **11 contagens**.

Veja agora o código em Java:

```
package lacos;
import javax.swing.JOptionPane;
public class lacofor {
    public static void main(String[] args){
        //declaração das variáveis.
        int i; //variável que será utilizada para o laço de
repetição

        for (i=0; i<10; i++){
            JOptionPane.showMessageDialog(null, i); // saída de
dados.
        } // fecha o laço de repetição for.

    } // fecha o método.

} // fecha a classe
```

Correspondendo ao Pseudocódigo informado anteriormente, no Java, são declaradas as variáveis que serão utilizadas. Neste caso, apenas o `i` como inteiro. Em seguida, inicia-se o laço de repetição **for** com a variável inicial 0 (zero), variável final 9 (nove) e o incremento com 1. Ou seja, o resultado irá mostrar as sequências dos números 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9.

Exercitando o Laço de Repetição "para...fim-para"

Agora, pense: como poderíamos elaborar um programa utilizando a estrutura para...fim-para e que mostre ao usuário os números de 01 até 10 na tela?



Agora, dê uma olhada em um pseudocódigo que resolve essa questão:

Programa mostrarnumeros


```
declare  
    num como inteiro  
  
Início  
Para num <= 1 até 10 passo  
1 faça  
    escreva(num) Fim-para  
  
Fim.
```

Veja, então, o Java:

```
public class ex02 {
```

```
public static void main(String[] args){  
  
    System.out.println("Os números de 0 a 10 são:");  
  
    for (int i=0; i<=10; i++){ //Laço de repetição for.  
        System.out.println(i); // saída de dados  
    }  
}
```

E este é o resultado:



The screenshot shows a Java IDE window with the 'Console' tab selected. The title bar indicates the application is 'ex02 [Java Application]' located at 'C:\Program Files\Java\jre1.8.0_151\'. The console output displays the text 'Os números de 0 a 10 são:' followed by the numbers 0 through 10, each on a new line.

```
<terminated> ex02 [Java Application] C:\Program Files\Java\jre1.8.0_151\  
Os números de 0 a 10 são:  
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

Observe que apenas usamos a sintaxe **System.out.println** nas linhas 5 e 7 do código para que a informação apareça no Console. Além disso, a estrutura do **for** iniciou com valor 0 e foi até 10 com passos 1 (linha 6 do código), ou seja, ele andou de um em um, como mostra a tela do resultado.

O Laço de Repetição “para...fim-para” é muito simples e fácil de usar. Porém se você quiser incrementar o seu uso, é possível utilizá-lo junto com a estrutura “se... senão...fim_se”.

Exercitando os Laços de Repetição e de Decisão

Agora, imagine que você precise calcular a média de 30 alunos e informar a situação de cada um deles. Ou seja, além de calcular e mostrar a média, o programa deverá e indicar se cada um dos 30 alunos está "Aprovado" ou "Reprovado".



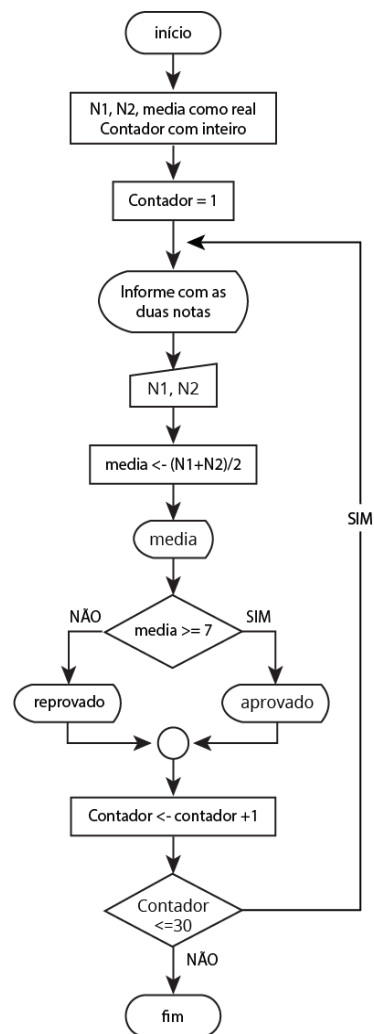
Agora, analise o pseudocódigo e o fluxograma a seguir para conferir se você acertou

PSEUDOCÓDIGO

```

Programa mostrar media e situacao
Declare
    n1, n2, media como real
    contador como inteiro
Início
    Para contador = 1 até 30 passo 1 faça
        escreva ("Informe sua primeira nota:")
        leia (nota1)
        escreva ("Informe sua primeira nota:")
        leia (nota2)
        media ← (n1 + n2)/2
        escreva ("A sua média é: " , media)
        se (media >=7,0) então
            escreva ("Aprovado")
        senão
            escreva ("Reprovado")
    Fim-se
Fim-para
Fim.
  
```

FLUXOGRAMA



Confira o programa em Java:

```

import java.util.Scanner;
public class ex06 {

    public static void main(String[] args){
  
```

```

        double n1, n2, media;
        n1 = 0;
        n2 = 0;
        // utilizando Scanner como visto nas agendas
anteriores.
        Scanner ler = new Scanner(System.in);
        for (int i=0; i<=30; i++){ //laço de repetição para
os 30 alunos)
            System.out.println("Digite a primeira nota:");
            n1 = ler.nextDouble();
            //usuário irá digitar a nota e este comando irá
ler e armazenar na variável.
            System.out.println("Digite a segunda nota:");
            n2 = ler.nextDouble();
            media = (n1 + n2)/2; // cálculo da média.
            if (media >= 7.0){
                System.out.println("Sua situação é: Aprovada" +
media); }
            else {
                System.out.println("Sua situação é:
Reprovada" + media);
            }
        }
    }
}

```

Observe que utilizamos o Laço de Repetição **para** (linha 11) e o Laço de decisão (linha 18) para resolver o problema informado anteriormente. Além disso, utilizamos o Scanner (linhas 1, 10, 13 e 16) para que o usuário pudesse digitar as notas e o programa realizar todo cálculo.

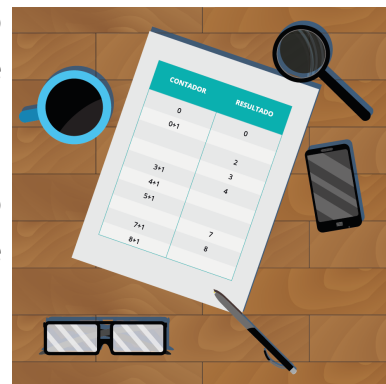
Repare que estamos utilizando diversos conceitos aprendidos nas unidades anteriores, ou seja, tudo que você está estudando é acumulativo. Procure utilizar todos os conceitos estudados para realizar um melhor programa que solucione o problema.

Teste de Mesa

Para que possamos nos certificar de que o algoritmo realizado está correto, antes de passar para a linguagem de programação (no nosso caso, Java) podemos testá-lo **simulando valores e verificar se o resultado é o esperado**. Esta simulação não é realizada no computador utilizando nenhum software, é realizada no papel. É muito simples: basta montar uma pequena tabela e começar a simular os valores utilizando o seu pseudocódigo.

Para exemplificar, vamos voltar ao primeiro algoritmo mostrado nesse material (Mostrar os números de 01 até 09 com passo 01) e aplicar o Teste de Mesa:

Para exemplificar, vamos voltar ao primeiro algoritmo mostrado nesse material (Mostrar os números de 01 até 09 com passo 01) e aplicar o Teste de Mesa:



CONTADOR	RESULTADO
0	0
0+1	1
1+1	2
2+1	3
3+1	4
4+1	5
5+1	6
6+1	7
7+1	8
8+1	9

Veja que você iniciou com zero e como o exercício pede o incremento, ou seja, precisa somar 1 ao resultado anterior.

E como foi dito anteriormente, como o valor final é $i < 10$, então iremos até o valor 9.

Revisando

Por fim, assista à videoaula do professor Rogério Silva, que retoma os conceitos que você estudou nessa agenda, especialmente o comando `for()`.

DS - Estruturas de Repetição - Comando for()



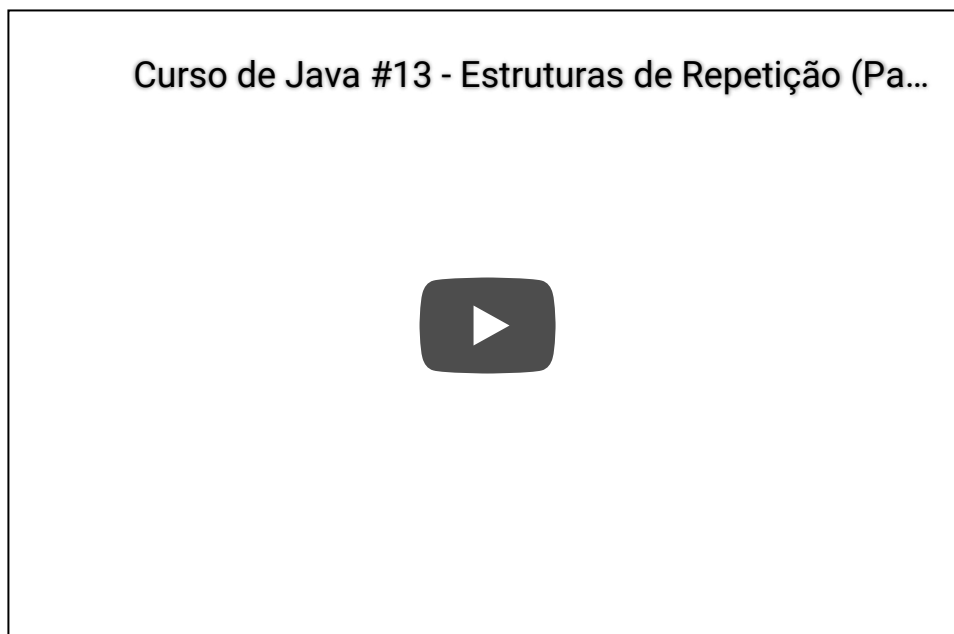
Ampliando Horizontes

Onde encontro mais informações sobre Estruturas de Repetição?

Para aprofundamento dos temas discutidos nesta agenda, veja os materiais indicados a seguir:

Curso de Java para Iniciantes

Para esta agenda, você pode assistir ao vídeo #13, que trata das Estruturas de Repetição.



Lógica de programação: a construção de algoritmos e estrutura de dados. FORBELLONE, André L. V; ELBERSPACHER, Henri Frederico. Editora Pearson, 2000.



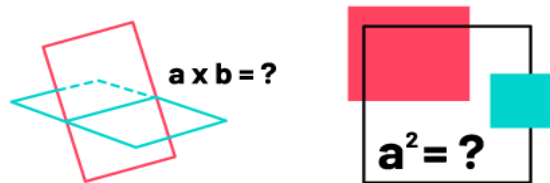
Lógica de programação e estruturas de dados com aplicações em Java. PUGA, Sandra; RISSETTI, Gerson. Editora Pearson, 2009.



Java para iniciantes. SCHILDT, Hebert. Editora Bookman, 2015.

Resumindo o Estudo

Nesta agenda, você teve seu primeiro contato com as Estruturas de Repetição e viu que elas são extremamente úteis na criação de programas que realizam atividades recorrentes. Você exercitou os comandos relacionados a essas estruturas desenvolvendo programas que auxiliaram Giovanna e Benício na realização de cálculos matemáticos, mas já sabe que o potencial dessas estruturas está relacionado à automatização de rotinas, em diversos contextos.



Na próxima agenda, você aprofundará ainda mais seus estudos sobre **Estruturas de Repetição!**

Até lá!

