

L gica de Programac o

Aula 07 - Repeti o para



Apresentação da Aula

Na aula anterior, você conheceu a estrutura de repetição **enquanto**, caracterizada por ser uma estrutura **sem** a variável de controle. Nesta aula, você conhecerá a estrutura de repetição **para**, que, diferentemente, é uma estrutura **com** variável de controle. Conhecerá, também, exemplos de situações nas quais deve utilizar essas estruturas de repetição, de acordo com o algoritmo que pretende criar.



Objetivos

Aprender a definir a sintaxe relativa ao comando **para** em estruturas de repetição com variável de controle.

Definir o nome da variável de controle da estrutura **para**.

Definir o passo de incremento/decremento da estrutura **para**.

Distinguir quando utilizar as estruturas **para** ou **enquanto**.

Construir algoritmos por meio das estruturas de repetição **para** e **enquanto**.

Repetindo...

Na aula anterior, você conheceu a estrutura de repetição **enquanto**, definida como pré-testada e **sem** variável de controle, lembra? Agora, dando continuidade ao estudo das estruturas de repetição, você conhecerá a estrutura de repetição **para**, que também é uma estrutura pré-testada, porém, **com** variável de controle.

A estrutura **para** também está presente na maioria das linguagens de programação e sua utilização é comum em boa parte dos algoritmos que necessitam realizar a repetição de uma sequência de comandos.

Vale lembrar que as estruturas de repetição permitem executar tarefas iterativas (Iteração é o processo chamado na programação de repetição de uma ou mais ações.

 Não confunda com **N**teração, que é uma ação que ocorre entre dois ou mais indivíduos.) de forma automática, eliminando, assim, a frequência e duplicidade de comandos que são executados repetidamente.

Estrutura de Repetição com Variável de Controle: SEM indicação de passo de incremento

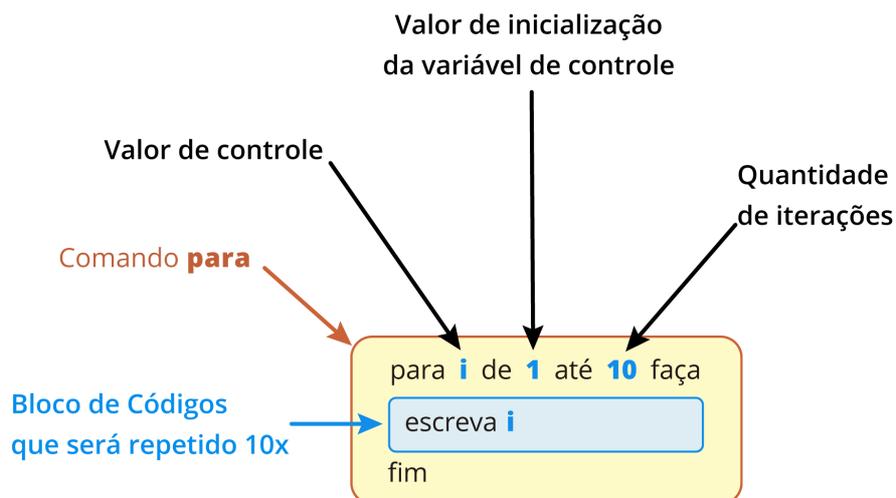
O comando **para**, assim como o comando **enquanto**, é também uma estrutura de repetição do tipo pré-teste, pois realiza a avaliação da condição antes da execução do bloco de comandos. A diferença entre essas duas estruturas de repetição diz respeito à presença da **variável de controle** existente na estrutura **para**.

A estrutura de repetição **para**, no **Potigol**, possui duas formas distintas. A primeira delas, mais simples, possui: a variável de controle, o valor de inicialização e a condição de parada (que também é nomeada por alguns como quantidade de iterações). Já a outra versão possui, além dos itens mencionados, o **indicador de**

incrementos para a variável de controle. Esse indicador, que você conhecerá detalhadamente na seção seguinte, determina o valor de incremento a cada iteração da estrutura.

Comece agora a estudar a estrutura de repetição **para** na forma mais simples, sem o indicador de **incrementos**. Sua estrutura é mostrada na Figura 01 abaixo:

Figura 01 - Estrutura de repetição **para** SEM indicação de passo



No exemplo apresentado, é exibida a estrutura de repetição **para** que realizará 10 iterações (repetições). Essa estrutura, como dito anteriormente, possui uma variável de controle, relativa, na figura, à variável '**i**', responsável por armazenar a contagem das iterações. No exemplo, o valor inicial dessa variável é 1.

Já o valor 10, presente nesse exemplo, representa a quantidade de iterações e, conseqüentemente, a condição de parada (encerramento da estrutura de repetição). Em outras palavras, nesse caso, quando o valor de '**i**' for maior que 10, a estrutura de repetição **para** não executará novamente o bloco de códigos definido entre o **faça** e o **fim**, ou seja, os comandos presentes no bloco de código serão repetidos 10 vezes. Execute o código abaixo e veja o resultado de sua execução.

```
1 para i de 1 até 10 faça
2   escreva i
3 fim
4
```

O comando que está dentro dessa estrutura de repetição é o '**escreva i**'. Note que ele foi executado e responsável por imprimir os números de 1 a 10, pois, a cada execução do comando **para**, o valor da variável de controle '**i**' foi modificado. Essa mudança do valor da variável de controle ocorre uma vez a cada iteração e é chamada de **incremento**. No exemplo do **para** apresentado, o incremento da variável de controle, também conhecido como **passo**, é realizado sempre por meio da soma **i + 1**, armazenando o resultado na variável '**i**'.

#FicaDica

Altere o exemplo apresentado, modificando o valor inicial de 1 para 5 e, então, execute o programa novamente e analise o novo resultado. Você também pode modificar o valor 10 para outro valor desejado. Faça isso e observe o comportamento do código a partir do novo resultado gerado.

Percebeu a modificação que o comando produziu no resultado?



Na estrutura de repetição **para**, quando ocorre a primeira execução, o valor da variável de controle torna-se igual ao definido por meio do valor de inicialização. Este, por sua vez, diz respeito ao parâmetro indicado imediatamente após a palavra **de** do comando **para**. No exemplo apresentado, o valor inicial da variável de controle foi definido como 1 (um).

Logo em seguida, antes da primeira execução do laço, é verificado se o valor da variável de controle é menor ou igual ao valor da condição de parada – parâmetro definido como 10, indicado imediatamente após o **até** do comando **para**. Se a condição for verdadeira, inicia-se a execução do bloco de códigos do laço **para** e, no exemplo, o comando '**escreva i**' é executado. Após a execução de todos os comandos internos ao bloco de códigos do laço, o valor da variável de controle é incrementado em **+1**, e essa soma ocorrerá uma única vez a cada execução do laço até que o valor de parada seja ultrapassado.



Atenção

Note que na estrutura de repetição **para** a quantidade de vezes que o bloco de código será executado já é conhecida, diferentemente da estrutura de repetição **enquanto**, que não se conhece previamente a quantidade de vezes que o laço será executado.

Agora, explore a característica relacionada à quantidade de repetições já conhecida para uma estrutura de repetição **para**. Você sabe qual é a soma de números de 1 até 5? Essa é fácil, pois são apenas cinco números e o resultado é 10, "certo"? E se eu quiser somar do número 7 até o número 22? Você saberia me dizer qual é o resultado? Explore o exemplo abaixo, ele apresenta a soma dos números de 1 até 5. Em seguida, altere o programa para que ele faça a soma de 7 até 22.

```
1 var soma := 0
2 para i de 1 até 5 faça
3   soma := soma + i
4 fim
5 escreva "A soma é {soma}."
6
```

Você conseguiu obter o resultado da soma dos números de 7 até 22? Foi fácil, não? Imagine se você tivesse que realizar essa soma manualmente na calculadora, somando número a número. Quanto tempo você acha que precisaria para somar todos esses números? Para essas situações em que é necessário realizar repetidamente os mesmos passos, como somar uma sequência de números, utiliza-se as estruturas de repetição, seja a estrutura **enquanto**, seja a estrutura **para**.

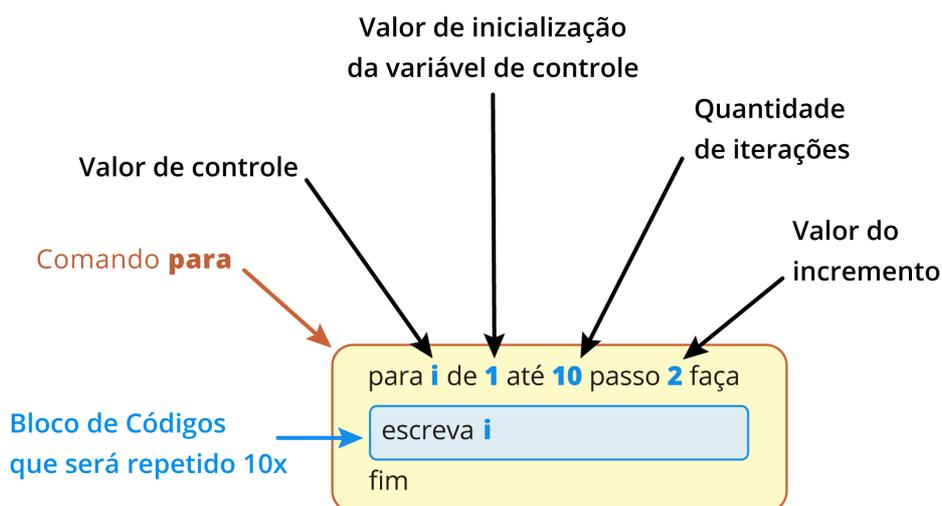
Estrutura de Repetição com Variável de Controle: COM indicação de passo de incremento

Agora, conheça a estrutura de repetição **para** com parâmetro de **passo**. Esse parâmetro permite definir qual o valor que será incrementado na variável de controle da estrutura de repetição a cada iteração. Nas estruturas com variável de controle, o **incremento** pode ser positivo ou negativo. Em outras palavras significa que o **passo** pode ser realizado positivamente, ou seja, somando-se um valor positivo à variável de controle a cada execução da estrutura de repetição, ou pode ser somado um determinado valor negativo, provocando, assim, o **decremento** (**É a modificação de um valor a partir da <u>diminuição gradual</u> em número, tamanho, quantidade ou intensidade.**) do valor armazenado na variável de controle. Veja, primeiro, o exemplo que realiza o **incremento** positivo de 2 a cada execução. Observe o código abaixo:

```
1 para i de 1 até 10 passo 2 faça
2   escreva i
3 fim
4
```

Repare que nesse exemplo há um novo parâmetro entre o valor **10** e a palavra **faça** na estrutura de repetição **para**. Esse parâmetro, denominado **passo**, define qual o valor que será somado ao valor de 'i' ao final de cada execução de laço do **para**, como mostra a Figura 02.

Figura 02 - Estrutura de repetição para SEM indicação de passo



Conforme o exemplo apresentado, o valor inicial da variável de controle 'i' é igual a 1. Ao final do primeiro laço, o valor do **incremento**, que no exemplo é 2, será somado ao valor presente na variável de controle, por esse motivo, no laço seguinte será apresentado o valor armazenado em 'i', que corresponderá a 3.

Na próxima repetição, será somado, novamente, o valor de passo 2 ao valor 3 presente na variável de controle, que assumirá um novo valor (**5**, resultado de 2+3). Esse **incremento** de dois em dois ocorrerá até que o conteúdo da variável de controle seja maior do que 10 e, assim, é encerrada a execução da estrutura de repetição **para**.

Agora, retorne ao bloco de código anterior e altere o valor da variável de controle 'i' de 1 para 2 e execute novamente o exemplo. Em seguida, observe quais são os valores que serão impressos por esse algoritmo.



E aí, conseguiu compreender as diferenças?

É importante que você modifique os exemplos para compreender cada estrutura de forma mais fácil.

Passo negativo no comando **para**

Além da possibilidade de se definir o tamanho dos **incrementos** realizados na variável de controle da estrutura de repetição **para**, é possível também definir que esse **incremento** ocorra na ordem inversa. Em outras palavras, significa que, ao invés do valor contido na variável de controle aumentar, é possível definir que ele diminua! Ou seja, vai de um valor maior até um valor menor.

Compreenda o exemplo abaixo que apresenta como fazer a execução da estrutura de repetição **para** na 'ordem inversa', ou seja, ao invés de ir de 1 até 10, a execução da estrutura de repetição será realizada de 10 até 1.

```
1 para i de 10 até 1 passo -1 faça
2   escreva i
3 fim
4
```

No exemplo acima, a variável de controle é iniciada com valor **10** e, a cada passo (execução do laço), esse valor será **decrementado** (diminuído) em **-1**. Isso ocorrerá até que o valor da variável de controle 'i' seja menor do que 1. Assim, serão impressos os números de 10 até 1, sequencialmente. Altere o valor de -1 para -2 e observe como se comportará o algoritmo. Note a partir de qual valor e quais os números serão impressos.

Agora que você compreendeu o uso do **para** com o uso de **decremento** unitário, lhe apresento um exemplo com **decrementos** de valor -2. O código abaixo realizará a impressão dos números pares de 10 até 1.

```
1 para i de 10 até 1 passo -2 faça
2   escreva i
3 fim
4
```

Que tal você alterar, agora, o exemplo acima e modificar o tamanho do **decremento**? Utilize novos valores (negativos). Você também pode alterar o valor inicial da variável de controle para que seja possível imprimir uma maior quantidade de números ao se utilizar passos muito grandes. Tente imprimir os números 100, 90, 80, 70, 60, 50, 40, 30, 20 e 10 (nessa ordem).

Personalizando a variável de controle

Você percebeu que em todos os exemplos a variável de controle utilizada foi a letra 'i'? Utilizar essa letra como variável de controle para as estruturas de repetição **para** é muito comum, porém, isso não é uma convenção. Dessa forma, caso você prefira, poderá definir outro nome para a variável de controle, de modo a tornar ainda mais fácil a leitura do algoritmo construído.

Abaixo, apresento-lhe um exemplo de código em que o nome da variável de controle é '**contador**'. A mudança para esse nome não interfere no funcionamento do programa apresentado, mas facilita a leitura do código.

```
1 para contador de 1 até 10 faça
2   escreva "Executando o laço de número {contador}"
3 fim
4
```



Vale lembrar que a definição do nome da variável de controle deve seguir as mesmas regras da definição de qualquer variável, conforme você aprendeu em aulas anteriores.

Portanto, fique atento à definição do nome dessa variável em seus algoritmos!

Aninhamento do comando para

A definição do nome da variável de controle em uma estrutura de repetição **para** torna-se importante, principalmente, quando é necessário realizar a construção de algoritmos com o **aninhamento** de duas ou mais dessas estruturas. Nesse caso, a estrutura de repetição **para interna** deve possuir o nome da variável de controle diferente do nome da variável de controle do **para externo**.

Em outras palavras, não se deve utilizar o mesmo nome da variável de controle em mais de uma estrutura **para**, caso seja necessário realizar o seu **aninhamento**. Se os nomes das variáveis de controle forem os mesmos, o comportamento de **incremento** da variável ocorrerá de forma inesperada.

No **aninhamento** da estrutura de repetição **para**, você terá a execução de um laço do **para externo** somente após a execução de todos os laços do **para interno**. Assim, somente após a execução de todos os laços da estrutura de repetição **para interna** é que é realizado um único laço na estrutura externa. No exemplo seguinte, é apresentada uma pequena tabuada para exemplificar como isso ocorre.

Observe com atenção o conteúdo que é impresso neste algoritmo e procure modificar os valores iniciais e finais de cada um dos comandos **para**, mas fique atento às interferências no resultado impresso.

Sugiro que você altere somente um dos elementos por vez, ou seja, experimente alterar, por exemplo, o valor 3 do **para interno**, definindo um novo valor de sua escolha. Verifique o que será impresso e só então, após realizar esse experimento, você irá alterar algum dos outros parâmetros (valor inicial da variável de controle, condição de parada ou valor de **passo**).



Não esqueça de sempre analisar quais são os impactos nos resultados gerados. Você consegue apontá-los?

```
1 # Exemplo com aninhamento de para's
2 para multiplicador de 1 até 4 faça
3   para multiplicando de 1 até 3 faça      # escreve a tabuada {1..4} x {1..3}
4     escreva "{multiplicador} * {multiplicando} = {multiplicador * multiplicando}"
5   fim
6 fim
7
```

Ficou claro como você deve utilizar esse recurso? Caso ainda tenha dúvidas sobre como aninhar as estruturas de repetição **para**, procure o seu professor mediador para que ele possa lhe auxiliar com esse exemplo.

Qual estrutura de repetição devo utilizar?

Após você conhecer a estrutura de repetição **para** e a estrutura **enquanto**, você deve estar se perguntando qual das duas estruturas é mais adequada para ser utilizada em seus algoritmos, certo? Essa é uma ótima pergunta!

Um programa pode ser escrito tanto com o uso do **para** como com o uso do **enquanto**. No entanto, há algumas situações que podem auxiliar na definição de qual estrutura você deve adotar.

A estrutura de repetição **enquanto** deve ser utilizada nas situações em que não se conhece a quantidade de repetições (laços) que serão executadas. Dessa forma, é melhor empregar uma estrutura sem a variável de controle, pois ela utiliza uma expressão lógica que será testada antes da execução de cada laço da estrutura **enquanto**.

Já a estrutura de repetição **para** deverá ser utilizada quando conhecermos o número de vezes em que o bloco de código será executado, ou seja, caso já saibamos a quantidade de repetições que será realizada do bloco de códigos.

Acredito que com essas informações fica fácil definir qual das duas estruturas de repetição é mais adequada durante a construção de seus algoritmos. De toda forma, procure exercitar cada uma delas e faça os exercícios de estruturas de repetição utilizando ambas. Assim, por experiência própria, você perceberá que a solução com uma delas é mais adequada que com a outra. Com essa prática você distinguirá melhor e com mais facilidade em quais situações você deve utilizar o **enquanto** ou **para**.



Atividade 01

Repetição **para**

Utilizando a estrutura de repetição **para**, construa um algoritmo que receberá um número n , ao final, calcula e depois imprime a soma de todos os números ímpares de 1 até número lido.



Atividade 02

Repetição **enquanto**

Utilizando a estrutura de repetição **enquanto**, construa um algoritmo que receberá um número n , ao final, calcula e depois imprime a soma de todos os números pares de 1 até número lido.



Atividade 03

Repetição **para**

Utilizando a estrutura de repetição **para**, construa um algoritmo que calcula e, ao final, imprime a soma de todos os números primos de 1 até o número lido.

Exercício 4 - Repetição enquanto

Utilizando a estrutura de repetição **enquanto**, construa um algoritmo que calcula e, ao final, apresenta a soma de todos os números primos de 1 até 100.

Conclusão

Chegamos ao fim de mais uma aula sobre estruturas de repetição. A partir de agora, você conhece as duas estruturas de repetição utilizadas na maioria das linguagens de programação: as estruturas **com variável de controle** e as **sem variável de controle**. Tais estruturas são fundamentais para a construção da maioria dos algoritmos que compõem os programas de computadores. Portanto, o seu entendimento desse conteúdo é primordial.

Caso você tenha alguma dúvida acerca de algum dos assuntos referentes às estruturas de repetição, procure o seu professor mediador, poste sua dúvida nos fóruns ou nos encontros online. Você também pode discutir sobre as estruturas de repetição com seus colegas de turma.



Resumo

Nesta aula, você conheceu a estrutura de repetição com variável de controle, pré-testada, denominada de estrutura de repetição **para**. Também viu que é possível definir a forma como é realizado **incremento** da variável de controle, podendo ele ser positivo ou negativo. Além disso, viu que é possível definir o tamanho do passo de **incremento** (de 1 em 1; de 2 em 2, etc.). Em seguida, conheceu a possibilidade de definir o nome da variável de controle, além do processo de **aninhamento** da estrutura **para**. Também conheceu quando é apropriado utilizar o **para** ou o **enquanto** na resolução de problemas que necessitem de estruturas de repetição.



Referências

Linguagem Potigol: Programação para todos. Disponível em: <<http://potigol.github.io/>>. Acesso em: 31 mar. 2018.