

Agenda 06: Estruturas de Repetição 2

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 sexta agenda do Módulo 1, em que você dará continuidade aos seus estudos sobre **Estruturas de Repetição**. Para começar, leia a tirinha:



(Fonte: <http://bichinhosdejardim.com/repeticao/> acesso em dezembro/2016)

Você reparou que fazemos diversas atividades repetidas ao longo da vida? E no **seu cotidiano**, consegue lembrar de ações que se repetem?

Todo dia você acorda em um determinado horário, come algo, sai de casa e pega ônibus, metrô ou trem (ou até os três!) para chegar à escola. Então, você assiste às aulas, faz as tarefas indicadas, volta da escola e segue com suas atividades rotineiras. Se pararmos um pouco e pensarmos quantas atividades repetimos...

Para o computador é a mesma coisa! Mas, ainda bem, há a Lógica de Programação e as Estruturas de Repetição que, somadas, ajudam a desenvolver diversos algoritmos para que as atividades rotineiras fiquem mais fáceis e práticas. Você já iniciou seus estudos sobre estruturas de repetição na agenda anterior, agora conhecerá dois novos Laços de Repetição.

Por que Aprender?

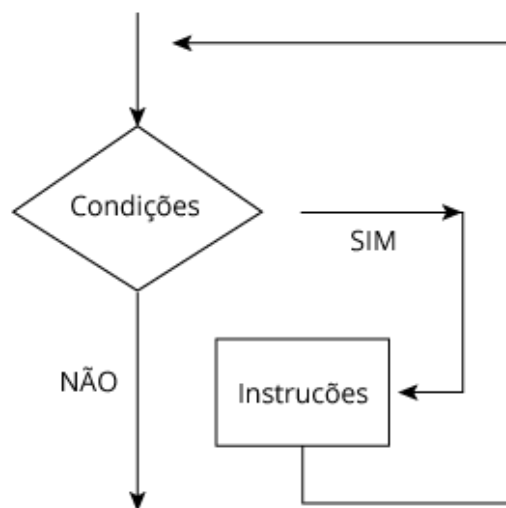
Na agenda passada você conheceu a Estrutura de Repetição **para...fim-para**. Dando sequência, agora você estudará os outros dois tipos de Laço de Repetição: **enquanto... fim-enquanto** e **repita... até que**, que têm características diferentes e irão facilitar bastante na solução de diversos problemas que você encontrará ao longo da Programação. Com isso, finalizamos os comandos das Estruturas de Repetição, que são muito usuais e práticos na programação e serão de grande valor nas agendas e módulos seguintes.

Para Começar o Assunto

Você já sabe que estudará duas novas estruturas de repetição, mas qual a diferença entre elas? Em quais situações sua utilização é mais adequada? Conheça-as a seguir.

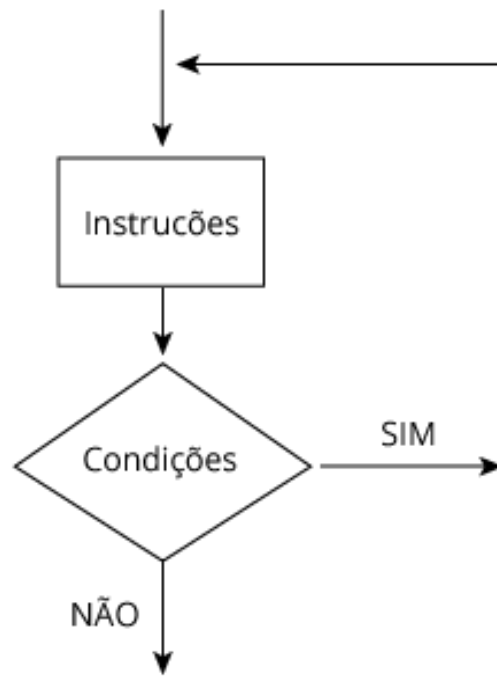
1. Enquanto... fim-enquanto

Essa estrutura, também conhecida como **enquanto** (while), é habitualmente utilizada quando não se sabe o número de repetições que ocorrerão. Ela só encerra sua execução quando a condição é satisfeita. Veja o fluxograma que representa essa informação:



2. Repita...até que

Essa estrutura, também conhecida por **do...While**, é muito parecida com a anterior e, por conta disso, existem vários programadores que não a utilizam. Ela também é aplicada quando não se conhece o número de repetições que ocorrerão e, do mesmo modo, só se encerra quando uma condição é satisfeita. Mas qual a diferença entre as duas estruturas? **Repita...até que** é **testada por último**, fazendo com que o trecho seja executado ao menos uma vez. Veja o fluxograma abaixo e compare com o apresentado para a estrutura anterior.

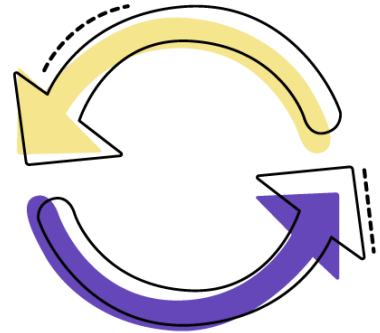


Fique tranquilo, os materiais indicados a seguir aprofundarão seus estudos sobre as Estruturas de Repetição e permitirão que você pratique um pouco mais esses laços.

Mergulhando no Tema

Conhecendo dois novos Laços de Repetição

Na agenda anterior, você conheceu o comando “para...fim-para”, utilizado em situações em que você sabe exatamente o número vezes que as instruções deverão ser repetidas. Como exemplo, você estudou programas que realizam instruções repetidamente para exibir aos usuários os números de 0 a 9.

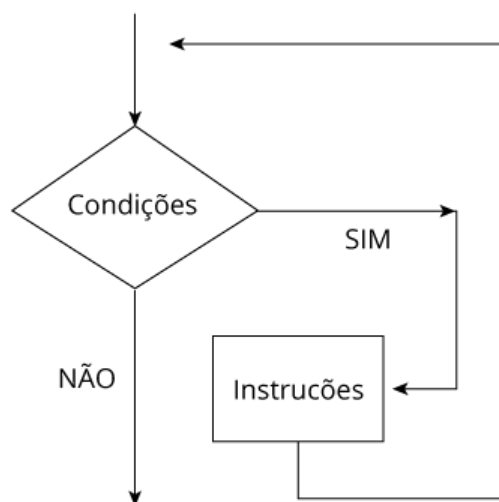


Mas, como fazer em situações nas quais não é possível definir quantas vezes a instrução deverá ser repetida?

Os **Laços de Repetição** abordados nessa agenda são utilizados nesses contextos. São comandos que só encerram suas repetições quando a condição é satisfeita. Dê uma olhada:

Enquanto... fim-enquanto

Essa estrutura, também conhecida por **enquanto** (while), avalia a condição e só então executa as instruções indicadas. Veja o fluxograma que representa o comando.



Agora, você conhecerá cada Estrutura de Repetição com maior profundidade.

Laço de Repetição “enquanto... fim-enquanto”

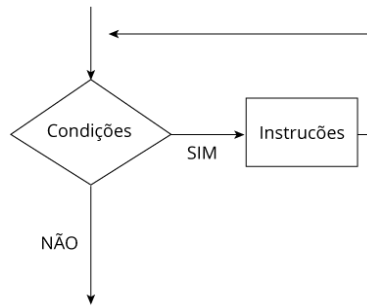
Este Laço de Repetição, como apresentado anteriormente, trabalha enquanto a condição for verdadeira e vai executando as instruções. Porém, a condição sendo falsa, ele sai do loop e vai

para o próximo comando na programação.

PSEUDOCÓDIGO

```
Enquanto <condição> faça  
    <Comando>  
    <Comando>  
    <Comando>  
Fim enquanto
```

FLUXOGRAMA



JAVA

```
while (condição){  
    instrução  
}
```

Agora que você conheceu a definição e a estrutura do laço, veja o exemplo a seguir.

Elabore um Algoritmo, um Fluxograma e um Programa em Java que mostre todos os números menores que 10.



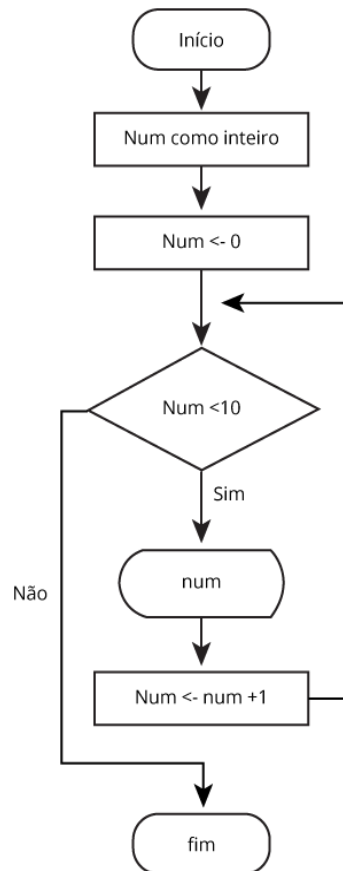
PSEUDOCÓDIGO

Programa ex

```

Declare
    num como inteiro
Início
    Num <- 0
    Enquanto (num < 10) faça
        escreva num
        num <- num+1
    fim-enquanto

```

FLUXOGRAMA

Agora, veja o programa em Java.

```

public class ex01 {

    public static void main(String[]args) {
        int num =0;
        while (num < 10) {
            System.out.println("Número " + num);
            num++;
        }
    }
}

```

Observe que na linha 5 temos a estrutura de repetição **enquanto** while tendo como condição a situação que o exercício colocou (números menores que 10). As linhas 6 e 7 apresentam os comandos que são executados dentro desta estrutura.

Como resultado, o computador apresenta:


```
<terminated> ex01 [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe
Número 0
Número 1
Número 2
Número 3
Número 4
Número 5
Número 6
Número 7
Número 8
Número 9
```

Você percebeu que a condição colocada acima foi “num < 10”, por isso o resultado apresentado são números de 0 até o 9.

Lembre-se sempre: o incremento não é automático

O Laço de repetição Enquanto...Fim-Enquanto **não possui incremento automático** como o laço Para...Fim...Para.

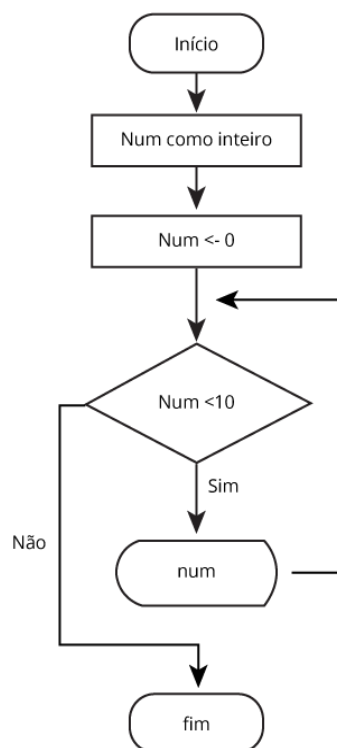
Fique atento a isso! O exemplo a seguir apresenta uma repetição infinita e demonstra as implicações da ausência de contador:

PSEUDOCÓDIGO

Programa ex

```
Declare
    num como inteiro
Início
    Num <- 0
    Enquanto (num < 10) faça
        escreva num
    fim-enquanto
```

FLUXOGRAMA

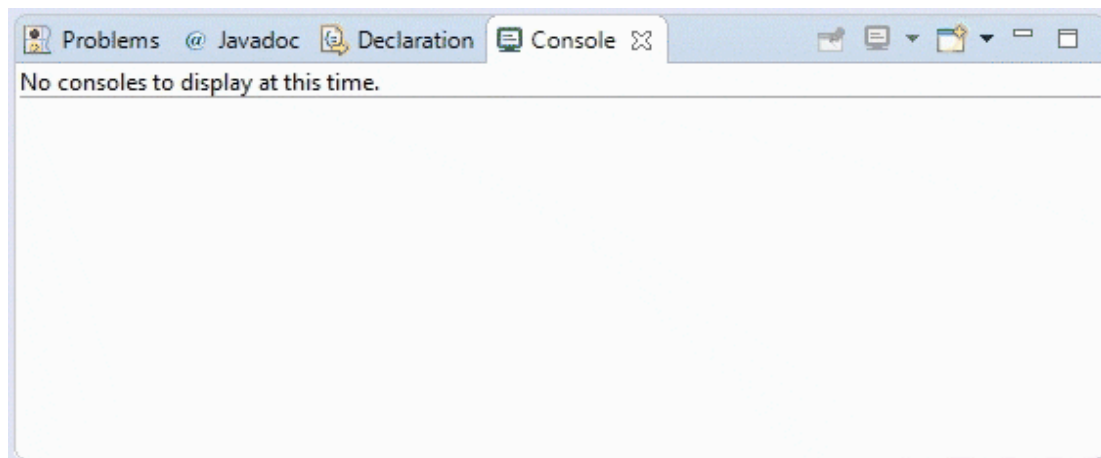


No Java:

```
public class exemploLoopInfinito {  
    public static void main(String[] args) {  
        int num = 0;  
        while (num < 10) {  
            System.out.println("Número " + num);  
        }  
    }  
}
```

Note que o programa é idêntico ao anterior com a exceção de que não temos o incremento do contador **num** (`num <- num+1`). Com isso, o contador fica eternamente com o valor de 0 (zero) e o programa nunca terá um fim.

Resultado:

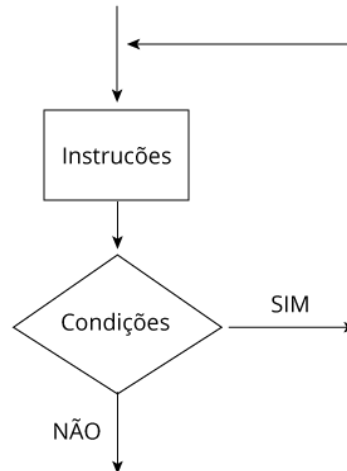


Laço de Repetição “repita... até que”

Como você já sabe, esta estrutura é parecida com a `while`, que você acabou de conhecer. A única diferença é que a condição dela é executada por último. Vamos aos detalhes para entender melhor?

PSEUDOCÓDIGO

Repita <condição> **faça**
 <Comando>
 <Comando>
 <Comando>
até que<Comando>

FLUXOGRAMA**JAVA**

```

do {
    instrução
}
while (condição)
  
```

Agora que você conheceu um pouco mais esta estrutura, vamos voltar ao exemplo anterior.



Elabore um Algoritmo, um Fluxograma e um Programa em Java que mostre todos os números menores que 10.

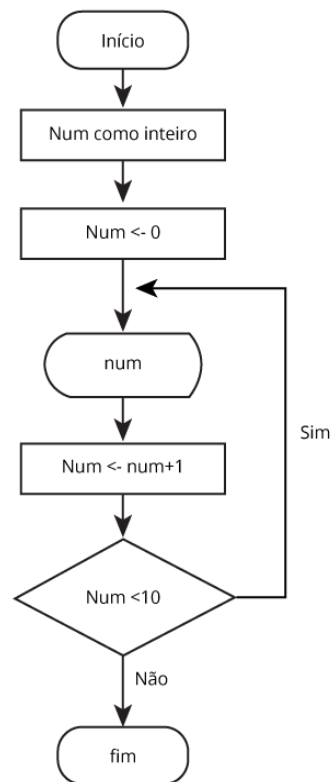
PSEUDOCÓDIGO

Programa ex01

```

Declare
    num como inteiro
Início
    repita
        escreva num
        num < num + 1
    até que (num < 10)
fim.

```

FLUXOGRAMA

Agora, veja o programa em Java:

```

public class ex01 {

    public static void main(String[]args) {
        int num =0;
        do{
            System.out.println("Número " + num);
            num++;
        }
        while (num < 10);
    }
}

```

Observe que, como o programa responde ao mesmo exercício, e o resultado é o mesmo. Porém, o programa foi construído utilizando outro Laço de repetição. Note que as linhas 6 até 10 apresentam a sintaxe do comando **repita até**.

Na videoaula a seguir, o professor Sandro Valérios retoma os conteúdos estudados até o momento.

DS - Estruturas de Repetição - Enquanto e repita



Utilizando comandos de repetição em conjunto com comando condicional if

Nas agendas anteriores você viu que é possível utilizar o comando **para** (for) em conjunto com o **se** (if) para atingir a solução de um problema na programação. Assim, como no caso do **para**, também podemos usar o **enquanto** (while) e o **repita até** (do while) em conjunto com o se (if) com o mesmo propósito.

Veja o exemplo a seguir, um programa de uma empresa financeira. Nele, o usuário informa seu cargo e salário e, a partir dessas informações, o programa informa quanto de empréstimo a pessoa pode obter.

Diretor = 30%

Gerente = 25%

Operacional = 20%

```
import java.util.Scanner;
    public class ex03 {

        public static void main(String[]args) {

            String nome, cargo;
            float sal, emprestimo=0;

            System.out.println("Por favor informe seu
nome");

            nome = new Scanner(System.in).nextLine();
```

```
        System.out.println("E agora informe seu  
cargo (Diretor, Gerente ou Operacional)");  
  
        cargo = new Scanner(System.in).nextLine();  
  
        while (cargo == "Diretor" || cargo ==  
"Gerente" || cargo == "Operacional") {  
            System.out.println("Cargo incorreto,  
por favor informe o cargo novamente");  
            cargo = new  
scanner(System.in).nextLine();  
        }  
  
        System.out.println("Agora informe o seu  
salário");  
  
        sal = new scanner(System.in).nextFloat();  
  
        if (cargo.equals("Gerente"))  
            emprestimo = sal*25/100;  
        else if (cargo.equals("Diretor"))  
            emprestimo = sal*30/100;  
        else  
            emprestimo = sal*20/100;  
  
        System.out.println("Olá" + nome);  
        System.out.println("Seu cargo é" + cargo);  
        System.out.println("Seu salário é" + sal);  
        System.out.println("Olá" + nome);  
        System.out.println("E você tem direito a  
pegar R$" + emprestimo + "de empréstimo");  
    }  
}
```

Você notou que além dos usos do Laço de repetição **while** (linhas 15 até 18), da estrutura de seleção **if** (linhas 23 até 28), foi utilizado, também, o **scanner** (linhas 1, 10, 13, 17 e 21)? Pois é, neste momento do curso você já tem diversos comandos em seu repertório, e deve utilizá-los conjuntamente para solucionar os desafios propostos.

Para finalizar, assista à videoaula do professor Rogério Silva, que sintetiza o conteúdo da agenda.

Ampliando Horizontes

Onde encontro mais informações sobre Estruturas de Repetição?

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

Estruturas de Repetição

Pode ser interessante que você faça uma revisão dos conceitos estudados na agenda anterior. Você pode assistir novamente às duas videoaulas a seguir:

Assistente de Desenvolvimento de Sistemas | M...



DS - Estruturas de Repetição - Comando for()



Curso de Java - Correção Exercícios Aula 17 (while, do-while, for)

Neste vídeo, Loiane Groner desenvolve alguns exercícios que utilizam os loops while, do-while e for.

Curso de Java #11 – Estruturas de Repetição (Parte 1)

Para esta unidade, assista aos vídeos #11 e #12, que tratam de Estruturas de Repetição.

Curso de Java #12 - Estruturas de Repetição (Pa...



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



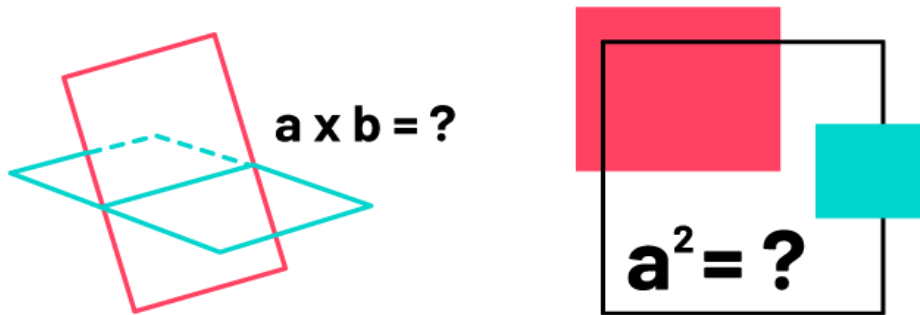
Lógica de programação e estruturas de dados com aplicações em Java. PUGA, Sandra; RISSETTI, Gerson. Editora Pearson, 2009.



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

Resumindo o Estudo

Nesta agenda, você completou seus estudos sobre as Estruturas de Repetição, conhecendo dois novos Laços de Repetição. Nas atividades desenvolvidas, você teve a oportunidade de resolver o desafio da semana anterior utilizando os comandos aprendidos nesta unidade. Agora você já tem domínio sobre comandos que permitirão desenvolvimento de programas mais concisos e organizados, e está pronto para avançar em seus estudos.



Na próxima agenda, você voltará sua atenção a um novo conceito: as variáveis do tipo **Vetores**.

Até lá!

