

Programação de Computadores





Material teórico



Responsável pelo Conteúdo:

Prof Ms Amilton Martha

Revisão Textual:

Profa Esp Vera Lídia de Sá Cicaroni

UNIDADE Comandos de Repetição



Vamos nos aprofundar nos algoritmos em Scratch? Dessa vez vamos aprender a usar os comandos de repetição. Com mais esse item em nosso grupo de comandos, vamos poder realizar algoritmos mais avançados.

O estudo de algoritmos constitui uma disciplina acumulativa, o que quer dizer que quanto mais comandos conhecermos, maior capacidade teremos para resolver algoritmos mais elaborados.





Atenção

Para um bom aproveitamento do curso, leia o material teórico atentamente antes de realizar as atividades. É importante também respeitar os prazos estabelecidos no cronograma.

Contextualização

Percebemos, na unidade anterior, que os algoritmos precisam de desvios condicionais. Em determinada parte do algoritmo, ele precisa tomar uma decisão para executar, ou não, um bloco de comandos.

Nesta unidade, nós vamos entender que além de executar um bloco de comandos em uma determinada condição, será preciso, em alguns casos, que esse bloco execute diversas vezes a mesma ação enquanto uma condição for verdadeira. Esses são os comandos de repetição, também chamados de laços de repetição ou mesmo loopings.

Veremos que isso será muito útil em diversos algoritmos e sem esses comandos ficaria impossível encontrar muitas soluções.

Material Teórico



O laço de repetição em pseudocódigo

Existem ocasiões em que é necessário efetuar a repetição de um trecho de código de programa um determinado número de vezes. Nesse caso, devemos criar um looping que efetue o processamento de um trecho de código inúmeras vezes. Os loopings são chamados laços de repetição.

Como exemplo, podemos pensar num programa que deva ler um número e imprimir na tela a informação, indicando se ele é par ou impar, porém, neste caso, queremos essa mesma informação sobre 5 números. Se não existissem os comandos de repetição, teríamos que repetir o código de verificação para saber se o número é par ou ímpar 5 vezes (CTRL+C e CTRL+V).

Portanto, sempre que quisermos que um bloco de instruções seja repetido inúmeras vezes, devemos usar comandos de repetição e jamais usar o recurso de copiar e colar.

Inicialmente vamos apresentar os laços de repetição em pseudocódigo e, em seguida, mostraremos os equivalentes no nosso ambiente Scratch.

1) Repetição com teste no início

Neste tipo de repetição, o teste lógico (condição) é feito no início do looping. Se a condição for falsa, o programa não entrará no looping e será jogado para a próxima linha após o seu fim. Enquanto, se a condição for verdadeira, os comandos serão repetidos. Veja a sintaxe em pseudocódigo:

Exemplo:

```
Algoritmo: Calcula se 5 números são pares
Início
      inteiro numero, resto, contador;
      contador = 1;
      enquanto(contador <= 5)
             escreva("Entre com um número");
             leia(numero);
             resto = numero MOD 2;
             se(resto=0)
                   escreva("O número é PAR");
             senão
                   escreva("O número é IMPAR");
            fim se
             contador = contador + 1;
      fim enquanto
Fim
```

2) Repetição com teste no final

Agora, o teste para executar o bloco é feito ao final, ou seja, mesmo a condição sendo falsa, pelo menos uma vez, o bloco será executado, ao contrário do caso anterior, em que o bloco só será executado se a condição for verdadeira no início. Veja em Portugol:

```
faça
<comandos>;
enquanto(<condição>);
```

Exemplo:

```
Algoritmo: Calcula se 5 números são pares
Início
      inteiro numero, resto, contador;
      contador = 1;
      faça
             escreva("Entre com um número");
             leia(numero);
             resto = numero MOD 2;
             se(resto=0)
                   escreva("O número é PAR");
             senão
                   escreva("O número é IMPAR");
            fim se
             contador = contador + 1;
      enquanto(contador \leq 5);
Fim
```

3) Repetição com variável de controle

Neste looping existe uma variável que controla a quantidade de vezes que o trecho será repetido. Esse tipo de controle é mais usado quando sabemos quantas vezes deve ser realizada a repetição, embora seja possível, nesse caso, utilizar qualquer um dos três tipos de repetição. Veja no pseudocódigo:

```
para(inicia variáveis; condição; incremento)
<comandos>;
fim para
```

Exemplo:

```
Algoritmo: Calcula se 5 números são pares
Início

inteiro numero, resto, contador;

para(contador=1; contador<=5; contador = contador+1)

escreva("Entre com um número");

leia(numero);

resto = numero MOD 2;

se(resto=0)

escreva("O número é PAR");

senão

escreva("O número é IMPAR");

fim se

fim para

Fim
```

O laço de repetição em Scratch

O Scratch possui quatro comandos de repetição diferentes que podem ser divididos em laços contados e laços condicionais. Os laços contados executam o bloco um número de vezes conhecido pelo usuário, já os laços condicionais dependem de uma condição para o bloco ser repetido.

1) O comando "sempre"

O primeiro desses comandos é o "sempre". O objetivo desse bloco é repetir um conjunto de comandos indefinidamente, por isso é conhecido como looping infinito.

Um detalhe importante é que não podemos inserir mais comandos após o comando "sempre". Note, pelo desenho da figura 1, que não é permitido encaixe de novos comandos abaixo dele.

Por exemplo, queremos que exista uma música de fundo durante toda a execução do seu programa. Quando acabar a música, ela deve recomeçar. Veja, na Figura 1, como ficaria esse programa.

Figura 1 – Uso do comando "sempre" em Scratch



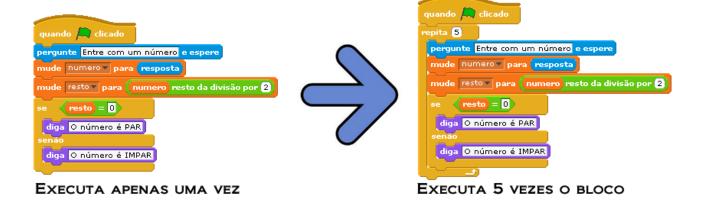
2) O comando "repita<n>"

Esse bloco é semelhante ao comando de repetição com variável de controle. O "repita" faz com que o bloco seja realizado <n> vezes. O <n> pode ser um valor fixou ou mesmo uma variável.

O comando "repita<n>" permite que outros comandos sejam inseridos após ele, conforme pode ser visualizado no desenho do comando abaixo.

Por exemplo, usando o algoritmo visto anteriormente para calcular se 5 números são pares ou ímpares, a Figura 2 mostra o uso do comando "repita<n>" para esse problema.

Figura 2 – Uso do comando repita < n > no Scratch



3) O comando "sempre se <condição>"

Esse comando executa o bloco enquanto a condição for verdadeira (true). Ele é equivalente ao comando de repetição com teste no início já apresentado em pseudocódigo e pode ser usado para repetir o bloco com uma condição que não dependa de um contador.

Por exemplo, podemos calcular o triplo de um número e podemos fazer o usuário escolher se quer começar novamente o bloco. Veja esse exemplo na Figura 3.

Figura 3 – Comando "sempre se < condição > " em Scratch

```
quando clicado

quando clicado

mude continuary para 3

sempre se continuar = 8

pergunte Entre com um número e espere

mude numero para resposta

diga junte 0 triplo vale numero * 3

EXECUTA APENAS UMA VEZ

EXECUTA SEMPRE QUE O USUÁRIO

RESPONDER "S"
```

Note, na Figura 3, lado direito, que é o usuário que controla se o bloco será repetido ou não. Como a condição está no começo do bloco, a variável "continuar" já é iniciada com "S"; caso contrário, o bloco não seria executado nenhuma vez.

Assim como o comando "sempre", esse comando também não permite que outros comandos sejam inseridos após ele. Note, novamente, pelo desenho do comando, que ele não permite encaixar mais comandos após ele.

4) O comando "repita até < condição > "

Diferente do comando "sempre se <condição>", que executa enquanto a condição for verdadeira, o comando "repita até <condição>" executa enquanto a condição for falsa. O looping só encerra quando a condição for verdadeira. Esse comando também permite que outros comandos sejam inseridos após ele.

Por exemplo, queremos imprimir a tabuada de um número digitado pelo usuário. Veja, na Figura 4, como ficaria o programa com Scratch.

Figura 4 – Uso do comando "repita até < condição > " em Scratch

```
quando clicado
mude tabuada para 1

pergunte digite um número para saber a sua tabuada e espere
mude num para resposta

repita até tabuada > 10 ou tabuada = 10

diga junte num junte igunte tabuada junte inum tabuada por 2 segundos
mude tabuada por 1
```

Na Figura 4, além da apresentação do comando de repetição "repita até <condição>", apresentamos o comando "mude <variável> por <valor>". Diferente do comando "mude <variável> para <valor>", que atribui o valor para a variável selecionada, esse comando adiciona (põe) o valor na variável, ou seja, o valor original da variável será acrescido do valor escolhido. No item "mude <tabuada> por 1", adiciona 1 na variável tabuada cada vez que passa por esse comando.

Agora que já aprendemos os comandos de repetição, podemos adicioná-los à nossa lista de comandos disponíveis para nossos algoritmos. Algoritmos mais elaborados necessitam da mistura de vários comandos, como mistura de comandos de repetição com comandos de decisão.

Por exemplo, queremos fazer um programa que sorteie um número entre 1 e 10, em que o usuário digite os "chutes" e, para cada tentativa, o programa forneça a dica se o "chute" foi alto ou baixo. Nesse caso, o usuário vai digitar um número para tentar acertar e isso vai se **repetir até** o usuário acertar.

Podemos notar que, no enunciado do problema, há uma indicação de qual comando usar. Veja o programa pronto na Figura 5.

Figura 5 – Programa para adivinhar um número

```
quando clicado

mude num para sorteie número entre 1 e 10

repita até num = chute

se chute > num

diga Menor por 2 segundos

senão

se chute < num

diga Maior por 2 segundos

pergunte qual é o chute? e espere

mude chute para resposta

diga Acertou!!!! por 2 segundos
```

Vejamos outro exemplo do uso da combinação de comandos de repetição e comandos de decisão. Um famoso algoritmo encontrado em livros propõe cadastrar o peso de 90 bois de um frigorífico e calcular qual o peso do boi mais gordo. Nesse caso, precisamos criar uma variável que armazene o valor do peso maior. A cada nova entrada de peso, o comando verifica se o novo dado é maior do que o armazenado.

Veja na Figura 6 como poderia ser implementado, em Scratch, o algoritmo do boi mais gordo.

Figura 06 – Programa para calcular o maior valor

```
quando clicado
mude contador para 1
mude gordo para 0
repita 10

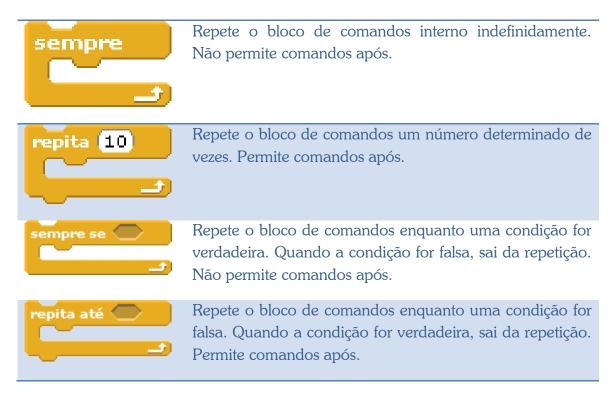
pergunte junte Entre com o peso do boi contador e espere
mude peso para resposta

se peso gordo
mude gordo para peso
mude contador por 1

diga junte O boi mais gordo pesa gordo
```

Veja os novos comandos que aprendemos nesta unidade:

➤ Aba Controle:



Aba Variáveis:



➤ Aba Operadores:



Material Complementar

1) Site da WikiPédia:

http://pt.wikipedia.org/wiki/Algoritmo

http://pt.wikipedia.org/wiki/Programa de computador

- 2) Livros Disponíveis na Biblioteca Virtual Universitária da Pearson Educationz
- FORBELLONE, A. L. V.; EBERSPACHER, H. F. **Lógica de Programação:** a construção de algoritmos e estrutura de dados. 3. ed. São Paulo: Makron Books do Brasil, 2005.
- MANZANO, J. A. N. G. Algoritmos: lógica para desenvolvimento de programação. 20. ed. ,São Paulo: Erica, 2007.
- VILARIM, G. O. **Algoritmos:** programação para iniciantes. Rio de Janeiro: Ciência Moderna, 2004.
- 3) Site oficial do Scratch:

http://scratch.mit.edu/

4) Apostila em português do Scratch:

http://oficinas.pensamentodigital.org.br/apostila iniciacao programacao.pdf

5) Manual de Referência do Scratch

http://info.scratch.mit.edu/Support/Reference Guide 1.4

Referências

Scratch: http://scratch.mit.edu/

DEITEL, H. M. JAVA. Como Programar. 8. ed. Sao Paulo:Pearson Prentice Hall, 2010.

Anotações





www.cruzeirodosulvirtual.com.br Campus Liberdade Rua Galvão Bueno, 868 CEP 01506-000 São Paulo SP Brasil Tel: (55 11) 3385-3000









