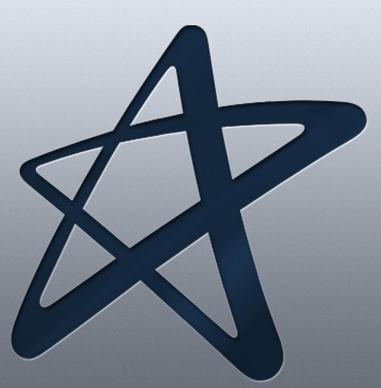


Programação de Computadores





Material teórico



Responsável pelo Conteúdo:

Profa. Ms. Amilton Martha

Revisão Textual:

Profa. Esp Vera Lídia de Sa Cicaroni

UNIDADE

Variáveis, Operações Numéricas e Desvios Condicionais



Nesta unidade, além de estudar o conceito de variáveis e conhecer seus tipos, vamos aprender como criar identificadores e trabalhar com alguns exemplos no Scratch. Vamos também trabalhar com operações numéricas e conhecer operadores relacionais e lógicos para serem usados no desvio condicional.

A partir de agora, vamos nos aprofundar mais no ambiente Scratch, a fim de treinar nossos algoritmos com muitas variáveis e comandos de decisão.





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

Um software ou programa é composto por um conjunto de comandos que, dispostos em certa ordem, produzem o efeito desejado. O algoritmo é uma sequência de passos lógicos para resolução de um problema específico. Esses passos são descritos por comandos.



O Scratch consegue capturar essa lógica por meio dos comandos, que são representados por peças de um quebra-cabeça. Cada peça representa um comando do algoritmo. Nosso propósito, aqui, é conhecer cada peça desse quebra-cabeça e saber onde encaixá-la.

Na unidade anterior, conhecemos as peças para entrada e saída de dados (pergunte e diga, respectivamente). Nesta unidade, vamos aprender algumas peças fundamentais para a construção de algoritmos: as variáveis e os desvios condicionais.

E claro que este é apenas o começo, pois, conhecendo mais peças desse quebracabeça, podemos criar algoritmos e programas cada vez mais avançados (e emocionantes).

Material Teórico



Variáveis

Variável é uma área de memória do computador na qual um valor pode ser armazenado para a utilização posterior em seu programa. Uma variável é identificada dentro do seu programa por um nome. Por exemplo, para armazenar o seu nome, a variável pode se chamar "nome"; para armazenar a idade, pode ser "idade" e assim por diante.

Escolher nomes de variáveis que sejam representativos do seu conteúdo faz com que os programas sejam auto documentados.

O nome de uma variável pode ser qualquer identificador válido. As regras para a criação de um identificador, na maioria das linguagens de programação, são:

- 1) não pode começar com números, apenas com letras ou underline;
- 2) não pode conter espaços em branco;
- 3) não pode conter caracteres especiais;
- 4) não pode ser um comando da linguagem.

Veja alguns exemplos de nomes de variáveis inválidos:

- 1tentativa: não pode começar com números;
- meu nome: não pode conter espaços;
- ficha#completa: não pode conter caracteres especiais;
- float: é um comando de uma linguagem.

As regras para a criação de variáveis podem ter diferenças de uma linguagem para outra, mas é importante ter em mente essas 4 regras, pois, usando-as, é possível criar variáveis em qualquer linguagem. Apesar de nosso ambiente Scratch permitir variáveis fora das regras, nós recomendamos usá-las sempre.

Na maioria das linguagens de programação, uma variável está sempre associada a um tipo, ou seja, ela é capaz de armazenar uma categoria de dados. De modo geral, temos 4 tipos de dados:

- 1) **Inteiro**: armazena dados numéricos que não possuem casa decimal. Usamos esse tipo de variável para armazenar idade, número de filhos, quantidade de carros, etc.
- 2) **Real**: armazena dados numéricos que precisam de casas decimais. Exemplos de variáveis reais são peso, altura, salário, etc.
- 3) **String**: Armazena dados caracteres, ou seja, informações alfanuméricas, como nome, endereço, sexo, etc.
- 4) **Lógico**: esse tipo de variável armazena apenas duas possíveis informações: verdadeiro (true) ou Falso (false).

Algumas linguagens de programação não possuem tipo, ou seja, as variáveis adaptamse ao conteúdo que lhes é atribuído. Por exemplo, se, em uma variável x, for colocado o valor 2, ela passa a ser do tipo inteira; se for atribuído o valor 44.99, passa a ser Real; e se for atribuído o valor "José", ela vira String. Dentre elas podemos citar Javascript, PHP, ASP e Scratch.

No Scratch, nós criamos variáveis na aba "Variáveis". Para a criação, clicamos em "criar uma Variável". Uma janela se abrirá e nela devemos colocar o nome da variável, seguindo as regras já apresentadas anteriormente, e informar se essa variável será válida para todos os objetos ou apenas para o objeto em questão. Cada objeto pode conter quantas variáveis o programador quiser e essas variáveis podem ser compartilhadas com outros objetos, por isso a existência dessa opção na criação de uma variável.

Operações Numéricas

Uma tarefa que o computador sabe fazer muito bem é realizar contas. Em computação, temos cinco operações básicas, sendo que as quatro primeiras são muito bem conhecidas por nós: soma, subtração, multiplicação e divisão. A última operação é conhecida como "módulo" e calcula o resto da divisão de dois números inteiros.

Veja, na Tabela 1, a seguir, as operações numéricas e sua equivalência no Scratch.

Operador Operação Scratch Exemplo 15+2=17 Soma 15-2=13 Subtração 15*2=30 Multiplicação Divisão 15/2 = 7.5Resto de uma 15mod2=1 mod 🦲 resto da divisão por divisão inteira

Tabela 1 - Operações Numéricas

Para exemplificar a criação de variáveis e o uso de operações numéricas no Scratch, vamos implementar o algoritmo da Figura 1 em Scratch.

Figura 1 - Algoritmo para a soma de dois números

```
início
inteiro num1;
inteiro num2;
inteiro resultado;
escreva "Digite um número"
leia num1;
escreva "Digite outro número";
leia num2;
resultado = num1+num2;
escreva "O valor da soma é" + resultado
fim
```

Vamos aos passos para a criação do algoritmo da Figura 1 no Scratch;

Passos	Descrição	Imagem
1	Criar um novo projeto no Scratch	Acquire Editor Compatible A) and Compatible (1) and
2	Criar as variáveis que serão usadas no programa. Nesse caso serão 3, "num1", "num2" e "resultado". Para isso, vá à opção "Variáveis".	criar uma Variável Apagar uma Variável Apagar uma Variável num1 para o num2 resultado Nome da variável? num2 resultado Nome da variável? num2 para o num2 para todos os objetos para este objeto apenas o cancelar Nome da variável? num2 para todos os objetos para este objeto apenas o cancelar Nome da variável? num2 para todos os objetos para este objeto apenas o cancelar o cancelar
3	Na opção "Controle", selecione "Quando bandeira clicado" e arraste para a área de programação.	quando 🔼 clicado

4	Na opção "Sensores", vamos solicitar os números ao usuário. Use o comando "pergunte <qual nome?="" o=""> e espere" alterando a mensagem para <digite número="" um="">.</digite></qual>	quando Clicado pergunte Digite um número e espere
5	Sempre que usamos o comando "pergunte", o que o usuário digitar fica armazenado em "resposta". Para transferir a resposta do usuário para a variável, vamos à opção "Variáveis" e usamos "mude <num1> para".</num1>	quando clicado pergunte Digite um número e espere mude num1 para 0
6	Agora, na opção "Sensores", arrastamos a resposta para a variável "num1".	quando clicado pergunte Digite um número e espere mude num1 para resposta
7	Vamos repetir os passos 4, 5 e 6 para perguntar outro número ao usuário.	quando clicado pergunte Digite um número e espere mude num1 v para resposta pergunte Digite outro número e espere mude num2 v para resposta
8	Na opção "Variáveis", agora, vamos calcular o resultado. Arraste o comando "mude <resultado> para <0>".</resultado>	quando clicado pergunte Digite um número e espere mude num1 v para resposta pergunte Digite outro número e espere mude num2 v para resposta mude resultado v para 0
9	Na opção "Operadores", vamos arrastar a opção de soma para o "resultado".	quando clicado pergunte Digite um número e espere mude num1 v para resposta pergunte Digite outro número e espere mude num2 v para resposta mude resultado v para +

```
10
        Na opção "Variáveis", agora, vamos
                                                          quando 🔼 clicado
         arrastar as variáveis num1 e num2.
                                                         pergunte Digite um número e espere
                                                          mude num1▼ para (resposta)
                                                         pergunte Digite outro número e espere
                                                         mude num2▼ para resposta
                                                          mude resultado para num 1 + num 2
11
        Agora vamos imprimir a resposta. Na
                                                          quando 牔 clicado
         opção "Aparência", arrastamos a opção
                                                         pergunte Digite um número e espere
         "Diga < Olá!>".
                                                         mude num1▼ para (resposta)
                                                         pergunte Digite outro número e espere
                                                         mude num2▼ para resposta
                                                         mude resultado para num 1 + num 2
                                                         diga Olá!
12
        Agora vamos juntar (concatenar) as
         frases "O valor da soma é" com o
                                                         pergunte Digite um número e espere
        resultado. Na opção "Operadores",
                                                         mude num1 para resposta
        arraste o "junte <olá> e <mundo>".
                                                         pergunte Digite outro número e espere
         Altera a frase para "O valor da soma é"
                                                         mude num2▼ para (resposta)
         e, na opção "variáveis", arraste a
                                                         mude resultado para num 1 + num 2
         variável "resultado".
                                                         diga junte O valor da soma é
```

Além das operações básicas, na opção "Operadores" temos outras operações matemáticas disponíveis no Scratch, que podem ser vistas na Tabela 2, a seguir:

Tabela 2 - Outras funções matemáticas do Scratch

Funções	Descrição
abs	Retorna o valor absoluto, não negativo de um número.
sqrt	Retorna a raiz quadrada de um número.
sin	Retorna um valor representando o seno de um ângulo.
cos	Retorna um valor representando o coseno de um ângulo.
tan	Retorna um valor representando a tangente de um ângulo.
asin	Retorna o arco-seno de um valor numérico.
acos	Retorna o arco-cosseno de um valor numérico.
atan	Retorna o arco-tangente de um valor numérico.
ln	Retorna o inverso do expoente de um valor.
log	Retorna o log natural de um número.
e^	Retorna o expoente natural de um valor específico.
10 ^	Retorna o valor de um número elevado à potência 10.

Desvios Condicionais

Até agora conseguimos fazer entradas nos programas, processar os valores e gerar saídas. Porém, nosso programa possui limites, pois, às vezes, é necessário tomar decisões de acordo com a resposta de um cálculo. Por exemplo, imagine um programa que calcule a média de um aluno e imprima o resultado. Deverá imprimir APROVADO, caso a média seja maior que 6, ou REPROVADO, caso não seja. Nota-se, claramente, a necessidade de uma decisão, ou seja, um **SE**.

Nesse ponto, o programa chega a um desvio, em que decide qual o caminho a seguir, baseado em uma condição (ou várias).

1) Desvio Condicional Simples

Para solucionar o problema das condições, vamos incorporar uma nova instrução no nosso pseudocódigo.

```
se (<condição>)
<lista de comandos do bloco verdadeiro>
fim se
```

Se a < condição > for verdadeira, o programa passará a executar a lista de comandos do bloco verdadeiro; caso contrário, não entrará nesse bloco.

Exemplo:

```
início

string nome;

real nota1, nota2, media;

escreva "Entre com o seu nome:";

leia nome;

escreva "Entre com a Nota 1";

leia nota1;

escreva "Entre com a Nota 2";

leia nota2;

media = (nota1 + nota2) / 2;

se (media>=6)

escreva nome + " você foi Aprovado";

fim se

fim
```

Veja como esse algoritmo ficaria em Scratch, na Figura 2.

Figura 2 - Desvio Condicional Simples em Scratch

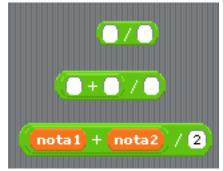
```
criar uma Variável
Apagar uma Variável
                                   pergunte Entre com o seu nome e espere
✓ media
                                    mude nome ▼ para resposta
▼ nome
                                   pergunte Entre com a Nota 1 e espere
 nota 1
                                    mude nota1 ▼ para resposta
▼ nota2
                                    pergunte Entre com a Nota 2 e espere
mude media▼ para 0
                                    mude nota2 ▼ para resposta
mude media por 1
                                    mude media v para (nota1 + nota2) / 2
mostra variável media 🔻
                                           media > 6 ou (media = 6
esconde variável media 🔻
                                      diga junte nome você foi Aprovado
Cria uma lista
```

A maioria dos comandos mostrados na Figura 2 já é conhecida por nós. Declarar as variáveis para nome, nota1, nota2 e média além de solicitar as informações de nome e notas para o usuário é tarefa repetida. As novidades aparecem a partir do cálculo da média.

Note que, para calcular a média, temos 2 operações matemáticas, uma soma e uma divisão. Note que as operações são montadas sobrepostas uma sobre a outra. Você pode montar a sua fórmula, colocando as operações em ordem inversa de operação.

No nosso exemplo, primeiro devemos fazer a soma e depois a divisão, portanto, no Scratch, montamos primeiro a divisão e depois a soma. Veja, na Figura 3, a sequência da montagem.

Figura 3 - Montagem de operações matemáticas



Outra novidade que apresentamos na Figura 2 são os operadores relacionais, que são usados para fazer comparações nas expressões lógicas. Na Tabela 3, temos os operadores relacionais e sua equivalência no Scratch.

Tabela 3 - Operadores Relacionais

	Algoritmo	Scratch
Menor	<	valor1 < valor2
Maior	>	valor1 > valor2
Igual	=	(valor1 = valor2)
Diferente	<>	não valor1 = valor2
Menor ou igual	<=	valor1 < valor2 ou valor1 = valor2
Maior ou igual	>=	valor1 > valor2 ou valor1 = valor2

Além dos operadores relacionais, também temos os operadores lógicos, que permitem combinar operadores relacionais. Na tabela 4, temos os operadores lógicos e sua equivalência no Scratch.

Tabela 4 - Operadores Lógicos no Scratch

Algoritmo	Scratch		
Е	valor1 > valor2 e valor2 > valor3		
OU	valor1 > valor2 ou valor2 > valor3		
NÃO	não valor1 = valor2		

Os operadores lógicos são usados em expressões lógicas e o resultado sempre será true (Verdadeiro) ou false (Falso). Na tabela 5, vemos a tabela verdade, quando agrupamos os operadores.

Tabela 5 - Tabela Verdade de operadores lógicos

P	Q	NÃO P	P e Q	P ou Q
V	V	F	V	V
V	F	F	F	V
F	V	V	F	V
F	F	V	F	F

O comando SE do Scratch possui sintaxe bem semelhante ao nosso algoritmo em pseudocódigo. A lista de instruções do bloco verdadeiro é colocado (encaixado) no meio do comando.

2) Desvio Condicional Composto

Usamos o desvio condicional composto quando uma sequência de ações deve ser executada caso a condição seja falsa.

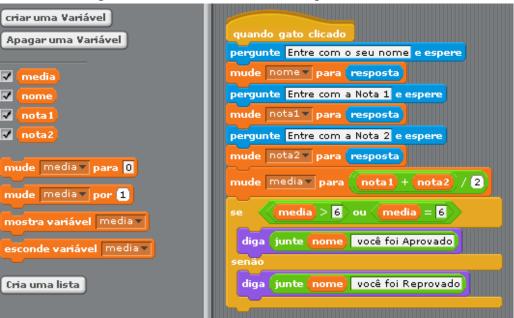
Se a <condição> for verdadeira, o programa passará a executar a lista de comandos do bloco verdadeiro; caso contrário, do bloco falso; e, nunca, de ambos os blocos.

Exemplo:

```
início
      string nome;
      real nota1, nota2, media;
      escreva "Entre com o seu nome:";
      leia nome:
      escreva "Entre com a Nota 1";
      leia nota1;
      escreva "Entre com a Nota 2";
      leia nota2:
      media = (nota1 + nota2) / 2;
      se (media>=6)
             escreva nome + "você foi Aprovado";
      senão
             escreva nome + "você foi Reprovado"
      fim se
fim
```

Podemos notar que, no algoritmo anterior, a única diferença inserida é que existe uma mensagem para o bloco verdadeiro e outra para o bloco falso. Veja, na Figura 4, como fica o algoritmo no Scratch.

Figura 4 - Desvio Condicional Composto em Scratch



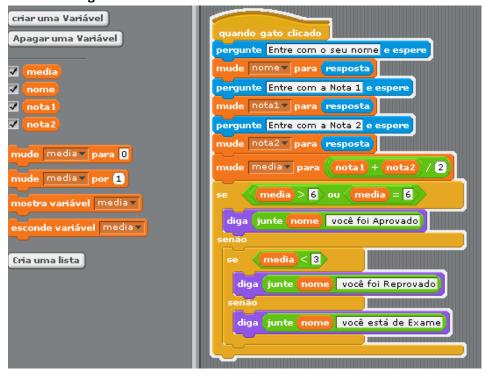
3) Desvios Condicionais Encadeados

Neste caso, necessitamos de condições encadeadas, nas quais há verificação de condições sucessivas, em que uma ação será executada se um conjunto de condições for aceito. Em pseudocódigo, teremos:

Exemplo:

```
início
      string nome;
      real nota1, nota2, media;
      escreva "Entre com o seu nome:";
      leia nome:
      escreva "Entre com a Nota 1";
      leia nota1:
      escreva "Entre com a Nota 2";
      leia nota2;
      media = (nota1 + nota2) / 2;
      se (media>=6)
             escreva nome + "você foi Aprovado";
      senão
             se (media<3)
                   escreva nome + "você foi Reprovado"
             senão
                    escreva nome + "você está de Exame"
             fim se
      fim se
fim
```

Figura 5 - Desvio Condicional Aninhado em Scratch



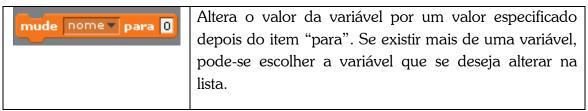
Na Figura 5, podemos notar que o uso de desvio condicional aninhado (ou encadeado) significa o uso de comandos SE-SENÃO dentro de comandos SE-SENÃO. Com o uso de apenas um comando SE-SENÃO, teremos 2 saídas possíveis; com o uso de 2 comandos SE-SENÃO, teremos 3 saídas possíveis. Desse modo, com 3 comandos SE-SENÃO, teremos 4 saídas possíveis e assim por diante.

Veja os novos comandos que aprendemos nesta unidade:

★ Aba Controle:

se	Desvio condicional simples, que executa os comandos dentro do bloco "se", se a condição for verdadeira.		
se senão	Desvio condicional composto, que executa os comandos dentro do bloco "se", se a condição for verdadeira, e executa os comandos do bloco "senão", se a condição for falsa.		

Aba Variáveis:



★ Aba Operadores:

	Operações aritméticas fundamentais entre dois valores: soma, subtração, multiplicação e divisão.
	Operadores Relacionais para a construção de condições. Esses operadores sempre retornam um valor lógico (true) para verdadeiro ou (false) para falso.
e ou não	Operadores Lógicos para a construção de condições mais complexas.
nesto da divisão por 📄	Operação aritmética MOD, que retorna o resto da divisão de 2 números inteiros.
raiz quadrada▼ de 10)	Calcula o resultado de uma função selecionada aplicada a um número específico.

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 Education
- ▶ 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

 $Scratch: \underline{http://scratch.mit.edu/}$

DEITEL, H. M. **JAVA: Como Programar**. 8. ed. São 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









