



x: 0

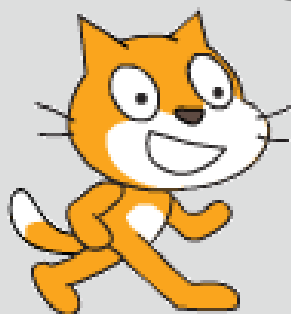
y: 0

quando clicar em 

mude para o pano de fundo **PRODUTO EDUCACIONAL**

diga **SCRATCH**

**SCRATCH**



**CONTRIBUIÇÕES AO ENSINO DE GEOMETRIA POR MEIO  
DA CRIAÇÃO DE OBJETOS DE APRENDIZAGEM**

**Airan Priscila de Farias Curci**

**Marcelo Souza Motta**

LONDRINA

2017





## **PRODUTO EDUCACIONAL**

# **O *SOFTWARE* DE PROGRAMAÇÃO SCRATCH NA FORMAÇÃO INICIAL DO PROFESSOR DE MATEMÁTICA POR MEIO DA CRIAÇÃO DE OBJETOS DE APRENDIZAGEM**

**Mestrado Profissional em Ensino de Matemática**

**UTFPR – Câmpus Londrina**

LONDRINA

2017

**AIRAN PRISCILA DE FARIAS CURCI**

**O *SOFTWARE* DE PROGRAMAÇÃO SCRATCH NA FORMAÇÃO  
INICIAL DO PROFESSOR DE MATEMÁTICA POR MEIO DA  
CRIAÇÃO DE OBJETOS DE APRENDIZAGEM**

Produto Educacional apresentado ao Programa de Pós-Graduação em Ensino de Matemática da Universidade Tecnológica Federal do Paraná – UTFPR como parte dos requisitos para a obtenção do título de “Mestre em Ensino de Matemática”.

Orientador: Prof. Dr. Marcelo Souza Motta

LONDRINA

2017

## TERMO DE LICENCIAMENTO

Esta Dissertação e o seu respectivo Produto Educacional estão licenciados sob uma Licença Creative Commons *atribuição uso não-comercial/compartilhamento sob a mesma licença 4.0 Brasil*. Para ver uma cópia desta licença, visite o endereço <http://creativecommons.org/licenses/by-nc-sa/4.0/> ou envie uma carta para Creative Commons, 171 Second Street, Suite 300, San Francisco, Califórnia 94105, USA.



# Apresentação

Caros colegas,

Este material é fruto de uma pesquisa de mestrado em Ensino de Matemática que aconteceu no período de 07 de junho à 30 de agosto de 2016 em uma Universidade Estadual do Estado do Paraná, com os alunos do quarto ano do curso de Licenciatura em Matemática. Este estudo trouxe o *software* de programação Scratch como uma proposta para o desenvolvimento de objetos de aprendizagem no formato de jogos digitais para o ensino de Geometria.

A interface gráfica, os recursos de mídia e a manipulação intuitiva proporcionada pelo *software*, fazem dele, um potencial recurso educacional no ensino de Matemática. O Scratch possibilita a criação de objetos simples, por meio de uma linguagem de programação intuitiva, favorecendo a interatividade em um ambiente dinâmico de ensino e aprendizagem.

Como sugestões para o ensino de Geometria, por meio da utilização de recursos tecnológicos, este material traz as principais ferramentas do Scratch, exemplos e um conjunto de atividades que promovem uma maior familiaridade com o *software*.

Esperamos que esse trabalho contribua de forma significativa com a prática docente!

Abraços!!

Prof.<sup>a</sup> Airan Priscila de Farias Curci

Prof. Dr. Marcelo Souza Motta

## LISTA DE FIGURAS

Figura 1 – Tela inicial do Scratch.....	12
Figura 2 – Barra de Menus: “Arquivo” e “Editar” selecionados.....	13
Figura 3 – Ferramentas de cursor .....	13
Figura 4 – Ferramentas que muda o idioma e acessa o site do Scratch.....	14
Figura 5 – Referência do palco do Scratch.....	14
Figura 6 – Palco padrão .....	15
Figura 7 – Lista de Atores .....	15
Figura 8 – Menu do Ator .....	16
Figura 9 – Aba Blocos com a paleta Movimento selecionada .....	17
Figura 10 – Os tipos de blocos do Scratch .....	17
Figura 11 – O formato dos blocos de função e dos blocos de comando .....	18
Figura 12 – Rastro do ator ao pressionar a seta para a direita. ....	19
Figura 13 – Aba Fantasias .....	19
Figura 14 – Aba Sons .....	20
Figura 15 - Aba Panos de fundo .....	20
Figura 16 – Paint Editor: Panos de fundo e Fantasias .....	21
Figura 17 – Centro de rotação da fantasia .....	21
Figura 18 – Cor transparente .....	22
Figura 19 – Encaixe dos blocos.....	22
Figura 20 – Janela de Dicas: Bloco vire à direita (turn right) da paleta Movimento .....	23

Figura 21 – Comandos de Movimento .....	27
Figura 22 – Comando com blocos de Movimento Absoluto.....	28
Figura 23 – Comando com blocos de Movimento Relativo.....	29
Figura 24 – Fantasia criada no Paint Editor com direção original .....	30
Figura 25 – Bloco aponte para a direção .....	30
Figura 26 – Script que desenha um octógono.....	31
Figura 27 – Script do octógono com o comando repita.....	32
Figura 28 – Padrão geométrico.....	32
Figura 29 – <i>Script</i> do padrão geométrico .....	33
Figura 30 – Centro de rotação do ator .....	33
Figura 31 – Pano de Fundo underwater2.....	34
Figura 32 – Atores: Shark e Fish2 .....	35
Figura 33 – Script para o ator Shark.....	35
Figura 34 – Script para o ator Fish2 .....	37
Figura 35 – Variável pontos .....	37
Figura 36 – Comando mude x/ mude y para posição aleatória .....	38
Figura 37 – Eixos coordenados com 60 passos de distância.....	38
Figura 38 – Posições para x e y. ....	39
Figura 39 – Comando do bloco espere até .....	40
Figura 40 – Criando a variável pontos.....	40
Figura 41 – Blocos da Paleta Aparência.....	44
Figura 42 – Animação do ator Gato .....	45

Figura 43 – Blocos que alteram o pano de fundo do Palco. ....	46
Figura 44 – Blocos de fala. ....	46
Figura 45 – Blocos de pensamento. ....	47
Figura 46 – Bloco “vá para a frente” para o ator avião .....	47
Figura 47 – Bloco vá camadas para trás aplicado nos atores pedra e Ana .....	48
Figura 48 – Script que altera o tamanho do ator ET enquanto se move .....	48
Figura 49 – Efeitos Gráficos do Scratch aplicados no ator .....	49
Figura 50 – Blocos de reprodução de som .....	50
Figura 51 – Menu suspenso do Bloco toque o som .....	50
Figura 52 – Blocos que criam efeitos sonoros .....	51
Figura 53 – Menu suspenso do bloco toque o tambor por batidas .....	51
Figura 54 – Pano de fundo final .....	52
Figura 55 – Pano de fundo <i>party room</i> .....	52
Figura 56 – Pano de fundo .....	53
Figura 57 – <i>Script</i> do Pano de Fundo <i>Party Rom</i> .....	53
Figura 58 – Criando o ator Globo .....	54
Figura 59 – Importando imagem como ator de arquivo local .....	54
Figura 60 – Ator Globo no Paint Editor e seu <i>script</i> .....	55
Figura 61 – Ator Telão no Paint Editor e seu <i>script</i> .....	55
Figura 62 – Criação e <i>script</i> do ator Luz1 .....	56
Figura 63 – Fantasias do ator Cubo .....	57
Figura 64 – <i>Script</i> do ator Cubo .....	57



Figura 65– Fantasias do ator que formam os padrões geométricos.....	59
Figura 66 – <i>Script</i> que desenha os padrões geométricos .....	60
Figura 67 – Alguns padrões geométricos formados a partir das 3 fantasias .....	61
Figura 68 – Criando o bloco Desenhe o padrão .....	62
Figura 69 – Definindo o bloco Desenhe o padrão a partir do comando quando receber .....	62
Figura 70 – <i>Script</i> usando o bloco Desenhe o padrão .....	63
Figura 71 – Adicionando entrada numérica a um novo bloco.....	64
Figura 72 – Alterando o rótulo number1 .....	64
Figura 73 – <i>Script</i> que desenha o ângulo externo.....	64
Figura 74 – Criando os blocos de variáveis valor1 e valor2 .....	66
Figura 75 – As fantasias do ator menina (Avery).....	67
Figura 76 – <i>Script</i> do ator Avery .....	68
Figura 77 – Fantasias dos atores dado1 e dado2 .....	68
Figura 78 – <i>Script</i> dos atores dado1 e dado2.....	69
Figura 79 – Aparência final do projeto “Lançamento de dados” .....	69
Figura 80 – Monitor da variável valor1 exibido no Placo.....	70
Figura 81 – Monitor da variável como controle deslizante .....	70
Figura 82 – Definindo o intervalo do controle deslizante .....	71
Figura 83 – Modos de exibição do monitor.....	71
Figura 84 – Palco do projeto Volume e área da superfície da esfera .....	72
Figura 85 – <i>Script</i> dos atores “menos” e “mais”, respectivamente .....	72
Figura 86 – <i>Script</i> da esfera.....	73

Figura 87 – Bloco personalizado Recalcular da paleta Mais Blocos .....	73
Figura 88 – <i>Script</i> usando o bloco pergunte e espere a resposta .....	75
Figura 89 – <i>Script</i> que determina o número de faces de um poliedro pela Relação de Euler ..	77
Figura 90 – Criando a lista “meses do ano” .....	81
Figura 91 – Monitor de uma lista ao ser criada exibida no Palco .....	81
Figura 92 – Acrescentando itens na lista .....	82
Figura 93 – Blocos para serem usados com as listas .....	82
Figura 94 – Blocos insira e apague em execução .....	83
Figura 95 – Menu suspenso do bloco apague .....	83
Figura 96 – Blocos substitua o item e insira na posição em execução .....	84
Figura 97 – <i>Script</i> que exhibe os elementos da lista “Meses do ano” .....	84
Figura 98 – <i>Script</i> que verifica se Julho pertence à lista “Meses do ano” .....	85
Figura 99 – <i>Script</i> que adiciona na lista itens fornecido pelo usuário .....	86
Figura 100 – Tela inicial ao usuário do projeto Classificação de Sólidos Geométricos .....	87
Figura 101 – O <i>script</i> principal do ator sólido .....	87
Figura 102 – O <i>script</i> que define o bloco “mostre a forma” .....	88
Figura 103 – O <i>script</i> que define o bloco confira a resposta .....	89
Figura 104 – O <i>script</i> dos atores correspondentes aos botões .....	89
Figura 105 – As listas “Poliedro” e “Não Poliedros” .....	90

## SUMÁRIO

<b>Capítulo 1   Tela Inicial e Introdução ao Scratch</b> .....	12
INTRODUÇÃO.....	12
TELA INICIAL.....	12
Barra de Menus.....	13
Ferramenta de Cursor.....	13
Palco.....	14
Lista de Atores.....	15
Abas.....	16
Janela de Scripts.....	22
Janela de Dicas.....	22
Exercitando 1.....	23
<b>Capítulo 2   Movimentando Atores e Desenhando Padrões</b> .....	26
INTRODUÇÃO.....	26
COMANDOS DE MOVIMENTO.....	26
CRIANDO PADRÕES.....	30
EXPLORANDO BLOCOS MOVIMENTO E OUTROS BLOCOS DE COMANDO.....	34
Projeto do Jogo Shark.....	34
Exercitando 2.....	41
<b>Capítulo 3   Criando animações e introduzindo sons</b> .....	44
INTRODUÇÃO.....	44
ANIMANDO ATORES.....	44
INTRODUZINDO SONS.....	50
CRIANDO ANIMAÇÕES COM OS BLOCOS DAS PALETAS APARÊNCIA E SOM.....	52
Exercitando 3.....	58
<b>Capítulo 4   Blocos personalizados e Broadcasting</b> .....	59

INTRODUÇÃO.....	59
BROADCASTING DE MENSAGENS .....	59
BLOCOS PERSONALIZADOS .....	61
Exercitando 4.....	65
<b>Capítulo 5   As Variáveis no Scratch .....</b>	<b>66</b>
INTRODUÇÃO.....	66
A PALETA DE BLOCOS VARIÁVEIS - VARIÁVEIS .....	66
MONITORES DE VARIÁVEIS .....	70
INSERÇÃO DE DADOS DE ENTRADA PELOS USUÁRIOS .....	74
Exercitando 5.....	78
<b>Capítulo 6   As Listas no Scratch .....</b>	<b>80</b>
INTRODUÇÃO.....	80
A PALETA DE BLOCOS VARIÁVEIS – LISTAS.....	80
Exercitando 6.....	90
<b>Referências .....</b>	<b>93</b>
<b>RESPOSTAS DO EXERCITANDO .....</b>	<b>94</b>

---

# Capítulo 1 | Tela Inicial e Introdução ao Scratch

---

## INTRODUÇÃO

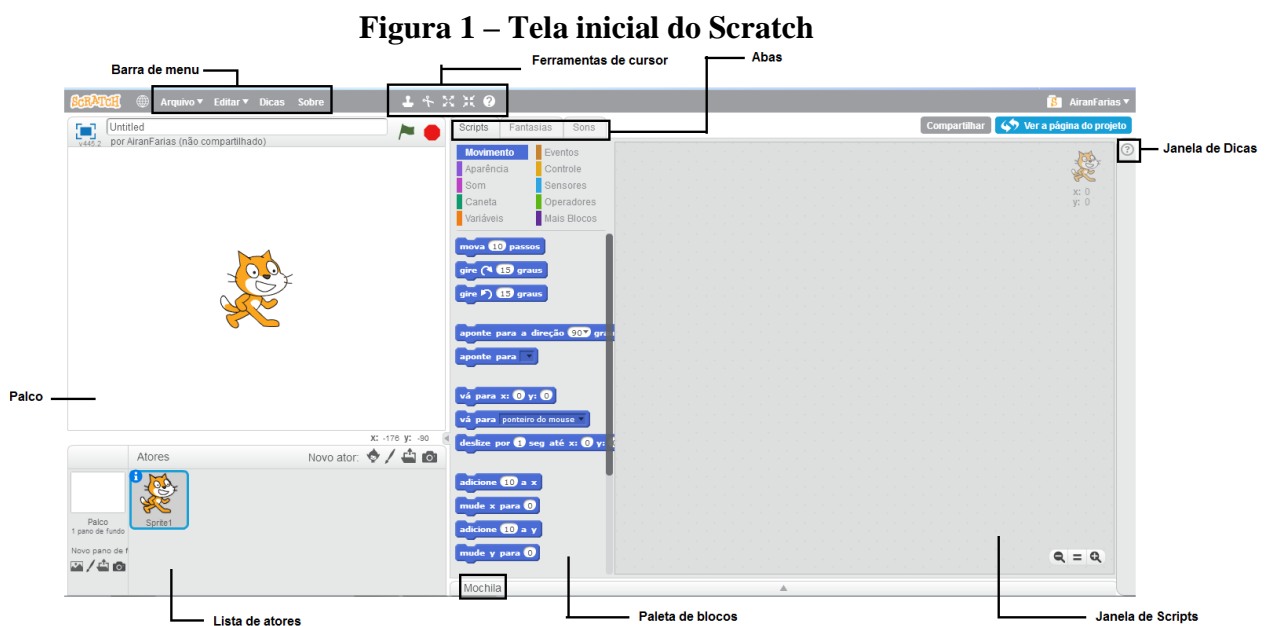
O Scratch é um software de programação gráfico desenvolvido pelo MIT (Massachusetts Institute of Technology), que permite a criação de projetos com recursos multimídia, como animações, simulações, tutoriais e jogos digitais.

O software pode ser acessado online pelo sitio do Scratch em <https://scratch.mit.edu/> ou ser feito o download do Editor Offline em <https://scratch.mit.edu/scratch2download/>.

Neste capítulo, apresentamos a tela inicial do Scratch e abordamos de forma geral o software e seus comandos.

## TELA INICIAL

Ao iniciarmos o Scratch nos deparamos com seguinte tela inicial, mostrada pela Figura 1.

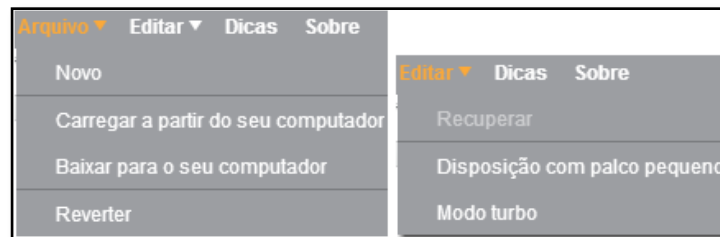


Fonte: elaborado pelos autores.

## Barra de Menus

Na barra de menus podemos: criar novos projetos, salvar e/ou carregar um projeto para o computador, desfazer alterações e recuperar a última ação apagada do projeto em andamento, deixar a disposição do palco menor de forma a maximizar a Janela de Scripts, aumentar a velocidade de alguns blocos, como por exemplo, o bloco **move**. A Figura 2 ilustra a barra de menus do Scratch.

**Figura 2 – Barra de Menus: “Arquivo” e “Editar” selecionados**



Fonte: elaborado pelos autores.

Além das opções descritas acima, encontradas em Arquivo e Editar, na barra de menus também contamos com as “Dicas” e o “Sobre”.

## Ferramenta de Cursor

Com estas ferramentas podemos duplicar, apagar, aumentar e diminuir imagens, respectivamente, como mostra a Figura 3.

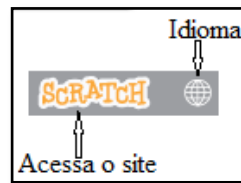
**Figura 3 – Ferramentas de cursor**



Fonte: elaborado pelos autores.

Ainda, conforme a Figura 4 é possível acessar o site do Scratch diretamente do software e trocar o idioma, que está disponível em mais de 50 línguas.

**Figura 4 – Ferramentas que muda o idioma e acessa o site do Scratch**

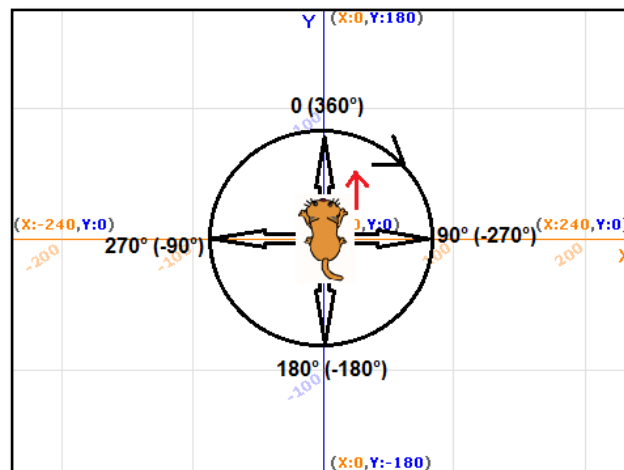


Fonte: elaborado pelos autores.

## Palco

O Palco é um plano cartesiano com 480 passos de largura (eixo x) e 360 passos de altura (eixo y). Neste espaço, o ator aponta para as seguintes direções:  $0^\circ$  sendo para cima,  $90^\circ$  para a direita,  $180^\circ$  para baixo e  $-90^\circ$  para a esquerda, como pode ser visto na Figura 5.

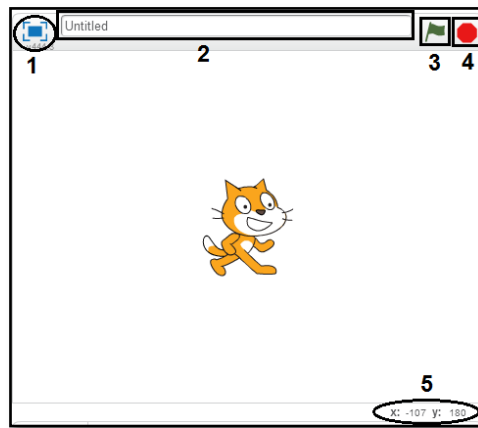
**Figura 5 – Referência do palco do Scratch**



Fonte: elaborado pelos autores.

É no Palco onde tudo acontece, ou seja, onde os Atores se movimentam e interagem. Ao iniciarmos o Scratch, o Palco sempre aparece em branco e com um único ator, o Gato, “garoto propaganda” do software. A Figura 6 mostra a tela padrão do Palco.

**Figura 6 – Palco padrão**



Fonte: elaborado pelos autores.

1. **Modo de apresentação:** Maximiza e minimiza o tamanho do Palco.
2. **Caixa de Edição:** Este espaço é destinado ao nome do projeto.
3. **Bandeira Verde:** Inicia o programa.
4. **Parar:** Finaliza o programa.
5. **Área de exibição da posição do mouse:** Exibe qual é a coordenada (x, y) do cursor no palco.

### Lista de Atores

Neste espaço ficam expostos todos os nomes e as miniaturas dos Atores do projeto.  
Ver Figura 7.

**Figura 7 – Lista de Atores**



Fonte: elaborado pelos autores.

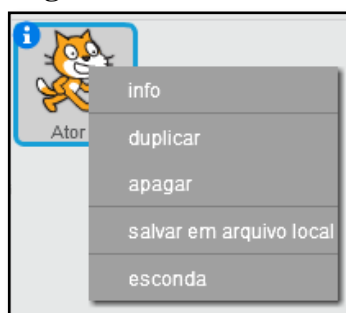
1. **Miniatura do Palco:** Ao selecionar a miniatura do Palco é possível mudar a sua fantasia e construir comandos para ele.



2. **Panos de fundo:** Nesses botões podemos alterar/ criar um novo pano de fundo para o Palco sem precisar selecionar a miniatura do Palco.
3. **Informações sobre o Ator:** é possível alterar o nome Ator, mostra as suas coordenadas no Palco (sua posição), a direção, e também é possível controlar o estilo de rotação, a sua visibilidade e sua mobilidade, podendo ser arrastado ou não.
4. **Miniatura do Ator:** é a imagem do Ator.
5. **Novo Ator:** Nesses botões, podemos escolher um novo Ator do acervo do Scratch, desenhar pelo *Paint Editor* do software, carregar uma imagem a partir de arquivos do computador ou usar a fotografia tirada pela *webcam* para dar vida a um novo Ator, respectivamente.

Cada Ator possui fantasias, sons e scripts próprios. Ao clicar com o botão direito do mouse, em cima da miniatura do Ator, é aberto um menu prático para ele, de acordo com a Figura 8.

**Figura 8 – Menu do Ator**



Fonte: elaborado pelos autores.

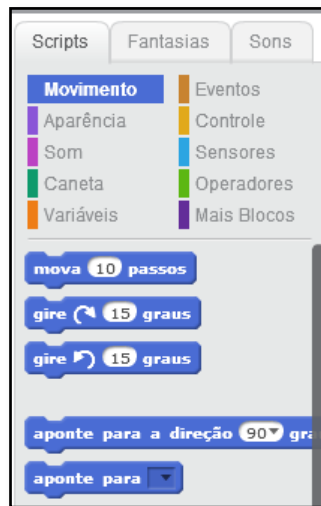
## Abas

Aqui contamos com três abas, “Scripts”, “Fantasias” e “Sons”.

### Scripts

Na aba Scripts encontramos os blocos de comando, que são a linguagem de programação do Scratch. É por meio do encaixe desses blocos que os comandos são criados. São mais de cem blocos diferenciados por cor e divididos em dez categorias. As categorias são chamadas de paletas, sendo elas: Movimento, Aparência, Som, Caneta, Variáveis, Eventos, Controle, Sensores, Operadores e Mais Blocos, como destaca a Figura 9.

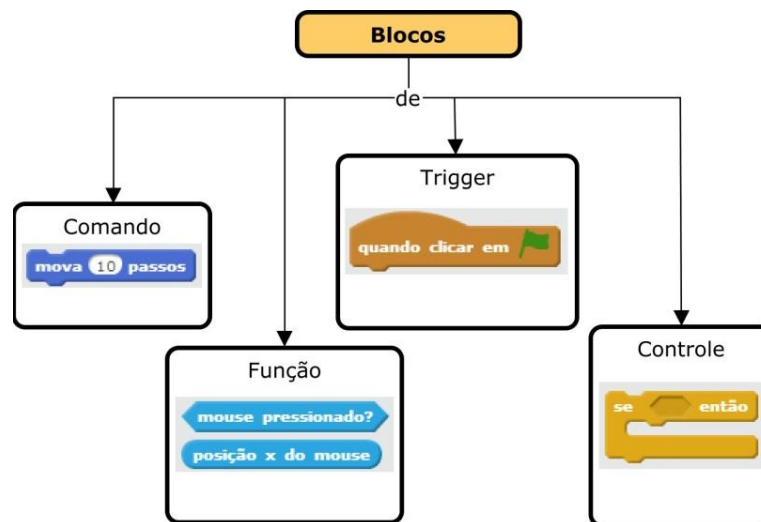
**Figura 9 – Aba Blocos com a paleta Movimento selecionada**



Fonte: elaborado pelos autores.

O Scratch possui quatro tipos de blocos, como apresentados na Figura 10.

**Figura 10 – Os tipos de blocos do Scratch**



Fonte: adaptado de Marji (2014, p. 41).

Os **blocos de comando** e os **blocos de controle** possuem uma reentrância na parte superior do bloco e uma saliência na parte inferior. Esses blocos podem ser empilhados.

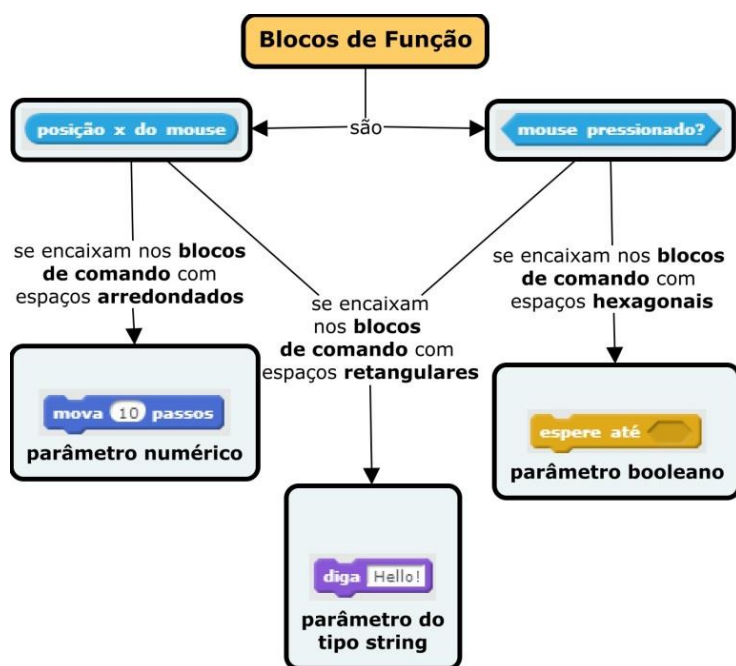
Os **blocos de função** não possuem saliências e retornam um valor. Eles são usados como entradas para outros blocos, logo não podem ser usados sozinhos para compor um script.

Os blocos de função possuem duas características:

- os blocos com extremidades arredondadas retornam números ou *strings*<sup>1</sup>.
- os blocos com extremidades pontiagudas, informam se algo é verdadeiro ou falso (valor *Booleano*).

Os blocos de comando com espaços arredondados só recebem blocos de função com extremidades arredondadas. Já os blocos de comando com espaços hexagonais recebem apenas blocos de função com as extremidades pontiagudas. Todavia, os blocos de comando com espaços retangulares aceitam qualquer bloco de função. A Figura 11, resume essas funções.

**Figura 11 – O formato dos blocos de função e dos blocos de comando**



Fonte: elaborado pelos autores

Os **blocos de trigger** possuem a parte superior arredondada e executam os blocos que estão abaixo dele. Eles figuram como um “chapéu”, pois são colocados sempre no topo de uma pilha de blocos.

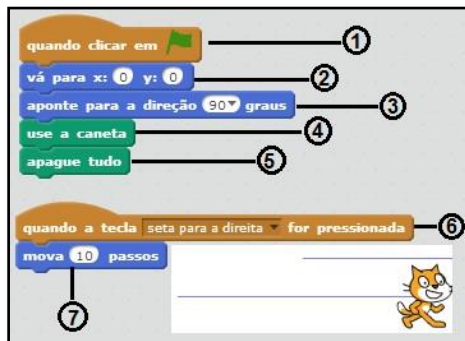
<sup>1</sup>String é um tipo de dado que serve para armazenar nomes, endereços, títulos de livros. Isto é, trata-se de uma sequência de caracteres incluindo letras, tanto maiúsculas quanto minúsculas, números e símbolos que podem ser digitados a partir do teclado do computador.

---

### Algo a mais 1...

Um comando interessante para darmos ao ator é quando podemos ver o “rastros” que ele deixa ao ser movimentado usando o bloco de comando Caneta. Combinando esse bloco com o bloco trigger, podemos fazer desenhos usando as setas do teclado (esquerda, direita, para cima e para baixo). A Figura 12 exemplifica essa ação.

**Figura 12 – Rastro do ator ao pressionar a seta para a direita.**



Quando clicarmos na bandeira verde (1) o ator irá para o centro do palco (2) voltado para a direção 90° (3). A caneta do ator será acionada (4) e o que foi feito será apagado (5).

Quando a seta para a direita for pressionada (6) o ator se moverá (7) neste sentido deixando seu rastro.

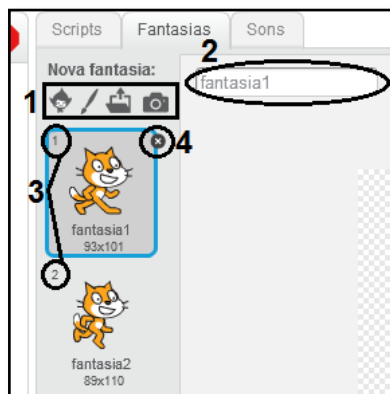
Fonte: elaborado pelos autores.

---

### Fantasias

A aba Fantasias é como se fosse um guarda-roupa ( Figura 13). Nela alteramos a aparência do Ator (1), o seu nome (2), gerenciamos a ordem das fantasias (3) e apagamos-las (4).

**Figura 13 – Aba Fantasias**



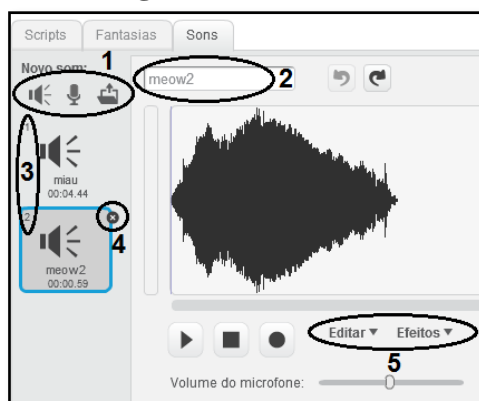
Fonte: elaborado pelos autores.

---

## Sons

A aba Sons permite que os Atores reproduzam sons. Esta aba funciona como a aba Fantasias. Nela, podemos adicionar sons por meio do acervo do Scratch, gravar um áudio ou escolher um arquivo do computador, conforme o item (1) da Figura 14. O Scratch lê somente arquivos de áudio no formato MP3 e WAV. Neste espaço, também podemos alterar o nome do som, gerenciar a ordem, apagar e editar (Figura 14).

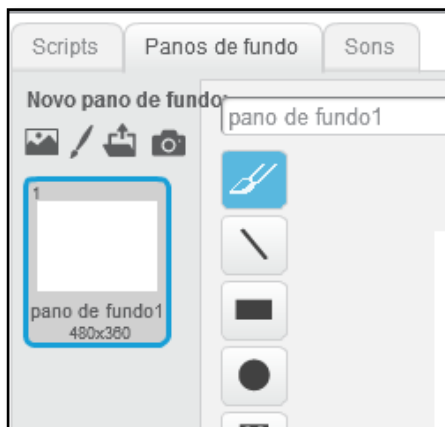
**Figura 14 – Aba Sons**



Fonte: elaborado pelos autores.

Além das três abas descritas, quando selecionamos a miniatura do Palco na Lista de Atores, a aba Fantasias muda para Panos de fundo, como mostra a Figura 15. Esta aba é semelhante a aba Fantasias, sendo possível mudar as imagens de fundo e gerenciá-las.

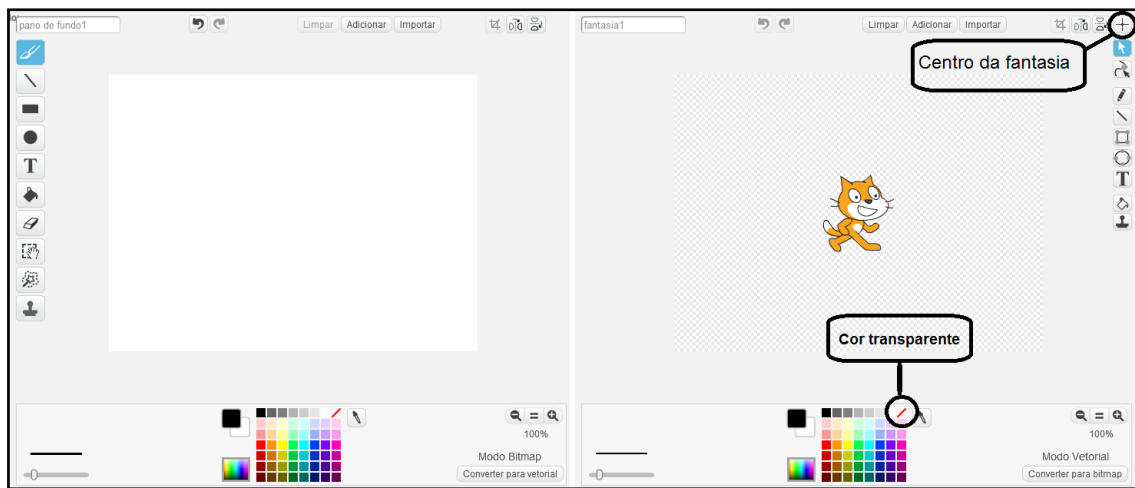
**Figura 15 - Aba Panos de fundo**



Fonte: elaborado pelos autores.

Ainda nas abas Fantasias e Panos de fundo podemos encontrar o *Paint Editor* do Scratch, em que podemos criar e/ ou editar fantasias e panos de fundo, como apresentado na Figura 16.

**Figura 16 – Paint Editor: Panos de fundo e Fantasias**

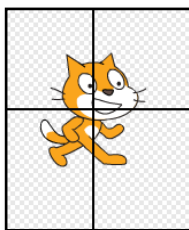


Fonte: elaborado pelos autores.

No *Paint Editor* da aba Fantasias, é importante destacar, o recurso que define o centro da imagem (fantasia) e a cor “transparente” (a ausência de cor).

Os Atores giram/ viram em relação a um ponto de referência, que é o centro da fantasia. Ver Figura 17.

**Figura 17 – Centro de rotação da fantasia**

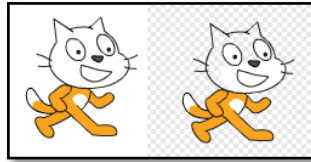


Fonte: elaborado pelos autores.

O centro da fantasia é determinado pela interseção dos dois eixos. Ele pode ser alterado de duas formas: arrastando os eixos para o novo centro ou ao clicar no ponto desejado.

A cor transparente permite que a aparência do Palco seja vista por trás da imagem que o sobrepõe, como demonstrado na Figura 18.

**Figura 18 – Cor transparente**

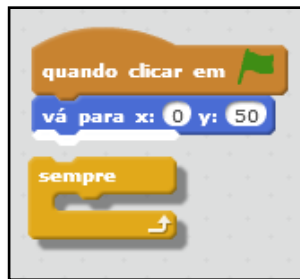


Fonte: elaborado pelos autores.

### Janela de Scripts

Na Janela de Scripts é onde programamos o que o Ator irá fazer. Para isso, arrastamos os blocos de comando da aba blocos para esse espaço, encaixando um ao outro. Os blocos que são possíveis serem encaixados destaca uma área em branco, conforme a Figura 19, quando são aproximados, feito isso basta soltar o bloco que ele se encaixa.

**Figura 19 – Encaixe dos blocos**



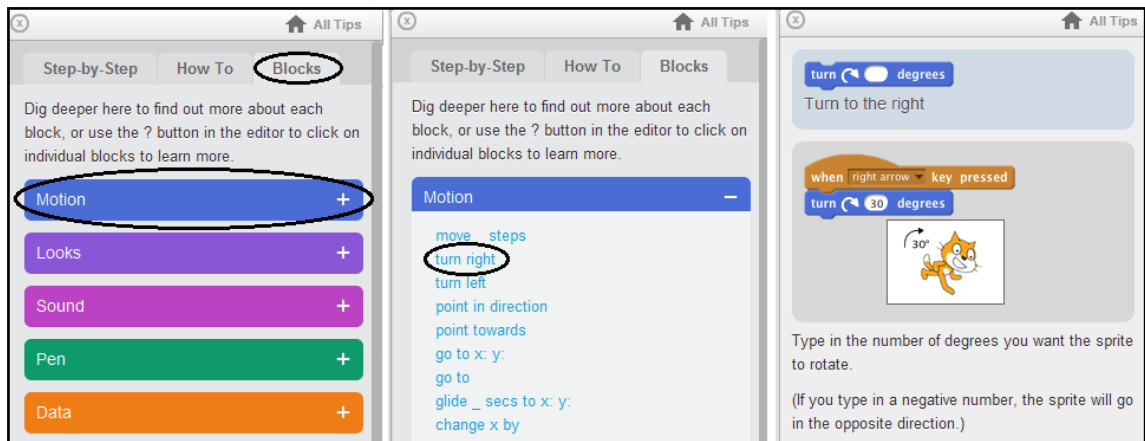
Fonte: elaborado pelos autores.

### Janela de Dicas

Nesta janela, como o próprio nome sugere, encontramos as “dicas” sobre as funções das ferramentas do software. Contudo, está em inglês, idioma original do Scratch.

Destacamos na Janela de Dicas, as informações a respeito dos blocos, que são breves descrições do funcionamento de cada bloco da paleta (Figura 20).

**Figura 20 – Janela de Dicas: Bloco vire à direita (turn right) da paleta Movimento**



Fonte: elaborado pelos autores.

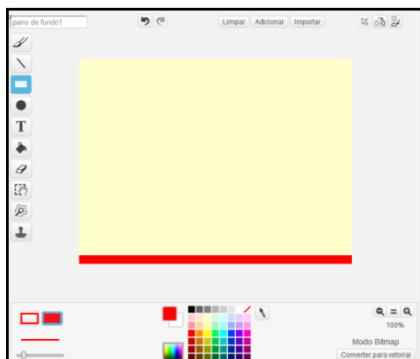
## Exercitando 1

1. Crie um script de forma que o ator diga que irá calcular a área de um trapézio com o valor da medida das bases sendo  $\frac{5}{9}$  e  $\frac{22}{9}$  e altura medindo  $\frac{2}{3}$  e que, em seguida, dê o resultado dessa expressão.
2. Construa um script onde o ator com a fantasia *pencil-a* construa um quadrado.
3. Analise a construção do Jogo ping-pong.

## Jogo ping-pong



### Pano de fundo: preparação do palco



- 1 Apague o ator Gato na Lista de Atores;
- 2 Mude a cor do placô;



- 3 Crie uma faixa retangular vermelha na parte inferior do palco.



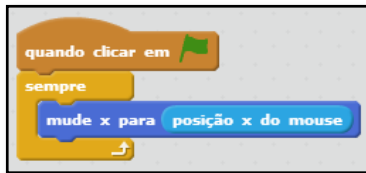
### Novos atores

1. Adicione o ator bola e o ator raquete da Lista de Atores;

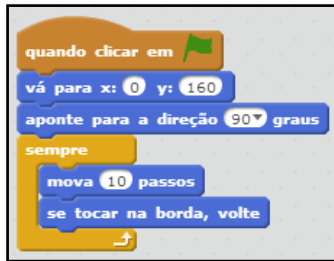


### Criando Scripts

1. Comando de execução da raquete

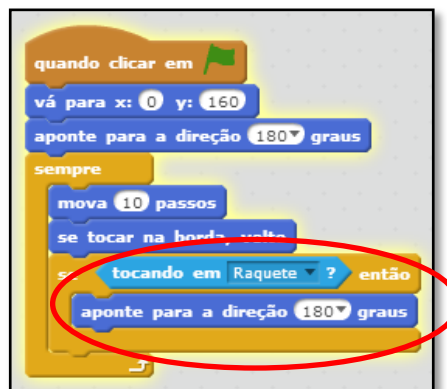


2. Comando de execução da bola

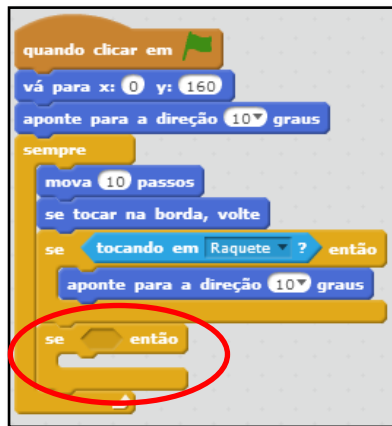


3. Comandos que fazem a raquete rebater a bola

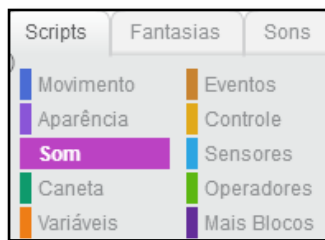
Acrescente ao script da bola no **bloco sempre** o seguinte comando:



4. Comandos que fazem a bola parar ao tocar a faixa retangular vermelha.



5. Introduzir som



Agora, faça as alterações necessárias nos scripts de forma que:

- a) A raquete sempre comece na mesma posição ao clicarmos na bandeira verde.
- b) A bola tome direções aleatórias ao tocar na borda do placó e na raquete.
- c) Insira os blocos adequados no script da Figura 4 no **bloco se, então** de forma que a bola pare ao tocar a superfície vermelha na parte inferior do palco.
- d) O ator bola tenha som ao tocar a raquete.

---

## Capítulo 2 | Movimentando Atores e Desenhando Padrões

---

### INTRODUÇÃO

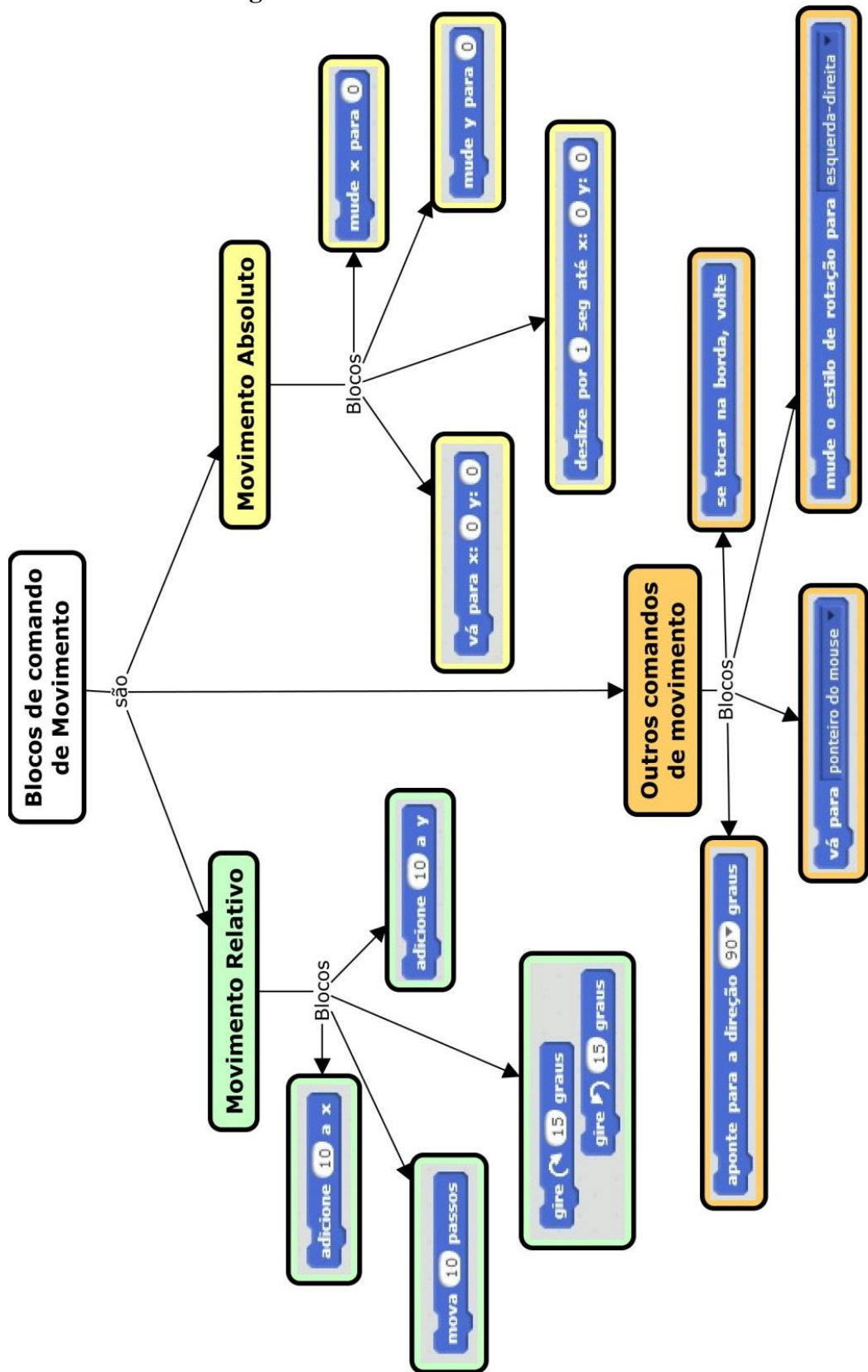
Como vimos no capítulo anterior, para que o Ator faça coisas é necessário dar os comandos a ele. Neste tópico, falaremos sobre os blocos dão movimento ao ator e dos blocos que otimizam a construção de padrões por meio da repetição.

### COMANDOS DE MOVIMENTO

Foi visto no capítulo 1, que o espaço em que o Ator se movimenta é um plano cartesiano, com centro na origem. O eixo x varia de -240 a 240 passos e o eixo y varia de -180 a 180 passos. Para que o Ator se movimente por esse espaço, precisamos dar comandos a ele.

O Scratch possui blocos de comandos de **movimento relativo**, que são aqueles que dependem de algo ou são relativos a algo, os de **movimento absoluto**, que são aqueles que levam diretamente para a posição desejada e mais alguns outros tipos de comandos de movimento conforme mostra a Figura 21.

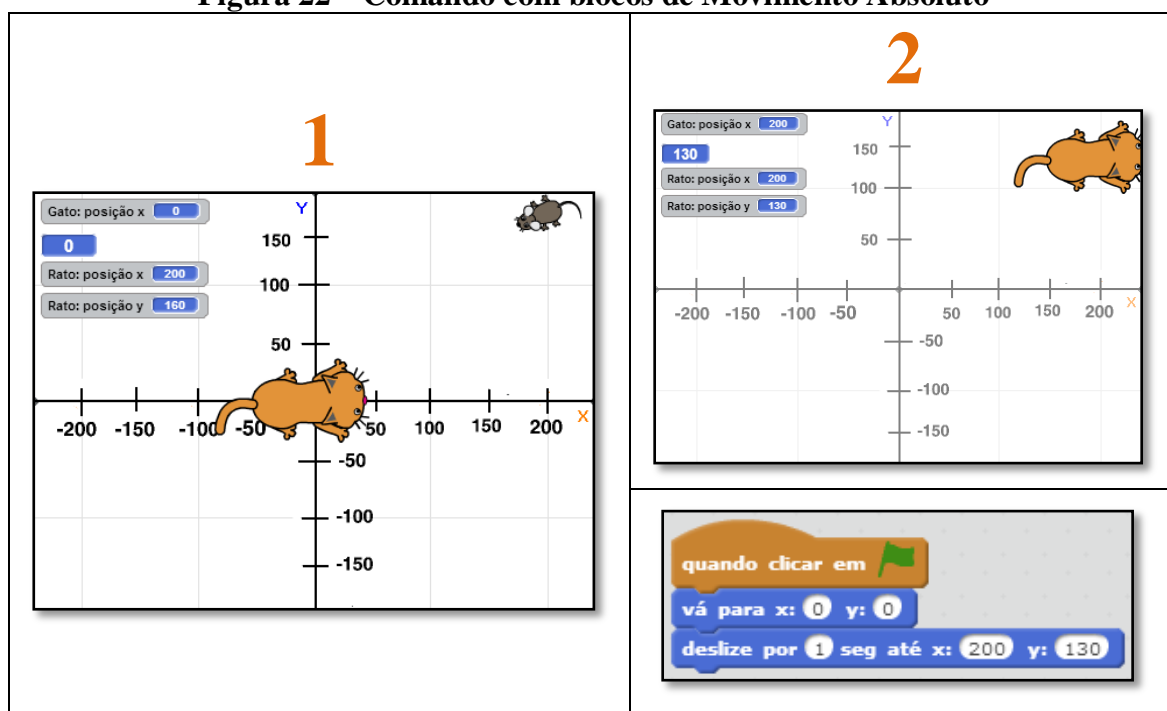
Figura 21 – Comandos de Movimento



Fonte: elaborado pelos autores.

Para exemplificar esses dois tipos de comandos, considere a situação abaixo representada pela Figura 22, em que é mostrado um script com blocos de movimento absoluto. Nela, há dois atores, o gato e o rato. O gato tem por objetivo alcançar o rato. A posição inicial do gato é (0, 0) e do rato é (200, 130) na imagem (1). Usando os blocos de movimento absoluto **vá para** e **deslize por** e a posição do rato, o script mostrado na imagem (3) ilustra o comando dado ao gato, em que: quando a bandeira verde for clicada, o gato irá para a posição (0, 0) e, em seguida, deslizará durante um segundo até a posição (200, 130), coincidindo com a posição do rato.

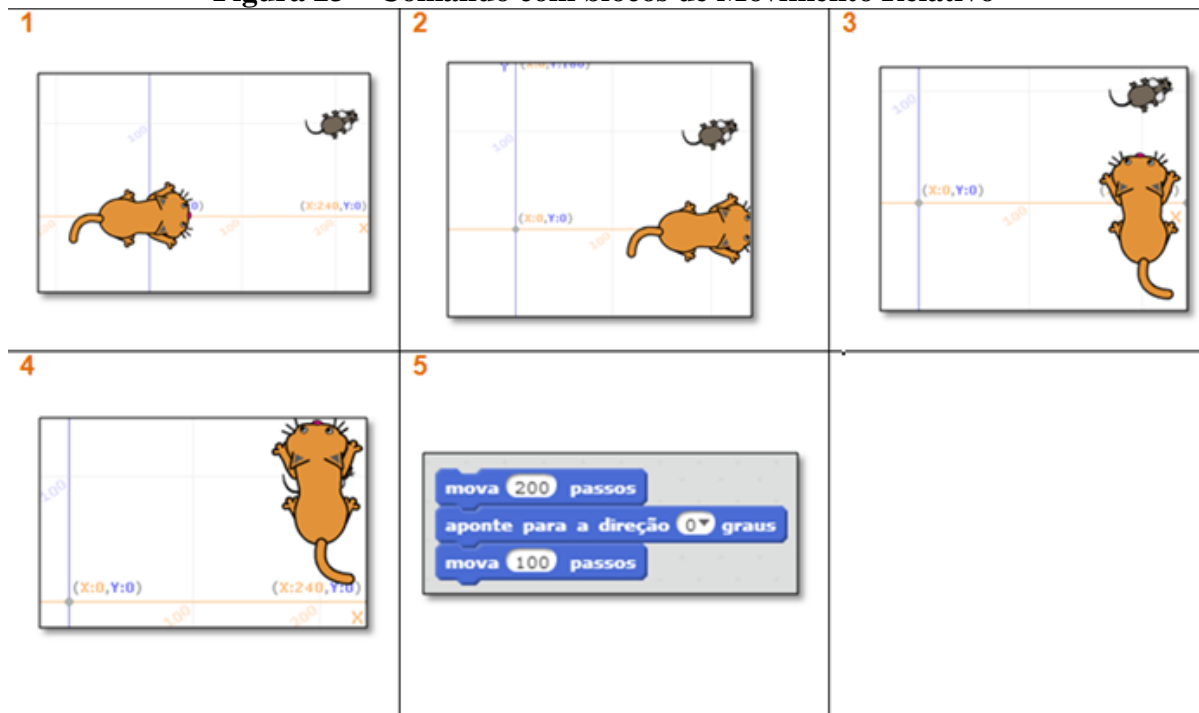
**Figura 22 – Comando com blocos de Movimento Absoluto**



Fonte: elaborado pelos autores.

Já a Figura 23 trás a mesma situação, no entanto, usa blocos de movimento relativo.

Figura 23 – Comando com blocos de Movimento Relativo

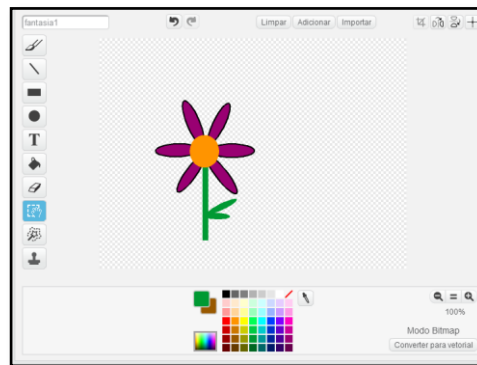


Fonte: elaborado pelos autores.

Segundo a Figura 23, com o bloco **mova passos** o ator gato irá mover a quantidade de passo designados, neste caso 200 passos. Em seguida, apontará para a direção  $0^\circ$  (com o bloco **aponte para**), isto é, para cima, e por fim, andará mais 100 passos com o bloco de movimento relativo **mova passos**.

Vale ressaltar, que o ator está originalmente orientado na direção  $90^\circ$ , ou seja, voltado para a direita. Na imagem 1 da Figura 22 e Figura 23, temos o ator gato como um bom exemplo, pois ele está em sua posição e direção original, direcionado para a direita (direção  $90^\circ$ ) e posicionado no centro do palco na posição (0, 0). Com isso, as fantasias que forem criadas no *Paint Editor* do Scratch, na aba Fantasias, conforme apresentado no Capítulo 1, também estarão voltados para a direita, de acordo com a Figura 24.

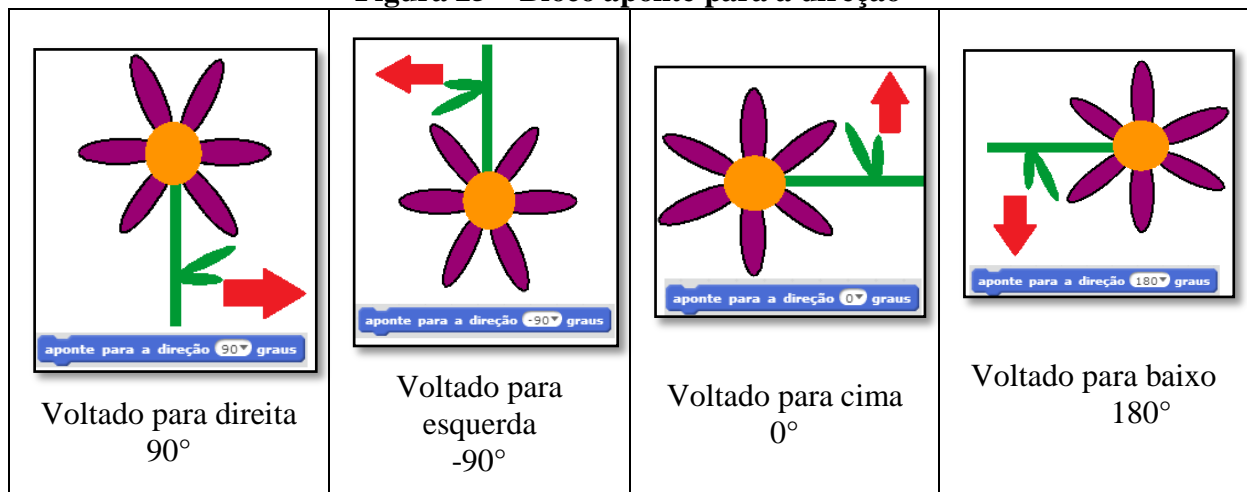
**Figura 24 – Fantasia criada no Paint Editor com direção original**



Fonte: elaborado pelos autores.

Usando o comando **aponte para a direção**, observe na Figura 25 como a flor se comporta.

**Figura 25 – Bloco aponte para a direção**



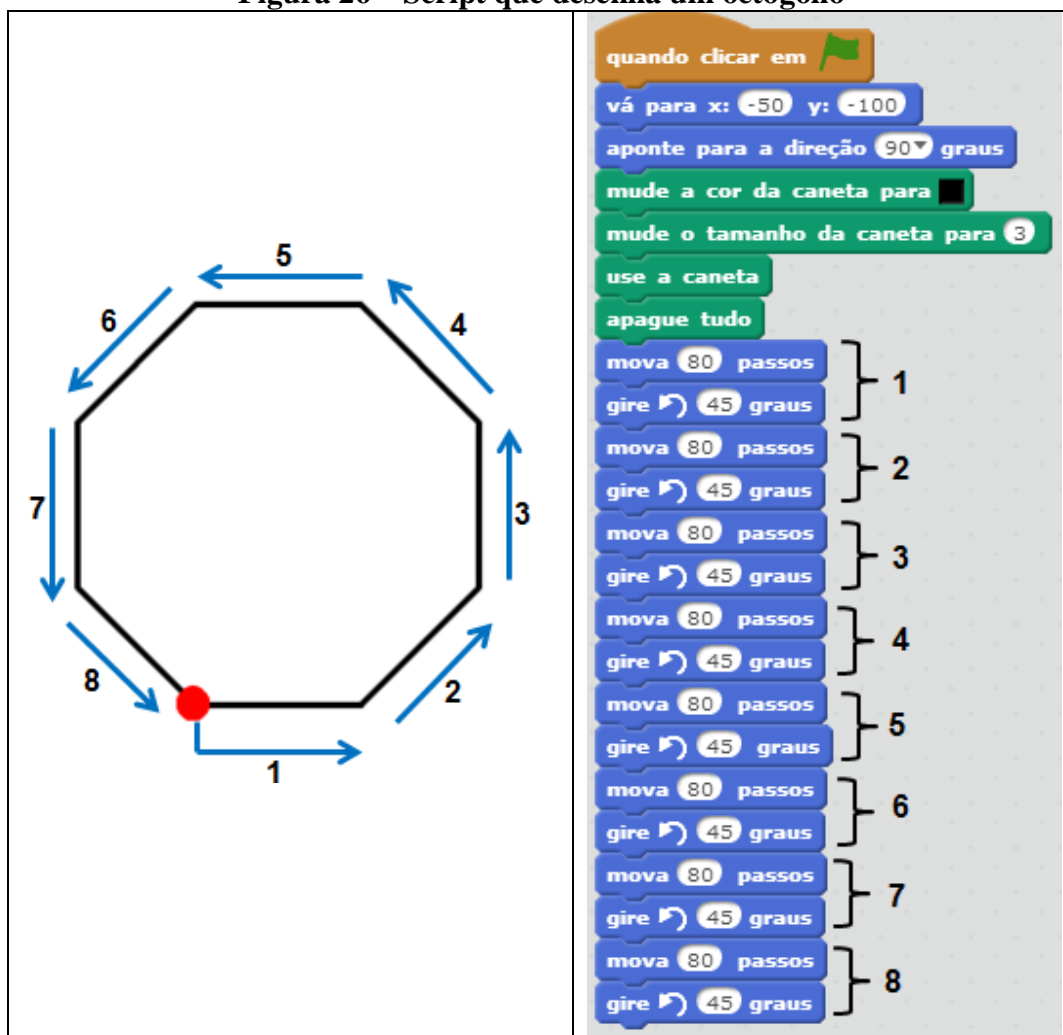
Fonte: elaborado pelos autores.

Com isso, para que o ator responda ao comando **aponte para a direção** conforme a orientação do palco do Scratch (ver Figura 5), ao desenhar a fantasia do ator no *Paint Editor*, esta deve estar voltada para a direita.

## CRIANDO PADRÕES

Um comando muito útil é o bloco de Controle **repita**. A Figura 26, trás o desenho de um octógono como exemplo.

Figura 26 – Script que desenha um octógono



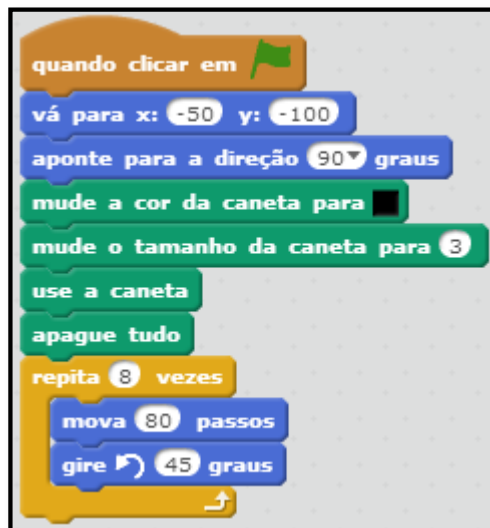
Fonte: elaborado pelos autores.

O script apresentado na Figura 26 utiliza os blocos de comando **mova passos** e **gire**. A forma ele está é muito extensa e trabalhosa, mesmo usando a ferramenta **duplicar**, para duplicar os comandos. Caso queiramos alterar o número de passos do bloco **mova passos**, teríamos que mudar de todos os oito blocos. Para projetos simples como o desenho de um octógono isso pode ser irrelevante, mas para programas mais extensos e sofisticados, esse tipo de comando pode se tornar um problema.

Para casos como esse, em que usamos muitos comandos repetidos, uma ótima ferramenta é o bloco de controle **repita**, o qual transforma scripts longos, em outros com apenas poucos comandos. Veja na Figura 27 o script do octógono com o comando **repita**.



Figura 27 – Script do octógono com o comando repita

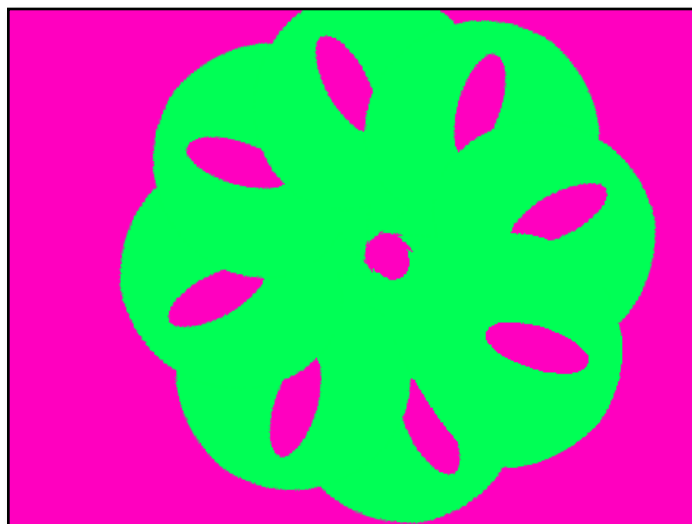


Fonte: elaborado pelos autores.

Ainda vale destacar aqui, que o desenho da figura será feito de acordo com a direção em que estiver o ator. Nesse caso, o ator “ponto vermelho” está voltado para a direita (posição 90°), sendo assim, após realizar o desenho do octógono ele retornará para a posição inicial (Figura 26).

Combinando outros comandos com o bloco **repita**, podemos criar padrões cada vez mais incríveis, como ilustra a imagem da Figura 28.

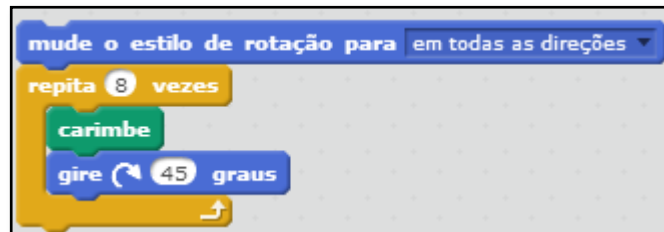
Figura 28 – Padrão geométrico



Fonte: elaborado pelos autores.

Para criar essa figura, combinamos o comando **repita** com os comandos: **carimbo**, **gire** e **mude o estilo de rotação para** com a opção em *todas as direções*. A Figura 29 mostra esse script.

**Figura 29 – Script do padrão geométrico**

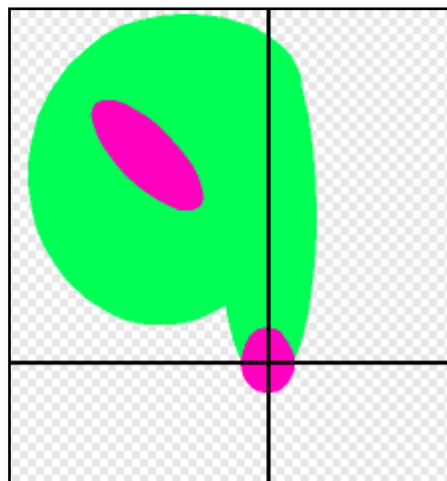


Fonte: elaborado pelos autores.

- 🔗 **Mude o estilo de rotação para** *em todas as direções*: este bloco faz com que o ator gire para todas as direções do palco;
- 🔗 **Repita**: este bloco é o que faz as repetições dos comandos internos a ele.
- 🔗 **Carimbe**: este comando permite a criação de figuras como àquela apresentada na Figura 28.
- 🔗 **Gire**: este comando diz em direção o ator deve girar.

A fantasia do ator foi feita no *Paint Editor* e para que as rotações fossem feitas de forma a gerarem a figura desejada (Figura 28) o centro de rotação da fantasia foi definido em sua parte inferior, como mostra a Figura 30.

**Figura 30 – Centro de rotação do ator**



Fonte: elaborado pelos autores.

Os giros feitos pelo ator sempre são feitos em torno do centro de rotação da fantasia.

## EXPLORANDO BLOCOS MOVIMENTO E OUTROS BLOCOS DE COMANDO

Neste tópico, será explorado um pouco mais dos blocos **Movimento** e outros **blocos de comando** por meio do script do Jogo Shark.

### Projeto do Jogo Shark

#### Preparando o Palco: Escolhendo o Pano de Fundo

Primeiramente escolhemos o Pano de Fundo para o Palco. No acervo de Panos de Fundo do Scratch contamos com três temas para fundo do mar, no entanto, lembrando que podemos criar um novo pelo *Paint Editor*, carregar um arquivo do computador ou até mesmo usar uma imagem fotografada pela *webcam*. Para esse projeto foi escolhido o tema *underwater2* do acervo do Scratch. Ver Figura 31.

**Figura 31 – Pano de Fundo underwater2**



Fonte: elaborado pelos autores.

Escolhido o pano de fundo, escolheremos os atores que atuarão no projeto. Dele farão parte um tubarão (ator Shark) e um peixe (ator Fish2), ambos escolhidos a partir do acervo do Scratch e com as cores de suas fantasias alteradas na aba Fantasias.

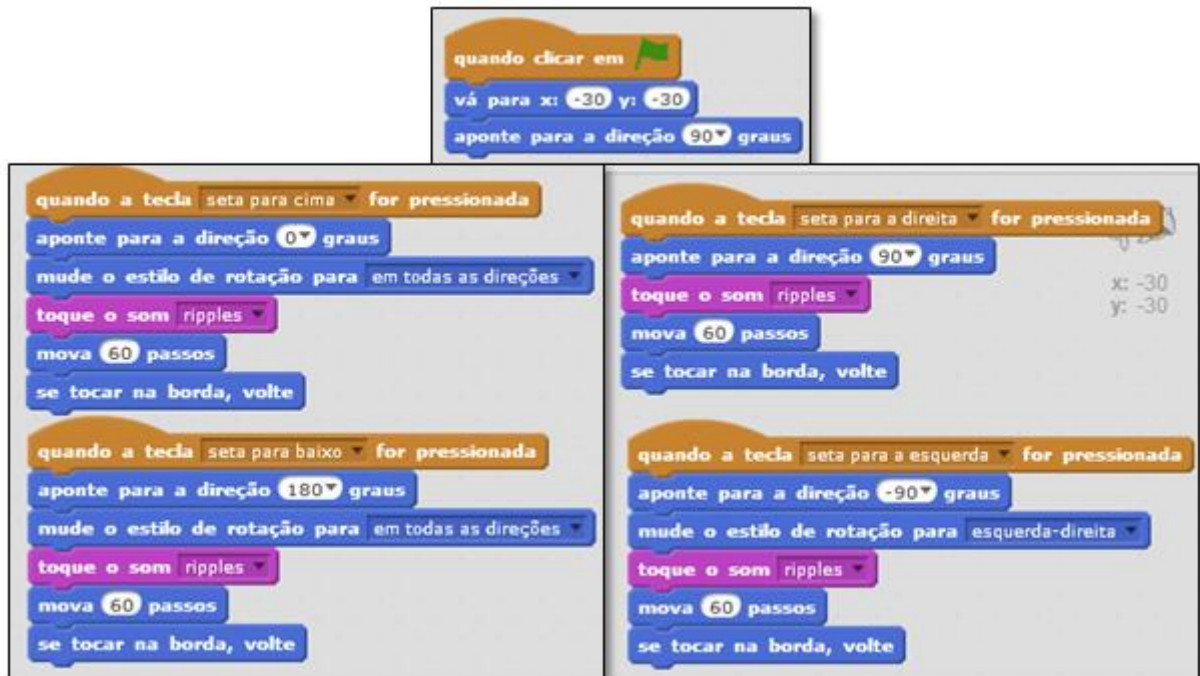
**Figura 32 – Atores: Shark e Fish2**



Fonte: elaborado pelos autores.

Preparado o palco com todos os atores desejados, partiremos para a criação do script do ator Shark, conforme a Figura 33.

**Figura 33 – Script para o ator Shark**



Fonte: elaborado pelos autores.

O script diz que quando a bandeira verde for acionada, o ator Shark irá para a posição (-30, -30) e se voltará para a direita. Os outros quatro scripts são para que o ator responda aos comandos das teclas de direção quando pressionadas. Observe que esses scripts são bem parecidos.

- **Quando a tecla para a direita for pressionada:**

1. O ator Shark se voltará para a direita;
2. Emitirá um som (escolhido do acervo da aba Som);
3. Moverá 60 passos a cada toque;
4. Retornará caso toque na borda do palco.

▪ **Quando a tecla para a esquerda for pressionada:**

1. O ator Shark se voltará para a esquerda;
2. O comando **mude o estilo de rotação para esquerda-direita** para que o ator não fique de ponta cabeça;
3. Emitirá um som (escolhido do acervo da aba Som);
4. Moverá 60 passos a cada toque;
5. Retornará caso toque na borda do palco.

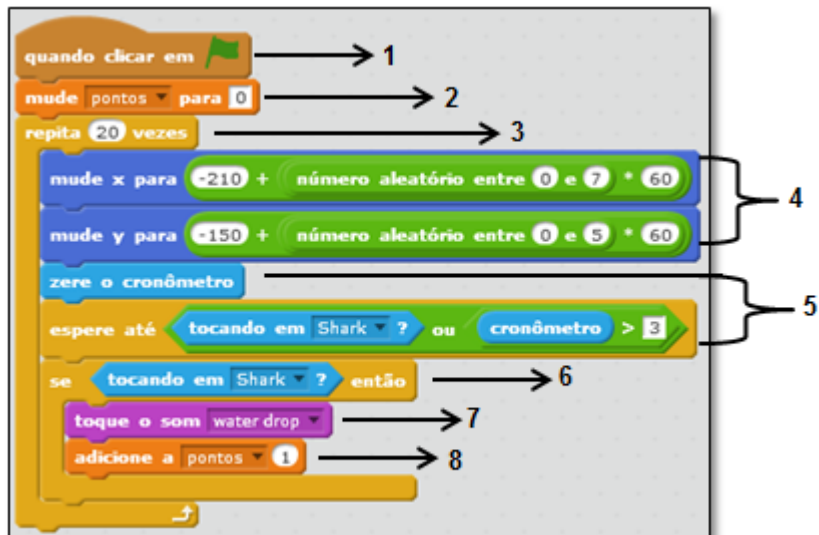
▪ **Quando a tecla para a cima ou para baixo for pressionada:**

1. O ator Shark se voltará para cima ou para baixo, respectivamente;
2. O comando **mude o estilo de rotação para todas as direções** fará com que o a boca do tubarão fique voltada para a direção acionada;
3. Emitirá um som (escolhido do acervo da aba Som);
4. Moverá 60 passos a cada toque;
5. Retornará caso toque a borda do palco.

Concluído o script para o ator Shark, é a vez de construir um script para o ator Fish2.

Ver Figura 34.

Figura 34 – Script para o ator Fish2



Fonte: elaborado pelos autores.

Como o script do ator Shark, este também iniciará quando a bandeira verde for clicada (1). Como desejamos contar quantos peixes o tubarão come, criamos uma variável denominada **pontos** na paleta Variáveis (2) que contabilizará essa quantidade, e o visor ficará visível no palco (Figura 35).

Figura 35 – Variável pontos



Fonte: elaborado pelos autores.

Em seguida, há o bloco **repita**, o qual mostrará 20 vezes (ou a quantidade que desejar) o peixe ao jogador (3) em uma posição aleatória (4). Também, dentro desse comando, o jogador terá um tempo para pegar o peixe (5) até que ele mude para a posição aleatória seguinte, de forma que o tempo sempre será iniciado em zero. Com o comando **espere até**, o peixe irá esperar até que o tubarão o toque ou que o tempo ultrapasse 3 segundos (pode alterá-

lo de forma a facilitar ou dificultar o jogo). Caso o tubarão toque o peixe, o bloco **se, então** (6) executará os comandos internos a ele, emitindo um som ao tocá-lo (7) e adicionando 1 ponto (8). Já, se o tempo de 3 segundos for ultrapassado, o peixe mudará para uma posição aleatória.

Neste script, ainda vale explicitar mais detalhadamente alguns comandos que são executados pelo bloco **repita**.

Os comandos que fazem o ator Fish2 mudar para uma posição aleatória (Figura 34 - 3) foram montados sequencialmente, como mostra a Figura 36:

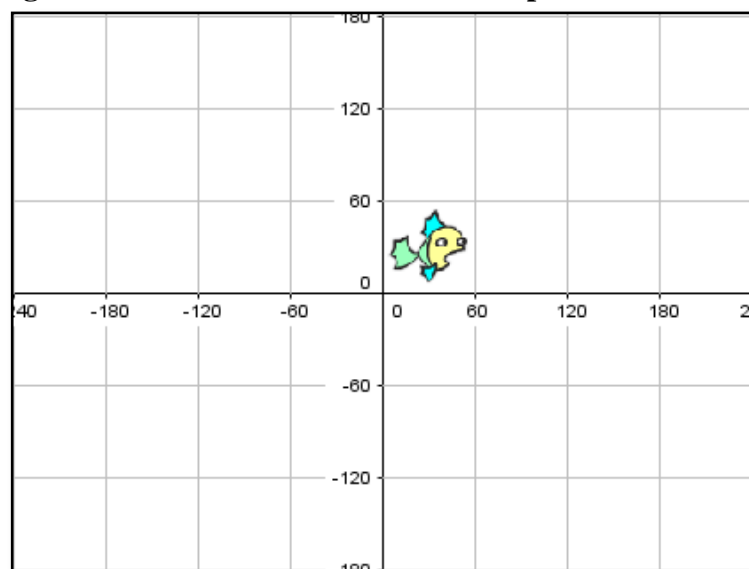
**Figura 36 – Comando mude x/ mude y para posição aleatória**



Fonte: elaborado pelos autores.

Os números que compõem os blocos não foram escolhidos arbitrariamente, mas estão de acordo com os eixos coordenados do palco do Scratch. Ver Figura 37.

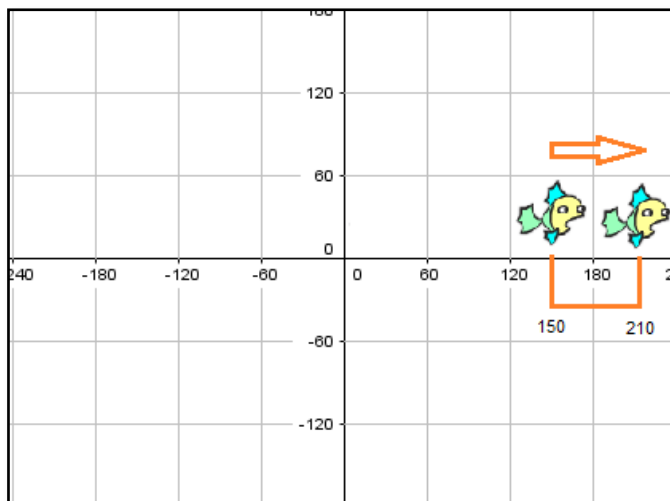
**Figura 37 – Eixos coordenados com 60 passos de distância**



Fonte: elaborado pelos autores.

Cada espaço corresponde a 60 passos, isto é, cada vez que o ator se move ele ocupa um quadrado do plano. Desta forma, a posição x do ator Fish2 pode ir de -210 a 210 e a posição y de -150 a 150. Ver Figura 38.

**Figura 38 – Posições para x e y.**



Fonte: elaborado pelos autores.

Assim, para que o peixe apareça aleatoriamente nos quadrados do Palco colocamos para:

- 🐟 **Posição x:** escolher um número aleatório entre 0 e 7 ( $7 \times 60 \div 2 = 210$ ), multiplicar esse número por 60 e por fim, adicionar a -210.
- 🐟 **Posição y:** escolher um número aleatório entre 0 e 5 ( $5 \times 60 \div 2 = 150$ ), multiplicar esse número por 60 e por fim, adicionar a -150.

Como falamos anteriormente, após o ator Fish2 mudar para uma posição aleatória, este ficará nessa posição durante 3 segundos ou até que o tubarão o alcance. Sendo assim, ele foi montado sequencialmente com os blocos apresentados na Figura 39):



Figura 39 – Comando do bloco espere até



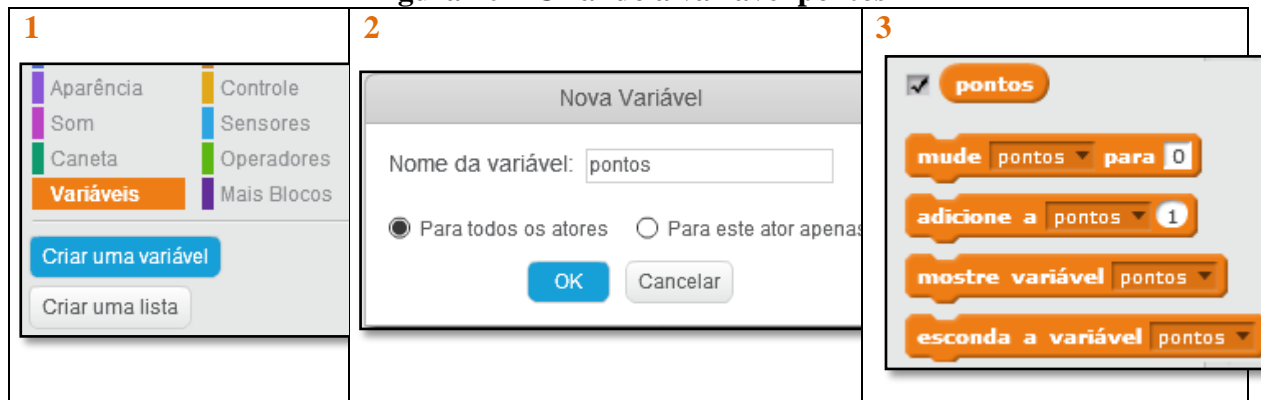
Fonte: elaborado pelos autores.

---

### Algo a mais 2...

Nomes como **pontos** (Figura 2.15) o qual usamos para contabilizar a quantidade de peixes alcançados são chamados de *variáveis*. Blocos desse tipo são criados na paleta Variáveis. Ver Figura 40.

Figura 40 – Criando a variável pontos



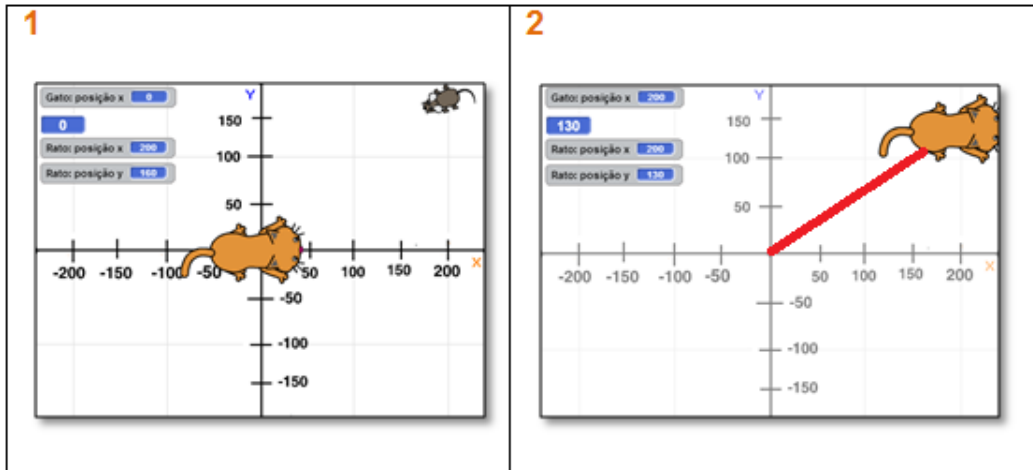
Fonte: elaborado pelos autores.

Essa paleta será melhor explorada no capítulo 5.

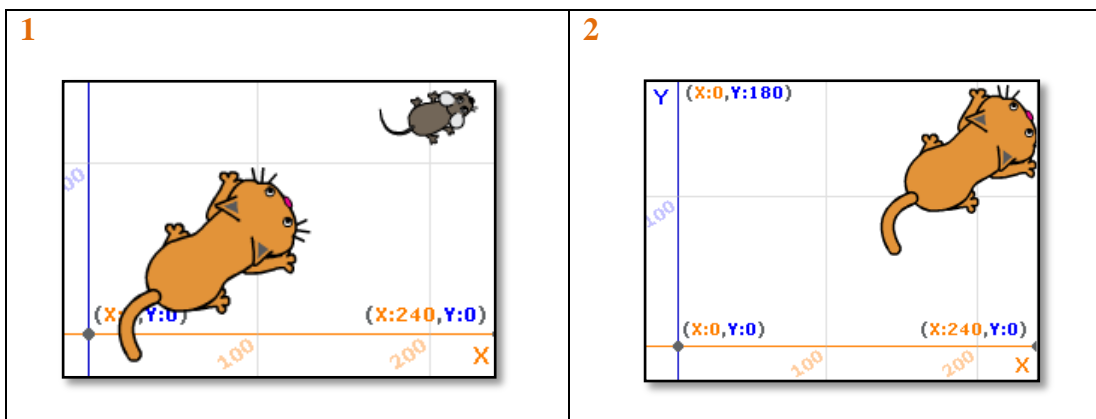
---

## Exercitando 2

1. Observe no script abaixo. Ao ser executado, o ator Gato não irá virar para o ator Rato ao alcançá-lo.



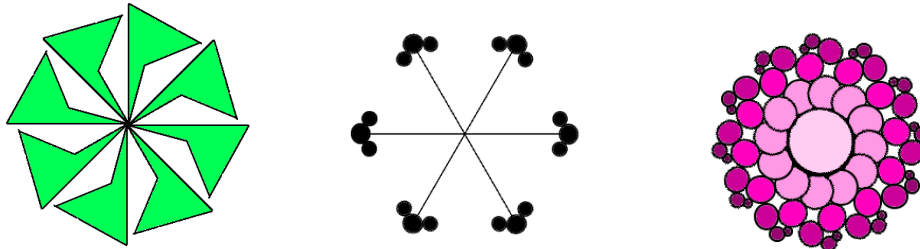
Insira um bloco de comando no script de forma que o Gato esteja virado em direção ao Rato ao alcançá-lo, conforme as imagens abaixo.



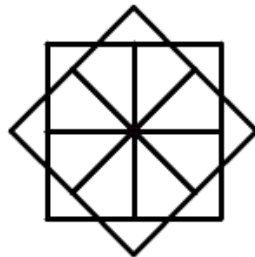
2. Crie um script com o comando **repita** de forma que com ele possam ser criados:

- a) Um triângulo.
- b) Um quadrado.
- c) Um pentágono.
- d) Um hexágono.

3. Escolha um dos padrões abaixo e crie um script para desenhá-lo.



4. Considere o padrão geométrico criado e seu script. Recrie o script adicionando os blocos de comando da Caneta necessários, em seguida execute-o e explique como o script funciona.



5. Este é o jogo **Viajando no Espaço**. Ele contém dois atores, o foguete e o planeta. Ao clicar na bandeira verde, o ator planeta cria 100 clones que ficam caindo pelo espaço, como ilustra a Figura 1. O ator foguete se move para a direita quando a tecla de direção para a direita é pressionada e para a esquerda quando a tecla para a esquerda é pressionada, e quando tocar na borda do palco retorna.

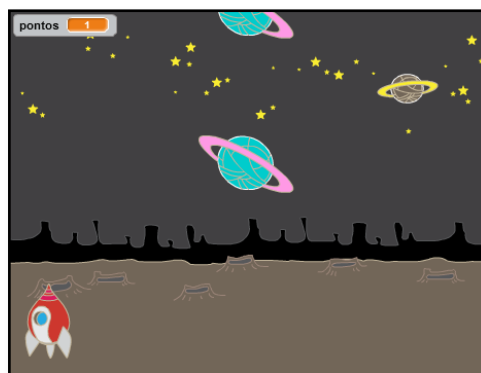


Figura 1: Palco do Viajando no Espaço

Monte o palco de acordo com a Figura 1 e com os respectivos atores foguete e planeta. Em seguida, recrie os scripts abaixo (Figura 2) e acrescente os blocos necessários para completar o jogo de forma que ele:

- a) Contenha o bloco **pontos** (da paleta Variáveis) visível no palco.
- b) Que o ator foguete se mova para a esquerda quando a tecla para esquerda for pressionada, para a direita quando a tecla para a direita for pressionada e que retorne ao tocar na borda do palco.
- c) Quando o ator planeta for tocado pelo foguete, um ponto seja acrescentado na variável **pontos** e um som seja emitido.

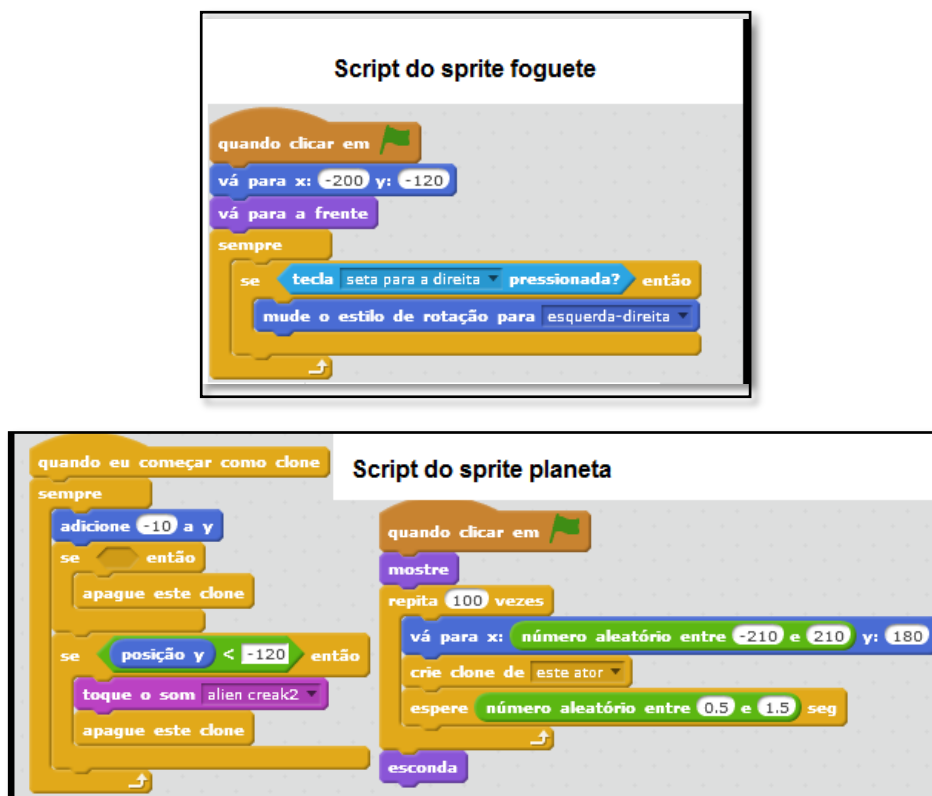


Figura 2: Script do ator foguete/ Script do ator planeta

---

## Capítulo 3 | Criando animações e introduzindo sons

---



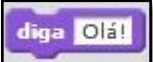






### INTRODUÇÃO

No capítulo anterior vimos como movimentar atores. Neste capítulo, criaremos animações e efeitos de imagem utilizando os blocos da paleta Aparência e introduzindo áudios, músicas e vozes nos projetos por meio da paleta Som.

### ANIMANDO ATORES

A Paleta de blocos Aparência possui 19 blocos. Neste tópico, será explorado apenas 13 desses blocos, os quais foram divididos em cinco categorias como vemos na Figura 41.

Figura 41 – Blocos da Paleta Aparência

Animação	
Pensamento e Fala	   
Visibilidade	   

<b>Tamanho</b>	adicione 10 ao tamanho mude o tamanho para 100 %
<b>Efeito</b>	apague os efeitos gráficos mude o efeito cor para 0

Fonte: elaborado pelos autores, 2017.

### Animando os movimentos

Neste tópico faremos com que os atores se movam pelo palco de forma animada.

A Figura 42 traz o *script* e o ator Gato com duas fantasias. Ao executar o *script*, temos a ilusão de o Gato estar andando.

**Figura 42 – Animação do ator Gato**



Fonte: elaborado pelos autores, 2016.

No *script* apresentado na Figura 42, ao clicar na bandeira verde será iniciado o laço do bloco **sempre**, em que o Gato com a fantasia1 andará 10 passos (bloco **mova 10 passos**), logo em seguida será dado um tempo de 0.3 segundos (bloco **espere 0.3 segundos**) e irá para a próxima fantasia (fantasia2) e assim sucessivamente. Quando o ator tocar na borda ele irá voltar (bloco **se tocar na borda volte**) e a sua rotação será apenas direita-esquerda (bloco **mude o estilo de rotação para**).

---

### Algo a mais 3...

Além de mudar a aparência de um ator alternando fantasias como vimos na Figura 41, também pode ser mudado o pano de fundo do palco, alterando a cena de uma história ou a

mudança de fase de um jogo, por exemplo, com os **blocos mude para o pano de fundo**, da paleta Aparência, e **quando o pano de fundo mudar para**, da paleta Eventos. Ver Figura 43.

**Figura 43 – Blocos que alteram o pano de fundo do Palco.**






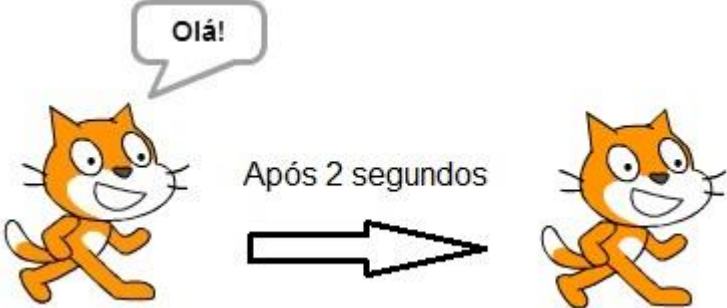
Fonte: elaborado pelos autores, 2017.

❓ Para mais informações acesse a **Janela de Dicas** na tela inicial do Scratch (Figura 1).

### **Pensamentos e falas**


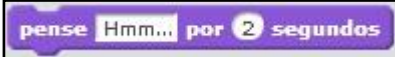

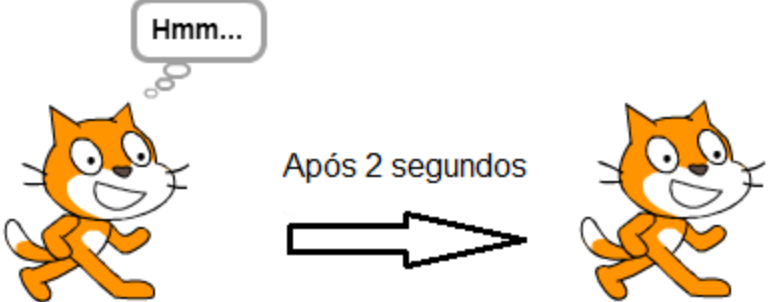
Alguns blocos adicionam falas e pensamentos ao ator. A Figura 44 e 45 mostram quais são esses blocos e como funcionam.

**Figura 44 – Blocos de fala.**

	
<b>Duração permanente</b>	<b>Duração por um tempo fixo</b>
	

Fonte: elaborado pelos atores, 2016.

Figura 45 – Blocos de pensamento.

	
Duração permanente	Duração por um tempo fixo
	

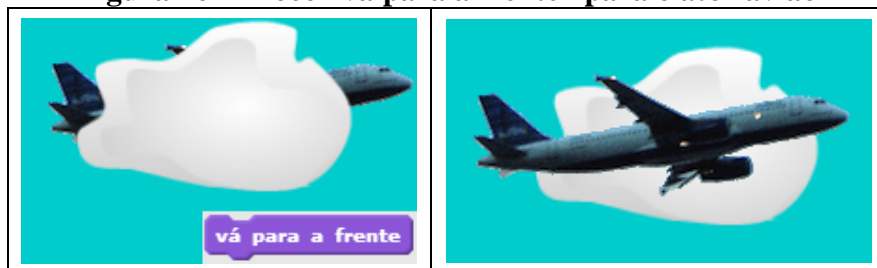
Fonte: elaborado pelos atores, 2016.

### Visibilidade

Alguns blocos alteram a visibilidade do ator, como os blocos **mostre**, **esconda**, **vá para a frente**, **vá camadas para trás**.

Os blocos **mostre** e **esconda** deixam o ator visível e oculto, assim como o próprio nome do bloco sugere. Já o bloco **vá para a frente**, determina que o ator ficará sobre um outro ator, conforme a Figura 46.

Figura 46 – Bloco “vá para a frente” para o ator avião

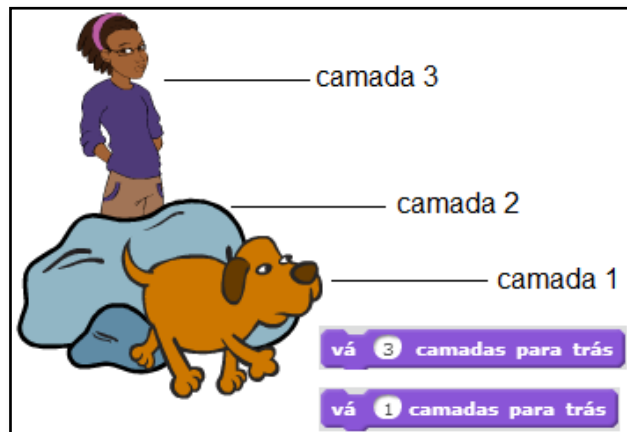


Fonte: elaborado pelos atores, 2016.

E o bloco **vá camadas para trás** envia o ator atrás de quantas camadas forem especificadas, como mostra a Figura 47, em que o ator Ana foi enviado três camadas para trás e o ator pedra uma camada.



**Figura 47 – Bloco vá camadas para trás aplicado nos atores pedra e Ana**



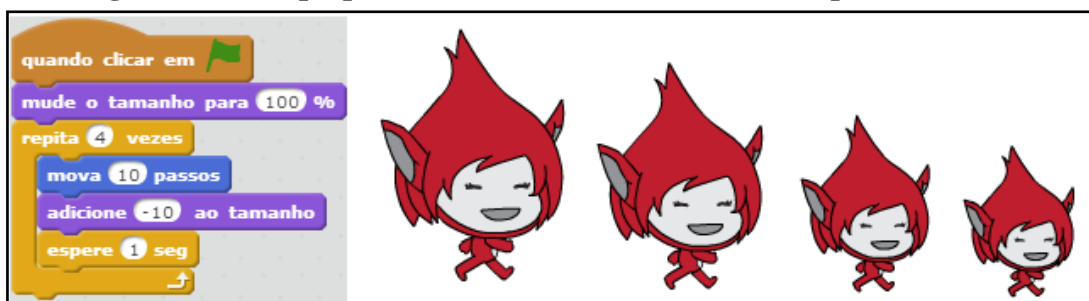
Fonte: elaborado pelos atores, 2016.

### Tamanho e Efeitos

O tamanho do ator pode ser alterado durante a execução de um *script* com os blocos **mude o tamanho para %** e **adicione ao tamanho**. No primeiro bloco, **mude o tamanho para %**, o tamanho do ator será alterado de acordo com um percentual do seu tamanho original. Já no segundo, **adicione ao tamanho**, o tamanho é modificado segundo uma quantidade definida do tamanho atual do ator.

A Figura 48 ilustra a alteração no tamanho do ator ET. Toda vez que a bandeira verde for clicada ele estará com 100% de seu tamanho original e conforme ele move 10 passos para a direita, o seu tamanho é reduzido em 10 do tamanho atual.

**Figura 48 – Script que altera o tamanho do ator ET enquanto se move**



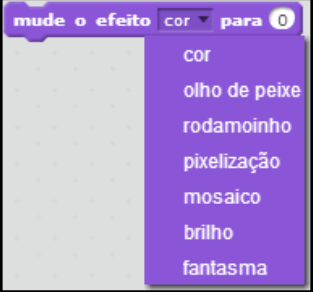


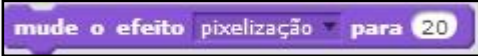


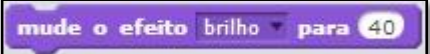




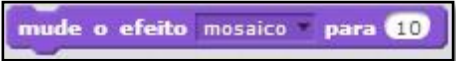

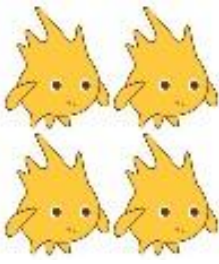
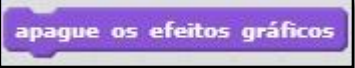

Fonte: elaborado pelos atores, 2016.

Também, podem ser adicionados efeitos de imagem às fantasias dos atores e aos panos de fundo com o bloco **mude o efeito para**.

Este bloco possui 7 efeitos: cor, olho de peixe, rodamoinho, pixelização, mosaico, brilho e fantasma. Eles são selecionados no menu do próprio bloco e a intensidade do efeito é determinada pelo número inserido.

Os efeitos adicionados também podem ser apagados com o bloco **apague os efeitos gráficos**. A Figura 49 mostra o que fazem esses efeitos gráficos.

**Figura 49 – Efeitos Gráficos do Scratch aplicados no ator**

Fonte: elaborado pelos atores, 2016.

## INTRODUZINDO SONS

Neste tópico, será explorada a Paleta de blocos Som. Esta contém 13 blocos, que serão apresentados em cinco categorias.

### **Blocos que reproduzem arquivos de áudio**

Contamos com três blocos que reproduzem arquivos de áudio no Scratch, conforme a Figura 50.

**Figura 50 – Blocos de reprodução de som**



Fonte: elaborado pelos atores, 2016.

São reproduzidos pelo Scratch arquivos de áudio no formato MP3 e WAV.

O bloco **toque o som até o fim** reproduz o som até o final, antes de mudar de comando, enquanto o bloco **toque o som** permite que seja executado o próximo comando antes do som terminar. E o bloco **pare todos os sons** para de reproduzir qualquer som imediatamente.

Os arquivos de áudio a serem reproduzidos pelos blocos **toque o som até o fim** e **toque o som** podem ser selecionados pelo menu suspenso do próprio bloco após ter sido selecionado na aba Sons ou pode ser feita uma gravação conforme mostra a Figura 51.

**Figura 51 – Menu suspenso do Bloco toque o som**



Fonte: elaborado pelos atores, 2016.

### **Blocos que reproduzem som de tambores e outros tipos de sons**

Os dois próximos blocos permitem criar efeitos sonoros, são eles: o bloco **toque o tambor por batidas** e o bloco **silêncio por batidas**. Ver Figura 52

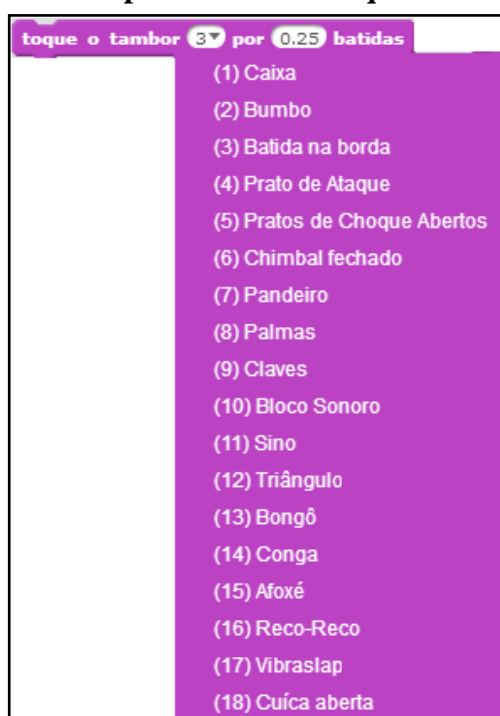
**Figura 52 – Blocos que criam efeitos sonoros.**



Fonte: elaborado pelos atores, 2016.

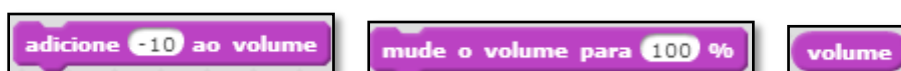
Com o bloco **silêncio por batidas** é possível adicionar pausas ao som. Já o bloco **toque o tambor por batidas** possui 18 tipos de sons de tambores que são selecionados por meio do menu suspenso do bloco conforme a Figura 53.

**Figura 53 – Menu suspenso do bloco toque o tambor por batidas**

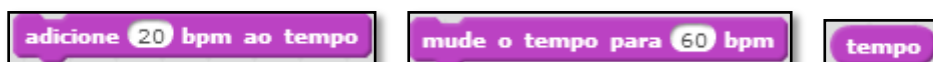


Fonte: elaborado pelos atores, 2016.

 **Blocos que controlam o volume do som**



 **Blocos que ajustam o tempo do som**



## CRIANDO ANIMAÇÕES COM OS BLOCOS DAS PALETAS APARÊNCIA E SOM

Usando ferramentas que foram abordadas até aqui, apresentamos uma animação em que um cubo altera a sua forma em um processo de planificação.

### Projeto: A dança do cubo

#### Preparando o Palco: Escolhendo o Pano de Fundo

A Figura 54 mostra o resultado final do pano de fundo do projeto, após as modificações que serão apresentadas a seguir.

**Figura 54 – Pano de fundo final**



Fonte: elaborado pelos atores, 2016.

Primeiramente, escolhemos o pano de fundo do palco. O tema escolhido é o *party room* da biblioteca do Scratch.

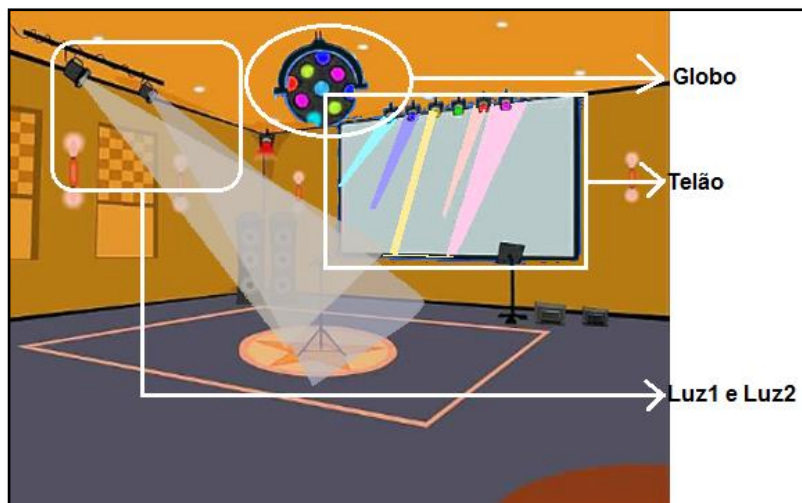
**Figura 55 – Pano de fundo *party room***



Fonte: elaborado pelos atores, 2016.

Analise as figuras 54 e 55, observe que há algumas diferenças entre ambas. Na figura 54 temos os atores **globo** e **telão** que se parecem como partes do pano de fundo original (Figura 55), além dos atores **luz1** e **luz2**. Ver Figura 56.

**Figura 56 – Pano de fundo**



Fonte: elaborado pelos atores, 2016.

Os atores **Globo** e **Telão** foram criados a partir do pano de fundo escolhido, pois desta forma é possível dar movimento ao pano de fundo de forma a deixar mais realista. E os atores **Luz1** e **Luz2** foram construídos no Paint Editor do Scratch.

Mas antes de mostrar como foram criados esses atores, terminaremos de preparar o palco.

Após escolhido o tema do pano de fundo, criaremos um *script* para ele, para que assim que a **bandeira verde** for clicada um som toque enquanto o programa é executado. O som foi selecionado a partir de um arquivo MP3 do computador do autor, mas poderia ser um da biblioteca do Scratch. Veja o *script* do palco na Figura 57.

**Figura 57 – Script do Pano de Fundo Party Rom**



Fonte: elaborado pelos atores, 2016.

## 🐱 Criando os atores

O projeto é composto de cinco atores:

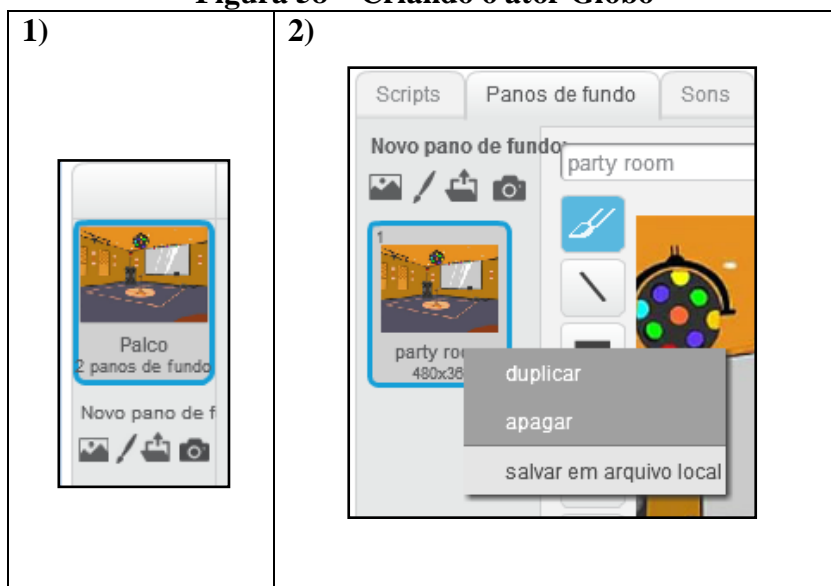
- **Globo**
- **Telão**
- **Luz1**
- **Luz2**
- **Cubo**

Todos os atores foram criados, nenhum deles pertence ao acervo do Scratch.

### 🐱 Criando o ator Globo

Selecione a miniatura do Palco e, em seguida, a aba Panos de Fundo. Na aba Panos de Fundo clique com o botão direito do *mouse* em cima da miniatura do pano de fundo *party room* e selecione **salvar em arquivo local** e salve em sua máquina. Ver Figura 58.

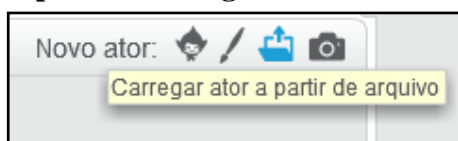
**Figura 58 – Criando o ator Globo**



Fonte: elaborado pelos atores, 2016.

Em seguida, na Janela de Atores, selecione **Carregar ator a partir de arquivo**, ver Figura 59, e selecione a imagem salva do pano de fundo.

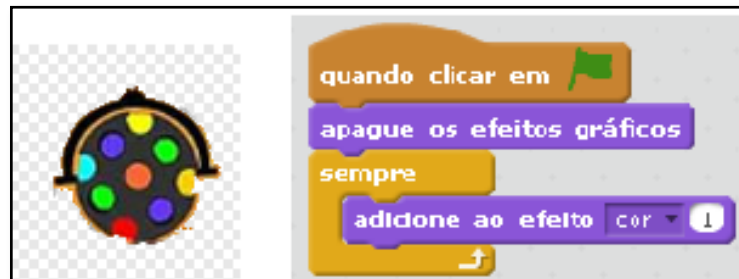
**Figura 59 – Importando imagem como ator de arquivo local**



Fonte: elaborado pelos atores, 2016.

Esse ator foi chamado de globo. Edite a sua fantasia no Paint Editor (use a cor transparente em torno do globo) e o posicione no palco de forma que fique exatamente em cima do desenho do globo do pano de fundo.

**Figura 60 – Ator Globo no Paint Editor e seu script**



Fonte: elaborado pelos atores, 2016.

A Figura 60 mostra como ficou a aparência final do ator globo após ser editado no Paint Editor e seu *script*. Durante a execução do programa, esse ator mudará de cor continuamente, criando a ilusão de que os círculos coloridos estão realmente mudando de cor, como se fossem luzes.

#### Criando o ator Telão

O ator Telão é criado da mesma forma que o ator Globo. Contudo, seu *script* possui um bloco a mais que o *script* do ator Globo, o bloco **vá camadas para trás**. Como o Telão pode ficar na frente do ator Cubo e dos atores Luz1 e Luz2, o enviaremos 3 camadas para trás, de forma que não sobreponha nenhum outro ator. Na Figura 61 pode ser visto o *script* e a aparência final do ator Telão.

**Figura 61 – Ator Telão no Paint Editor e seu script**



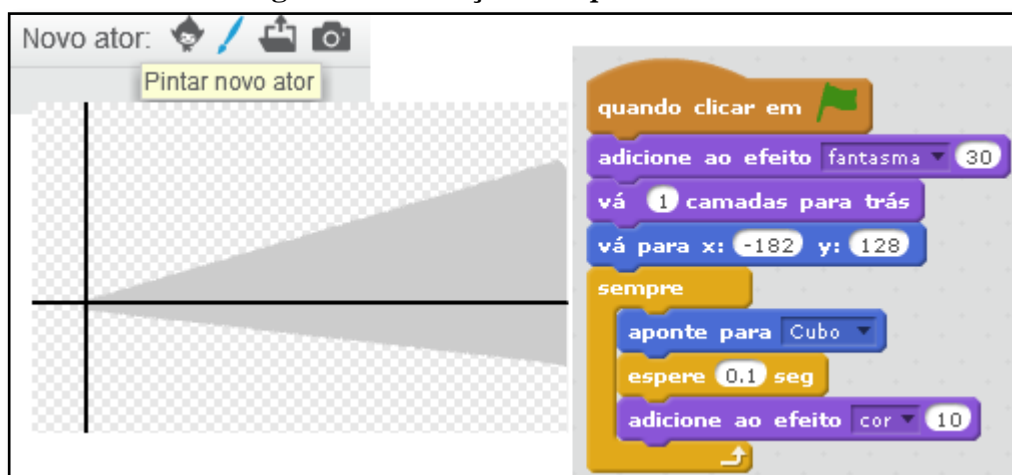
Fonte: elaborado pelos atores, 2016.



### Criando o ator Luz1 e Luz2

Os atores Luz1 e Luz2 foram criados no Paint Editor do Scratch a partir da Janela de Atores em formato de cone, com o centro na ponta da imagem. O seu *script* é criado de forma em que este ator represente um fecho de luz, deixando-o transparente (bloco **mude o efeito para**) e que siga (bloco **aponte para**) o ator Cubo enquanto este se movimenta. E para que ele não sobreponha o ator Cubo, o enviamos uma camada para trás com o bloco **vá camadas para trás**, conforme a Figura 62.

Figura 62 – Criação e *script* do ator Luz1



Fonte: elaborado pelos autores, 2016.

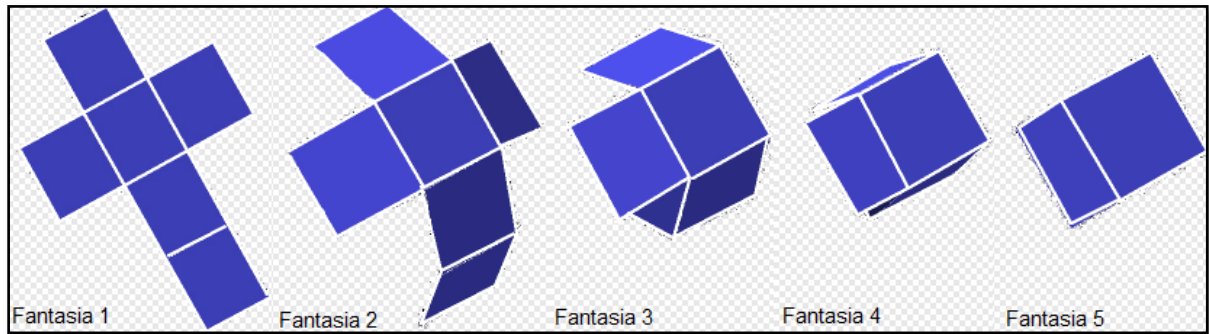
As coordenadas (x, y) do ator Luz1 e Luz2 dependerá da posição do ator no Palco. Verifique as coordenadas deles no canto inferior direito do Palco, posicionando o ponteiro do mouse no local desejado.

### Criando o ator Cubo

O ator Cubo foi criado a partir do software Poly<sup>2</sup>. Ele possui cinco fantasias, que vai desde ele totalmente fechado, até a sua total planificação, conforme a Figura 63.

<sup>2</sup> Disponível para download em <<http://www.peda.com/poly/>>.

**Figura 63 – Fantasias do ator Cubo**



Fonte: elaborado pelos atores, 2016.

Para que ele se movimente, de forma em que ele abra e feche, criamos o *script* apresentando na Figura 64.

**Figura 64 – Script do ator Cubo**



Fonte: elaborado pelos atores, 2016.

Após ter criado todos os atores e seus respectivos scripts, o projeto “A dança do cubo” estará pronto. Basta clicar na bandeira verde e ver o programa em execução, o qual deverá tocar uma música e os atores “globo”, “Telão”, “Luz1” e “Luz2” deverão mudar de cor como se as luzes do ambiente realmente estivessem em ação, enquanto o ator Cubo muda a sua forma.

### Exercitando 3

1. Crie um programa em que um homem de palito se movimenta pelo palco.
2. Com o pano de fundo **underwater3**, crie um programa utilizando diferentes efeitos gráficos de forma a efetuar a animação do aquário.

#### Sugestões:

🐾 Crie oito atores para este projeto:

- Peixe
  - Bolhas1
  - Bolhas2
  - Bolhas3
  - Alga1
  - Alga2
  - Alga3
  - Coral
- Use o efeito gráfico **rodamoinho** no Palco, de forma a dar uma aparência ondulante.
  - Mova o **Peixe** pelo aquário enquanto troca a sua fantasia.
  - Aplique o efeito **fantasma** ao ator **Alga**.
  - Use o efeito **cor** nos atores **Coral** e **Bolhas**.
3. Crie um projeto onde dois alunos conversam sobre a aula de geometria.

---

## Capítulo 4 | Blocos personalizados e Broadcasting

---

### INTRODUÇÃO

Será explorada nesse capítulo a criação de blocos personalizados da paleta Mais Blocos. Também, será abordado o Broadcasting de mensagem, que permite a coordenação entre diversos atores por meio de mensagens.

### BROADCASTING DE MENSAGENS

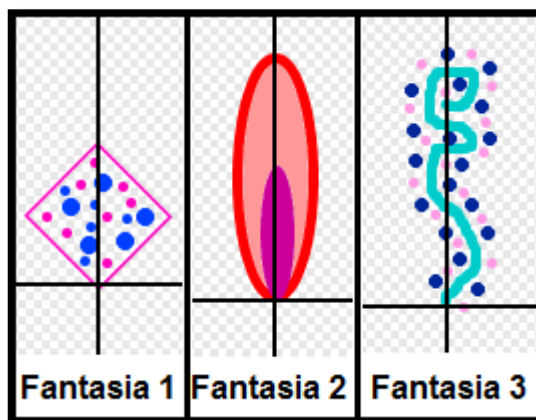
O Scratch contém três blocos de *broadcast*: **quando receber**, **envie a todos** e **envie a todos e espere**. Esses blocos coordenam o trabalho de vários atores ao mesmo tempo, fazendo que executem os seus *scripts* ao receberem as mensagens.

#### **Coordenando atores por meio do broadcasting de mensagens**

Criaremos um projeto em que um ator que possui três fantasias desenhará seguidamente três padrões geométricos utilizando broadcast de mensagens.

A Figura 65 mostra as fantasias do ator que compõem os padrões geométricos.

**Figura 65– Fantasias do ator que formam os padrões geométricos**

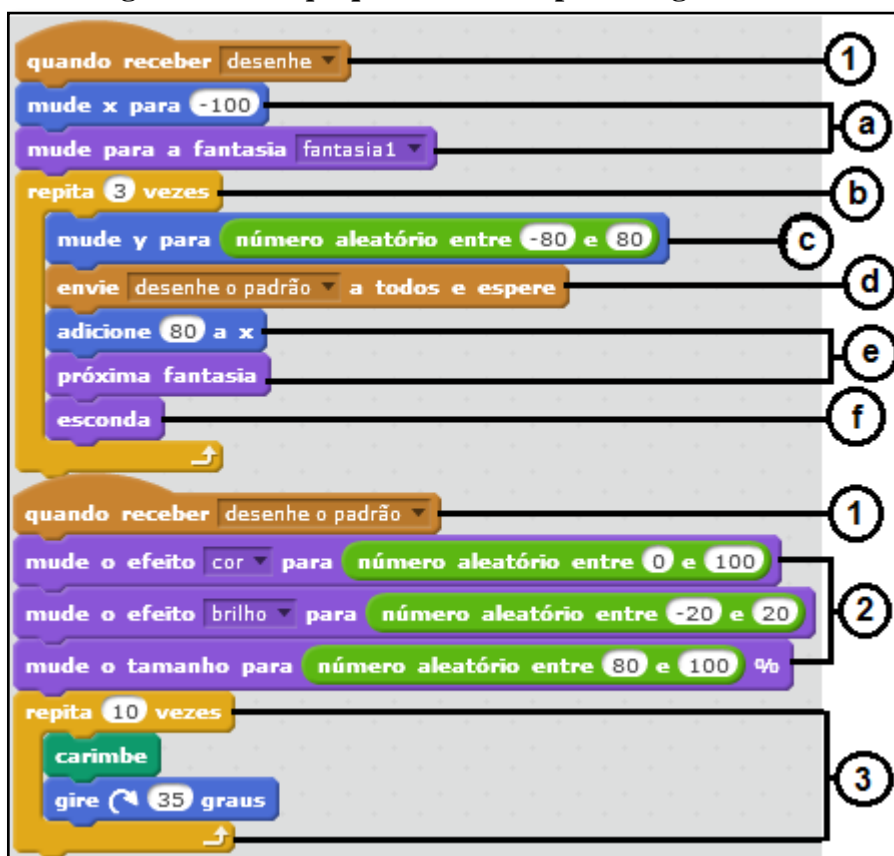


Fonte: elaborado pelos autores, 2017.

Esse projeto possui dois *scripts*, um que desenha os três padrões geométricos e outro que determina o desenho do padrão geométrico. O ator iniciará o *script* assim que receber a

mensagem **desenhe** e **desenhe o padrão** (1). Assim, é alterado o efeito da cor, o efeito do brilho e o tamanho da fantasia aleatoriamente (2), e são carimbadas 10 cópias a cada giro da fantasia (3). O comando descrito em (a) determina a posição x da fantasia e qual usar para desenhar o primeiro padrão. O laço (b) determina a quantidade de padrões que serão desenhados, definindo aleatoriamente a posição y do ator, chamando por meio do *broadcast* de mensagem (d) o padrão a ser desenhado. Feito um padrão, o ator prepara-se para desenhar o próximo padrão geométrico (e) e ao finalizar o laço esconde o ator (f), conforme a Figura 66.

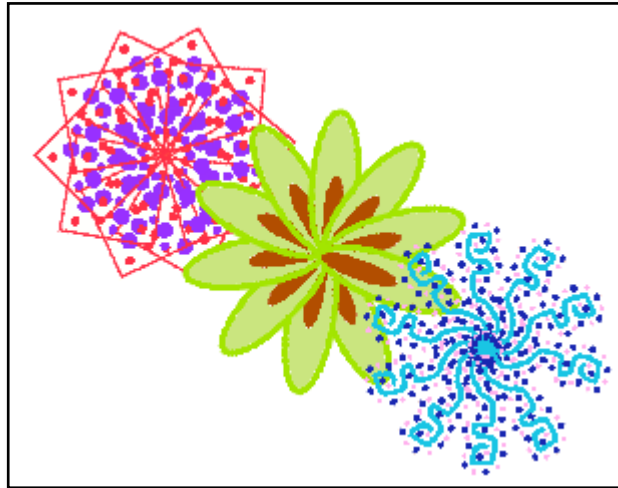
**Figura 66 – Script que desenha os padrões geométricos**



Fonte: elaborado pelos autores, 2017.

A Figura 67 mostra alguns dos padrões geométricos formados pelo *script* apresentado na Figura 66 e suas três fantasias criadas mostradas na Figura 65.

**Figura 67 – Alguns padrões geométricos formados a partir das 3 fantasias**



Fonte: elaborado pelos autores, 2017.

Esse exemplo demonstrou como o *broadcasting* de mensagem funciona.

Os blocos de *broadcast* “**envie a todos e espere**” e “**envie a todos**” funcionam de modo semelhante. Enquanto o bloco “**envie a todos e espere**” espera até que todos os atores que receberão a mensagem tenham terminado de executar seu comando, o bloco “**envie a todos**” envia a mensagem para todos os atores, inclusive para si mesmo.

## **BLOCOS PERSONALIZADOS**

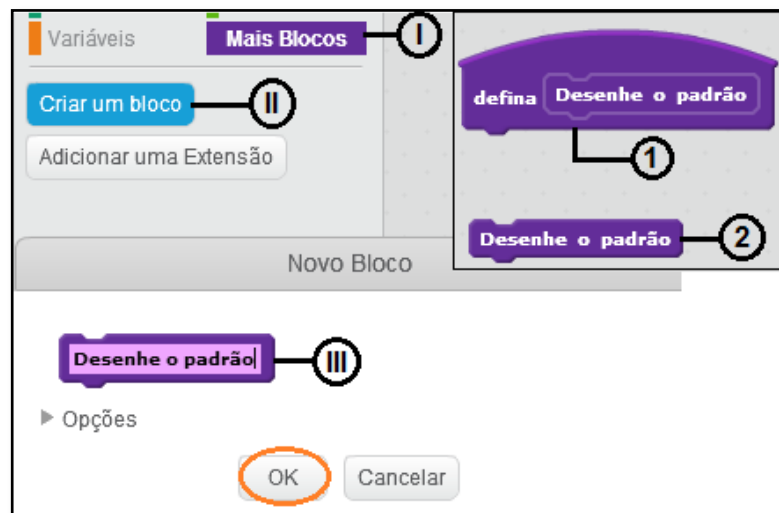
Com a paleta **Mais Blocos** é possível criar blocos personalizados, os quais podem ser usados como os demais blocos da Paleta de Blocos.

A seguir, faremos alterações no programa que cria padrões geométricos da seção anterior para melhor explorar a paleta **Mais Blocos**.

### **🐜 Coordenando atores por meio dos blocos personalizados**

Na paleta **Mais Blocos** podemos criar um bloco personalizado que desenha os três padrões geométricos da seção anterior. Para isso, clicamos em **Mais Blocos** (I) e, em seguida, **criar um bloco** (II), nomeando-o (III). Ver Figura 68.

Figura 68 – Criando o bloco Desenhe o padrão

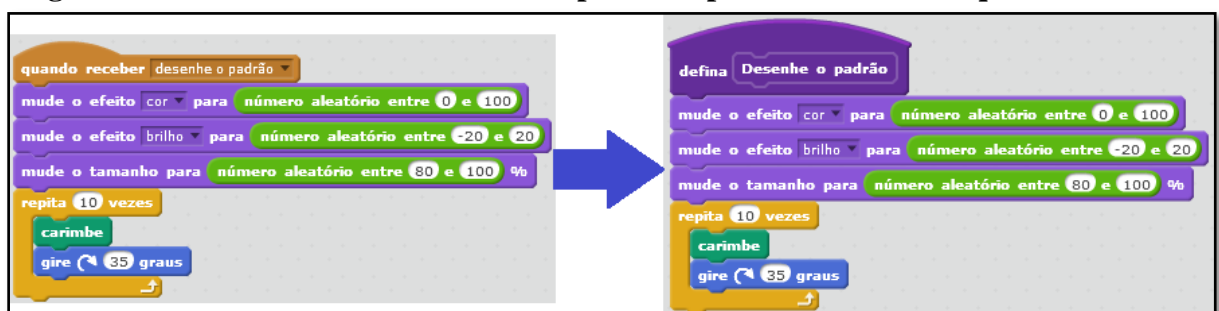


Fonte: elaborado pelos autores, 2017.

Ao clicar em **ok**, o novo bloco está pronto e com ele, dois blocos são criados automaticamente, um que aparece na paleta **Mais Blocos** (2) e outro na **Janela de scripts** (1), como mostrado na Figura 68.

Para que o novo bloco funcione precisamos definir como deve proceder por meio de comandos. Para este caso, temos que dar os comandos ao bloco **Desenhe o padrão** para que desenhe o padrão geométrico. A 69 mostra o *script* que determina a ação do bloco “**Defina Desenhe o padrão**”. No *script* da Figura 66, usava-se um comando de *broadcasting* de mensagem com o bloco “**quando receber**” (imagem à esquerda da Figura 69) e o mesmo comando desse *script* foi usado para determinar o bloco **Desenhe o padrão** (imagem à direita da Figura 69).

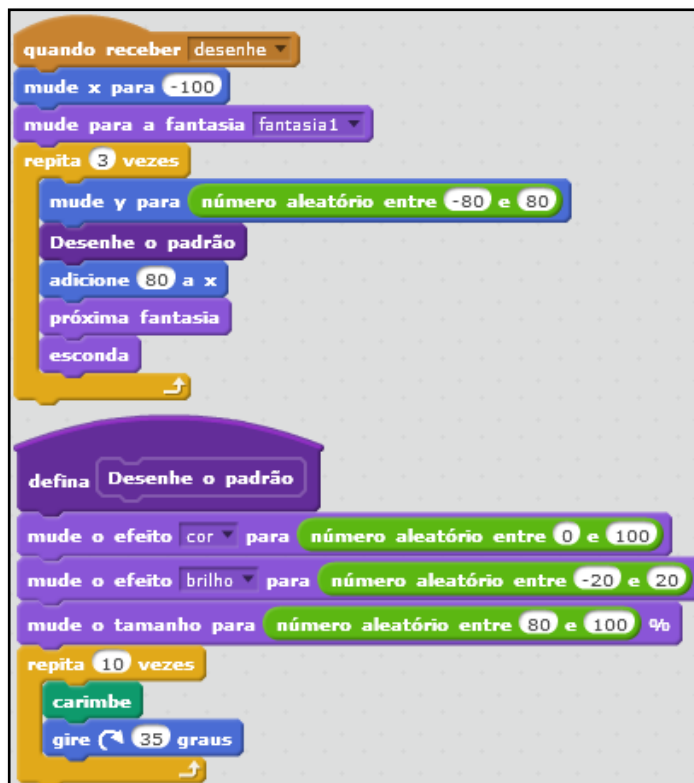
Figura 69 – Definindo o bloco Desenhe o padrão a partir do comando quando receber



Fonte: elaborado pelos autores, 2017.

Determinado o bloco **Desenhe o padrão**, vamos substituí-lo pelo bloco **envie a todos e espere**, Figura 66 – (d). O novo *script* para o programa que desenha os três padrões geométricos está descrito na Figura 70.

Figura 70 – *Script* usando o bloco Desenhe o padrão



Fonte: elaborado pelos autores, 2017.

A paleta **Mais Blocos**, permite criar blocos personalizados que podem facilitar a criação, testagem e depuração de *scripts* de programas mais complexos.

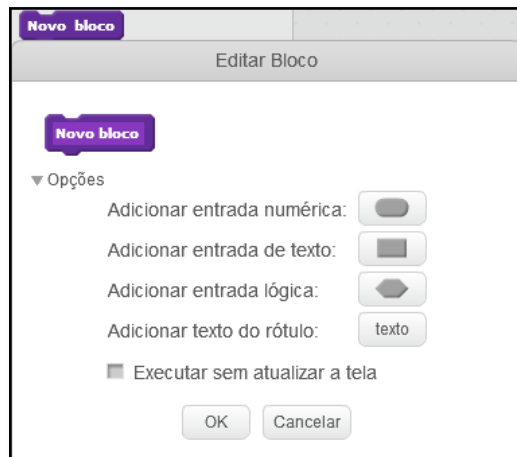
---

#### Algo a mais 4...

Podemos criar blocos personalizados que possuem entradas reservadas para a alteração de parâmetros, como o bloco **mova**, por exemplo. Para isso, basta clicar com o botão direito do *mouse* em cima do bloco criado e ir em editar. A caixa mostrada na Figura 71 abrirá, então, selecione o menu opções.



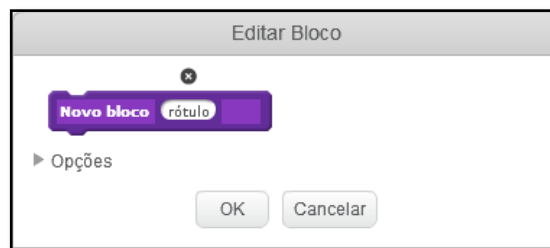
**Figura 71 – Adicionando entrada numérica a um novo bloco**



Fonte: elaborado pelos autores, 2017.

Ao selecionar essa opção, uma entrada aparecerá no bloco **Defina “Novo Bloco”** e um rótulo denominado **number1**. De forma que o rótulo dessa entrada seja mais significativo, renomeie para algo que melhor descreva essa entrada. Ver Figura 72.

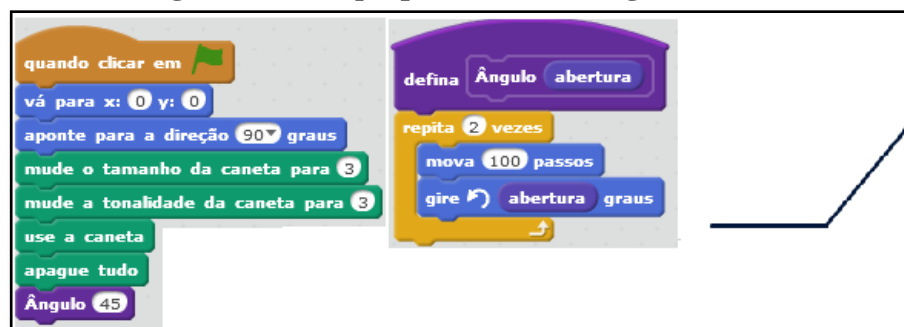
**Figura 72 – Alterando o rótulo number1**



Fonte: elaborado pelos autores, 2017.

### Exemplo de aplicação

**Figura 73 – Script que desenha o ângulo externo**

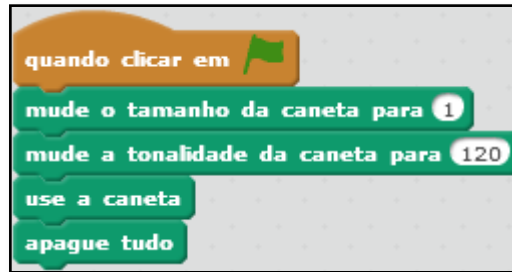


Fonte: elaborado pelos autores, 2017.

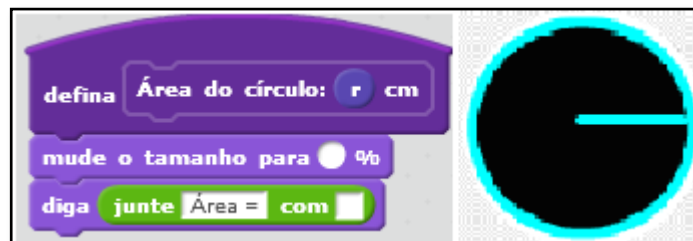
## Exercitando 4

1. Complete o *script* criando um bloco personalizado da paleta Mais Blocos que desenhe um quadrado de forma que o tamanho do lado pode ser alterado.

**Dica:** crie um bloco denominado **quadrado** e complete o *script* da figura.



2. Crie um bloco que calcule a área da circunferência dado o seu raio e que a fantasia do ator mude de tamanho de acordo com o raio, conforme a figura a seguir. Utilize  $\pi = 3,14$ .



3. Desenhe uma flor criando três blocos personalizados. Um que desenhe a pétala da flor. Um que desenha a flor a partir do bloco criado para a pétala e um que desenhe toda a flor, contendo o caule com uma folha (bloco que desenha uma pétala) e a flor utilizando ambos blocos criados anteriormente.

**Dica:** Use a definição a seguir para o bloco que desenha a pétala da flor.



---

## Capítulo 5 | As Variáveis no Scratch

---

### INTRODUÇÃO

Será explorada neste capítulo a paleta de blocos Variáveis.

### A PALETA DE BLOCOS VARIÁVEIS - VARIÁVEIS

O Scratch suporta três tipos de dados: os valores **booleanos**, que são os *verdadeiros* e *falsos*, **números**, que podem ser *inteiros* ou *decimais* e **strings**, que armazenam *palavras*, *números* e *símbolos*, como por exemplo +, -, @.

Uma **variável** é um espaço na memória do computador que possui um nome e armazena dados. O projeto *Lançamento de Dados* ilustra como as variáveis funcionam.

#### 🐿 Criando e usando variáveis: Lançamento de Dados

Esse projeto possui três atores, dois dados e o de uma menina. É um simulador de lançamento de dados, em que a menina fala o resultado da soma dos valores obtidos em cada dado. Para isso, é necessário construir dois blocos de variáveis, **valor1** e **valor2**, pois serão esses blocos que armazenarão o valor dos respectivos dados.

Figura 74 – Criando os blocos de variáveis valor1 e valor2



Fonte: elaborado pelos autores, 2017.

Conforme mostrado na Figura 74, assim que os blocos **valor1** e **valor2** (1) são criados, outros quatro blocos relacionados a eles também aparecem na paleta Variáveis. Esses blocos (1), além de serem o nome da variável, pela caixa de seleção, seu monitor aparece ou não no Palco. O bloco **mude para** (2) permite definir o valor da variável, enquanto o bloco **adicione a** (3) permite alterar o valor da variável em um valor fixo. Já os blocos **mostre/ esconda a variável** (4) permitem mostrar/ esconder o monitor da variável no Palco enquanto o *script* estiver sendo executado.

---

### Algo a mais 5...

Quando criamos uma variável precisamos definir seu *escopo*. Para isso, basta selecionar a opção **Para todos os atores** ou **Para este ator apenas** na caixa **Nova Variável**.

---

#### **Script do ator menina**

O ator da menina, **Avery**, foi escolhido da biblioteca do Scratch. Este ator possui duas fantasias, como mostra a Figura 75.

**Figura 75 – As fantasias do ator menina (Avery)**



Fonte: elaborado pelos autores, 2017.

Com as variáveis necessárias criadas é possível montar o *script* do ator **Avery**, conforme a Figura 76.

Figura 76 – Script do ator Avery



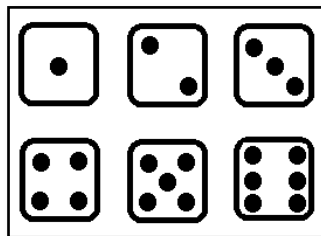
Fonte: elaborado pelos autores, 2017.

Quando clicar na bandeira verde, será atribuído um valor aleatório para as variáveis **valor1** e **valor2** entre 1 e 6. Em seguida, o *script* envia o *broadcast* de mensagem **role** para os atores dos dados (dado1 e dado2) notificando-os para que mudem suas fantasias de acordo com os valores correspondentes das variáveis **valor1** e **valor2**. Assim que os atores **dado1** e **dado2** executarem seus comandos, o *script* será retomado e mostrará o resultado da soma dos números apresentados nas faces dos dados, modificando também a fantasia do ator **Avery** de forma que dê a ilusão de estar falando.

#### Script dos atores dado1 e dado2

Os atores **dado1** e **dado2** possuem seis fantasias cada um, como mostra a Figura 77. Tais fantasias foram criadas em um software de edição.

Figura 77 – Fantasias dos atores dado1 e dado2



Fonte: elaborado pelos autores, 2017.

Como o ator **Avery** enviou o *broadcast* de mensagem **role** para os atores **dado1** e **dado2**, os *scripts* deles precisam iniciar com o bloco que irá receber essa mensagem (**quando receber**) de forma a executar o comando. A mensagem enviada para estes atores era para que

mudassem a fantasia para o número armazenado nas variáveis. O valor armazenado em **valor1** corresponde aos valores do ator **dado1** e o valor armazenado na variável **valor2** corresponde ao ator **dado2**.

Para que se tenha a impressão dos dados estarem rolando, as fantasias são alternadas aleatoriamente entre 1 e 6 em um laço. E por fim, permanecerão as fantasias que apresentam os números armazenados nas variáveis. A Figura 78 mostra o *script* dos atores **dado1** e **dado2**.

Figura 78 – *Script* dos atores **dado1** e **dado2**

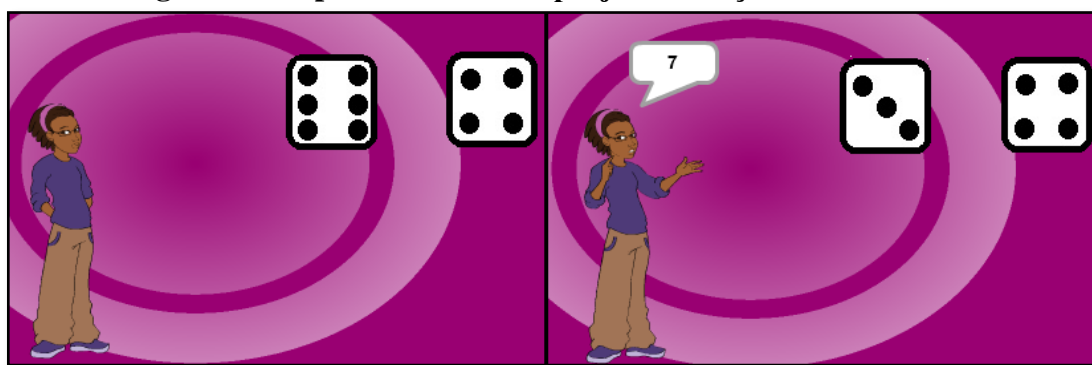


Fonte: elaborado pelos autores, 2017.

Note que os *scripts* de **dado1** e **dado2** são semelhantes, diferenciando-se apenas na variável. Pois, uma vez que **dado1** altera sua fantasia para o número armazenado na variável **valor1**, o ator **dado2** fixa sua fantasia para o número armazenado em **valor2**.

Criados ambos os *scripts* para os dados, clique na bandeira verde para testar. A Figura 79 mostra como fica a aparência final do projeto.

Figura 79 – Aparência final do projeto “Lançamento de dados”



Fonte: elaborado pelos autores, 2017.

A imagem da esquerda mostra como fica o projeto quando a bandeira verde é clicada. Após 2 segundos o ator **Avery** volta à posição inicial, representada pela imagem da direita.

Por meio do projeto “Lançamento de Dados” podemos explorar como funciona o uso dos blocos de variáveis no Scratch.

## MONITORES DE VARIÁVEIS

Os monitores de variáveis podem ser exibidos no Palco por meio da caixa de seleção, conforme a Figura 80.

**Figura 80 – Monitor da variável valor1 exibido no Placo**



Fonte: elaborado pelos autores, 2017.

No monitor é exibido o valor corrente da variável.

Eles podem servir como leitores, como apresentado na Figura 80, ou também como controles, que exibem e permitem alterar o conteúdo da variável por meio de um **controle deslizante**, como pode ser visto na Figura 81.

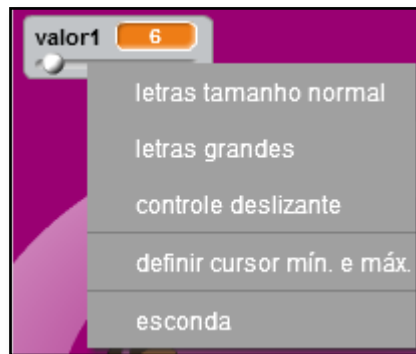
**Figura 81 – Monitor da variável como controle deslizante**



Fonte: elaborado pelos autores, 2017.

O intervalo do controle deslizante pode ser delimitado em um valor máximo e mínimo. Para isso, basta clicar com o botão direito do mouse em cima do monitor e selecionar a opção **definir cursor mín. e máx.** como mostra a Figura 82.

**Figura 82 – Definindo o intervalo do controle deslizante**



Fonte: elaborado pelos autores, 2017.

Com um clique duplo sobre o monitor, a sua forma pode ser alterada para normal, grande ou controle deslizante, conforme a Figura 83.

**Figura 83 – Modos de exibição do monitor**



Fonte: elaborado pelos autores, 2017.

### Usando monitores de variáveis: Volume e área da superfície da esfera

O projeto “Volume e área da superfície da esfera” ilustra como os monitores podem ser utilizados nas criações. Este projeto calcula a área da superfície da esfera e o seu volume.

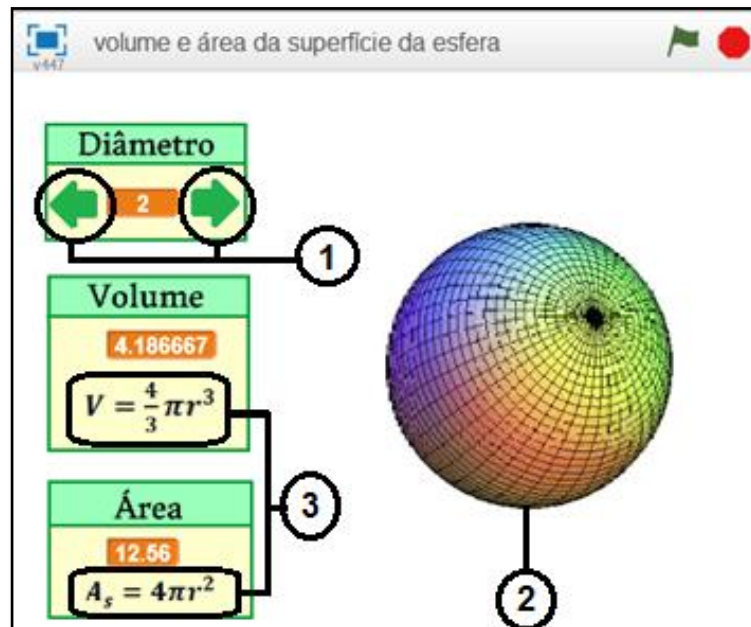
O programa contém três atores, a **esfera**, uma seta indicando para a esquerda e uma seta indicando para a direita, as quais foram denominadas de **menos** e **mais**, respectivamente.

A Figura 84 mostra como ficará o Palco do projeto. Os atores das setas (1) foram construídos em um *software* de edição, o ator **esfera**<sup>3</sup> (2) foi retirado do banco de dados da *internet* e o pano de fundo foi construído no *Paint Editor* do Scratch. As equações (3) do volume da esfera e a área de sua superfície foram escritas em um *software* de edição de equações matemáticas, salvas como imagem e importadas para a área de edição do Palco na aba Panos de fundo. Os monitores que apresentam os valores para o Diâmetro, Volume e Área, são exibidos por meio da paleta Variáveis.

<sup>3</sup> Disponível em: <<http://enciclopedia.us.es/index.php/Esfera>>. Acesso em: 05 jul. 2016.



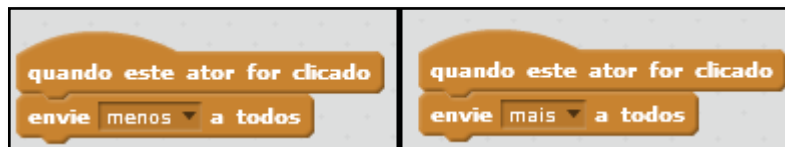
Figura 84 – Palco do projeto Volume e área da superfície da esfera



Fonte: elaborado pelos autores, 2017.

Os *scripts* dos atores **mais** e **menos**, representados pelas setas, enviam um *broadcast* de mensagem para informar que foram clicados, como mostra a Figura 85.

Figura 85 – *Script* dos atores “menos” e “mais”, respectivamente



Fonte: elaborado pelos autores, 2017.

Quando o ator **esfera** receber as mensagens dos atores **mais** e **menos**, seu *script* irá executar os comandos apresentados na Figura 86. Conforme os atores **mais** e **menos** são clicados, a **esfera** altera seu tamanho correspondendo aos valores do diâmetro iguais a “1”, “1,5”, “2”, “2,5” e “3”.

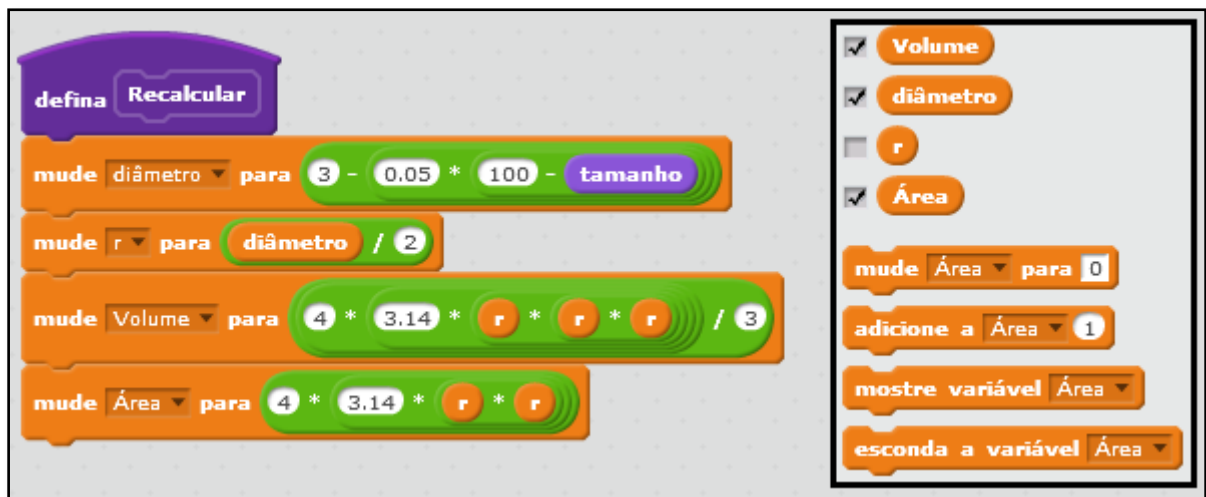
Figura 86 – Script da esfera



Fonte: elaborado pelos autores, 2017.

Assim que a **esfera** muda de tamanho, começa a executar o comando do bloco personalizado **Recalcular**, para que os cálculos da área da superfície e do volume sejam atualizados. A Figura 87 mostra como esse bloco foi construído.

Figura 87 – Bloco personalizado Recalcular da paleta Mais Blocos



Fonte: elaborado pelos autores, 2017.

Para a construção do bloco **Recalcular** foi necessário a construção das variáveis **Volume**, **Diâmetro**, **r** (raio) e **Área**, pois são estas variáveis que compõem tal bloco, e também, os monitores são exibidos por meio delas, conforme visto na Figura 84.

Inicialmente, a variável **diâmetro** tem seu valor definido pela fórmula:

$$\text{diâmetro} = 3 - 0,05 \cdot (100 - \text{tamanho})$$

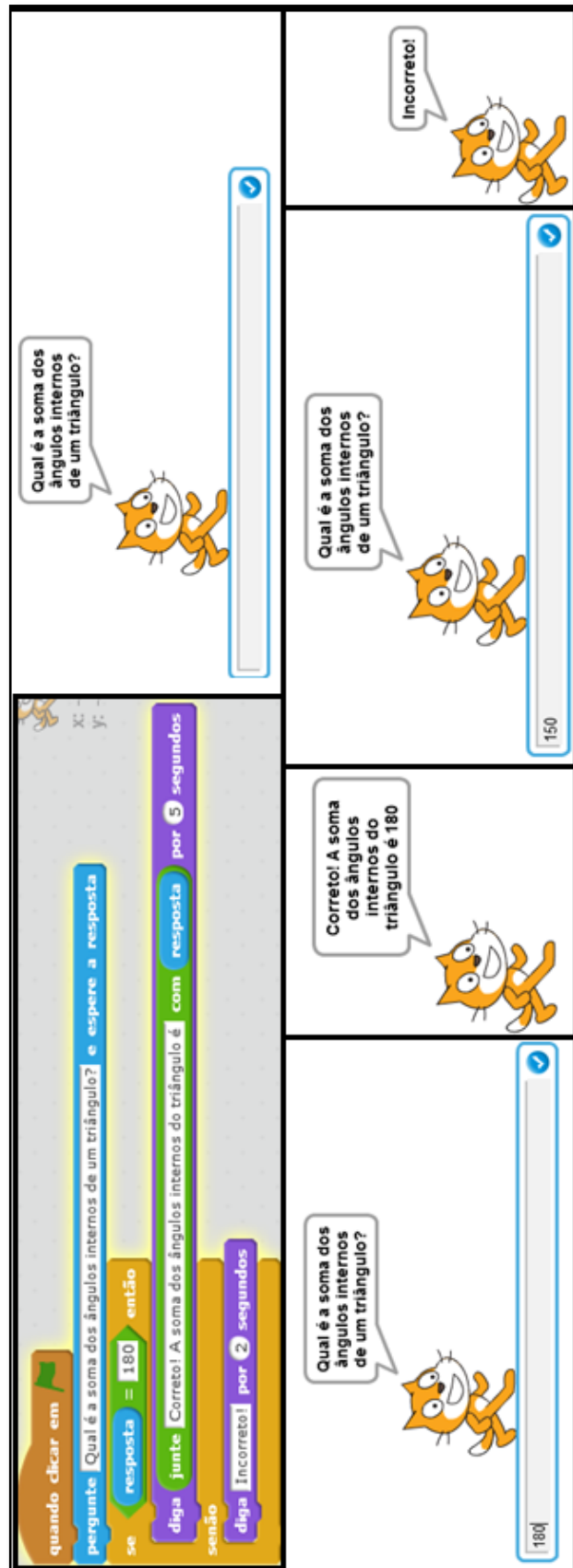
Logo, obteremos para o diâmetro os valores 1, 1,5, 2, 2,5 e 3, como dito anteriormente. Em seguida, o diâmetro é dividido por 2, definindo o valor de **r** (raio). Então,

os valores do volume e da área são calculados, e assim, os valores aparecem automaticamente nos monitores correspondentes que estão visíveis no Palco.

## **INSERÇÃO DE DADOS DE ENTRADA PELOS USUÁRIOS**

Para criarmos projetos que requerem dos usuários a inserção de respostas a perguntas, contamos com o bloco **pergunte e espere a resposta** da paleta Sensores. Veja na Figura 88 um exemplo de aplicação em que o ator Gato pergunta qual é a soma dos ângulos internos de um triângulo, espera a resposta do usuário e diz ao mesmo que está correto, caso a resposta seja 180 graus, ou incorreto para outro valor.

Figura 88 – Script usando o bloco pergunte e espere a resposta



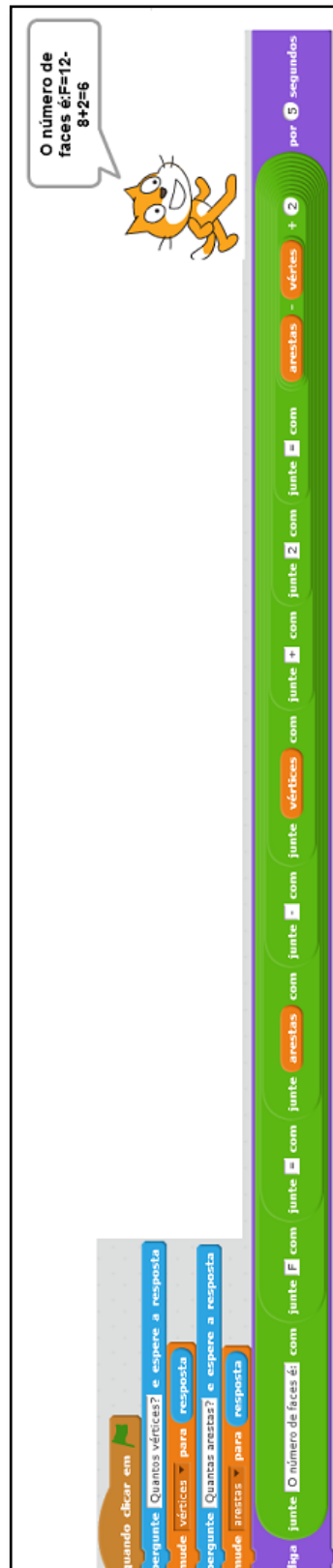
Fonte: elaborado pelos autores, 2017.

Mas, também, podemos utilizar variáveis para armazenar valores inseridos pelo usuário, de forma a criar todo o tipo de *strings* e exibir mensagens personalizadas. O projeto “Relação de Euler” ilustra como funciona essa técnica.

### **Relação de Euler**

O *script* da Figura 89 solicita dois números ao usuário, o número de vértices e o número de arestas de um poliedro. Então, a equação  $F = A - V + 2$  é calculada e o número de faces é exibido por meio do comando “**diga por segundos**”.

Figura 89 – Script que determina o número de faces de um poliedro pela Relação de Euler



Fonte: elaborado pelos autores, 2017.

O programa usa duas variáveis para salvar os valores inseridos pelo usuário, a variável **vértices** e a variável **arestas**. Então, concatenados com o bloco **junte com**, forma a composição da fala final exibida no balão.

A técnica de usar o bloco **junte com** aninhados com os blocos de variáveis para armazenar valores inseridos pelo usuário possibilita a criação de todo tipo de mensagens personalizadas em nossos projetos.

---

### Exercitando 5

1. Crie um programa que calcule a velocidade média em  $\text{km/h}$  com os valores da distância e do tempo inseridos pelo usuário. Mostre a velocidade média, juntamente com uma mensagem apropriada ao usuário.

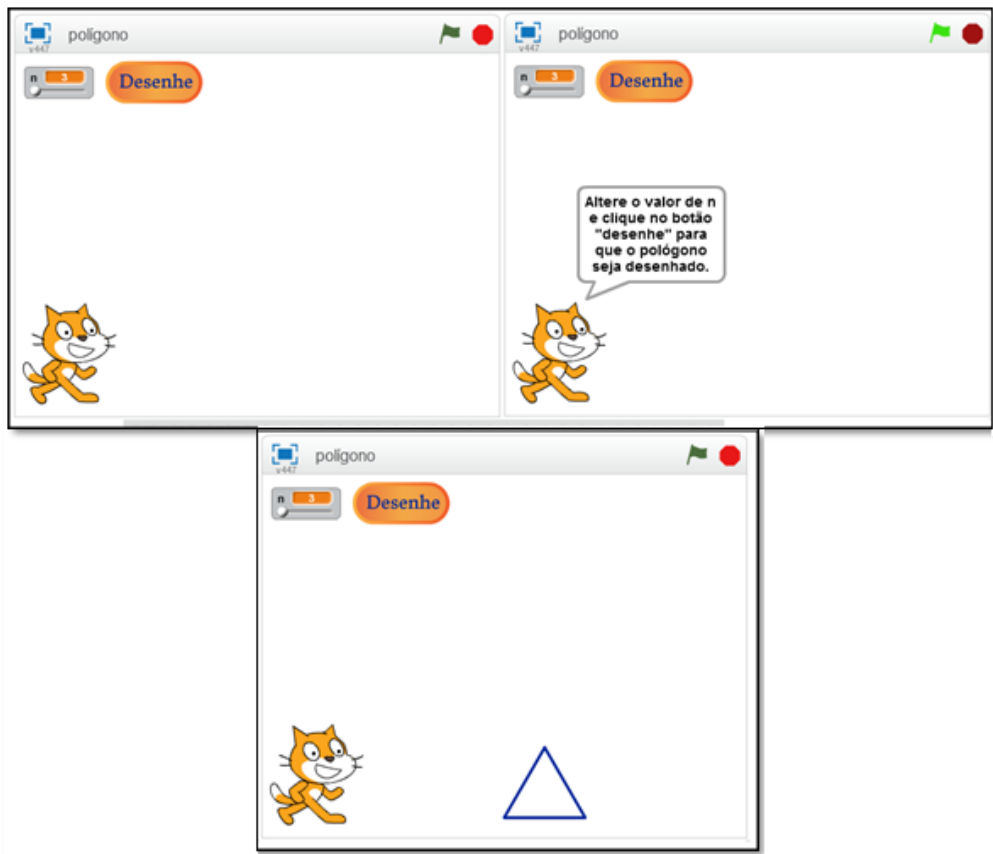
2. Crie um programa que leia os tamanhos dos dois catetos de um triângulo retângulo inseridos pelo usuário e calcule o tamanho da hipotenusa.

3. Crie um programa que desenhe um polígono com **n** lados quando um botão for clicado. Utilize o monitor como um controle deslizante para alterar o número **n** de lados.

O programa deve conter as seguintes instruções:

- 👤 Três atores, um que dá as instruções iniciais, outro ator oculto que desenha os polígonos e um ator botão.
- 👤 Um ator que dê instrução: **Altere o valor de n e clique no botão “desenhe” para que o polígono seja desenhado.**
- 👤 Crie uma variável **n**.
- 👤 Crie um bloco personalizado que desenhe um polígono dado o número de lados **n**.

**Dica:** A figura abaixo é um exemplo de como pode ficar o projeto.





---

## Capítulo 6 | As Listas no Scratch

---

### INTRODUÇÃO

No capítulo anterior falamos a respeito da criação de **variáveis**. Neste capítulo continuaremos abordando a paleta de blocos Variáveis, porém será explorada a criação de **listas**.

### A PALETA DE BLOCOS VARIÁVEIS – LISTAS

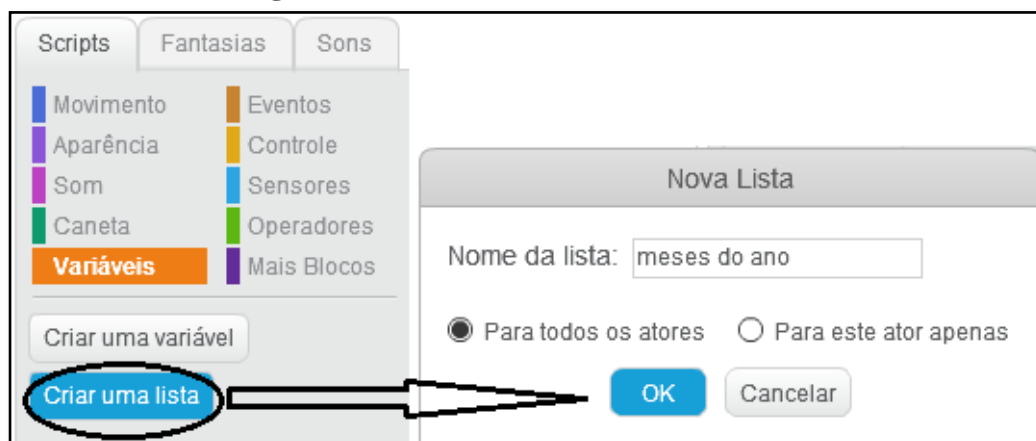
As **listas** são úteis para o armazenamento de um conjunto de valores, como números de telefone, títulos de livros, lista de compras, dentre outros. Isso não é possível nas **variáveis**, uma vez que armazenam apenas um valor por vez.

Os itens de uma **lista** podem ser acionados por meio de sua posição, por exemplo, o primeiro item tem índice igual a 1, o segundo igual a 2 e assim por diante.

#### Criando e usando listas

Criar uma **lista** é bem semelhante à criação de uma **variável**. Criaremos uma **lista** que armazena os doze meses do ano. Para isso, na paleta de blocos *Variáveis* selecione **criar uma lista**, aparecerá a caixa de diálogo *Nova Lista*, apresentada na Figura 90 à direita, que devemos nomeá-la e definir o seu escopo - **Para todos os atores** ou **Para este ator apenas**.

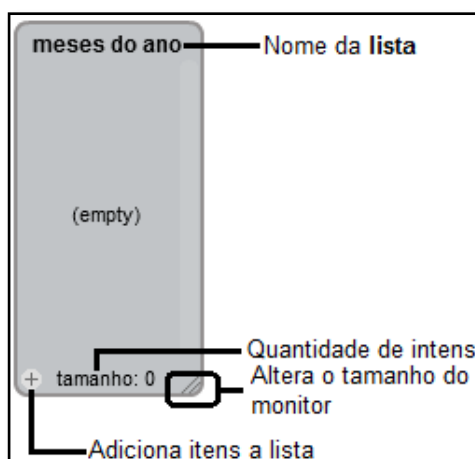
**Figura 90 – Criando a lista “meses do ano”**



Fonte: elaborado pelos autores, 2017.

Ao clicar em **OK**, é criada uma **lista vazia** e o monitor da lista aparece no Palco. Veja a Figura 91.

**Figura 91 – Monitor de uma lista ao ser criada exibida no Palco**



Fonte: elaborado pelos autores, 2017.

Como se deseja criar uma lista que contenha os meses do ano, vamos clicar doze vezes no sinal de + no canto inferior esquerdo do monitor, conforme foi mostrado na Figura 91, para que doze entradas sejam criadas. Em seguida, basta inserir os meses do ano (ver Figura 92).

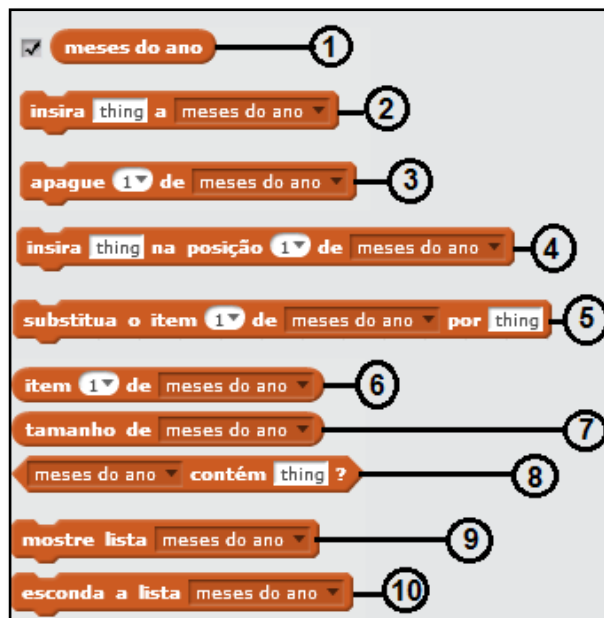
**Figura 92 – Acrescentando itens na lista**



Fonte: elaborado pelos autores, 2017.

Também, aparecerão na “paleta de blocos”, dez blocos que podem ser usados com as **listas**, conforme a Figura 93.

**Figura 93 – Blocos para serem usados com as listas**



Fonte: elaborado pelos autores, 2017.

Esses comandos podem ser usados para a manipulação do conteúdo da **lista** enquanto o *script* é executado. A seção seguinte explorará mais detalhadamente cada um desses blocos.

### **Comandos de lista**

A Figura 93 apresentou todos os blocos que são adicionados pelo Scratch quando uma lista é criada.

1. **Nome da lista:** Ativando a caixa de seleção, o monitor da lista é exibido ou não no Palco.
2. **“Inserir” e “apagar”:** O bloco **insira**, insere um novo item no final de uma lista, enquanto o bloco **apague** remove um item de uma posição específica. A Figura 94 mostra como esses comandos funcionam.

**Figura 94 – Blocos insira e apague em execução**



Fonte: elaborado pelos autores, 2017.

Observe que primeiro é excluído o nome “Andrea” da “lista Amigos” (item 3) e, em seguida, adicionado o nome “Day” no final da lista (item 4).

O bloco **apague** possui um menu suspenso, conforme mostra a Figura 95, em que o número correspondente a **1** apaga o primeiro item da lista (no nosso exemplo “Ana”), **último** apaga o último item da lista (no nosso exemplo “Day”, considerando a lista da direita na Figura 94) e **todos** apaga todos os itens da lista.

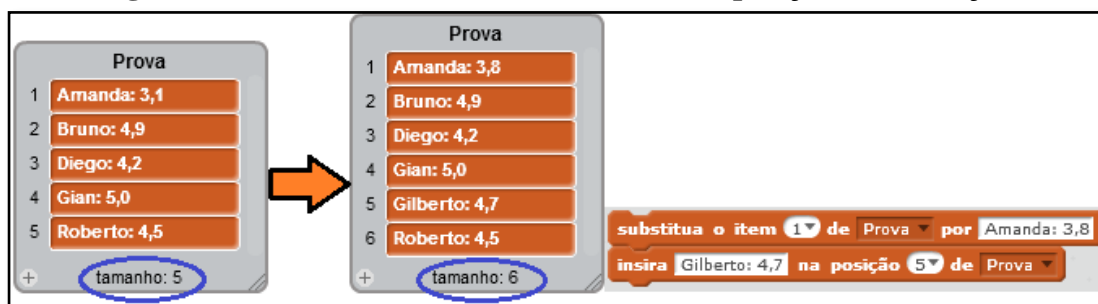
**Figura 95 – Menu suspenso do bloco apague**



Fonte: elaborado pelos autores, 2017.

3. **“Insira na posição” e “substitua o item”:** Esses blocos permitem inserir um item na lista na posição desejada e alterar um dado dela. A Figura 96 ilustra uma lista contendo o nome dos alunos e a nota da prova de matemática.

Figura 96 – Blocos substitua o item e insira na posição em execução



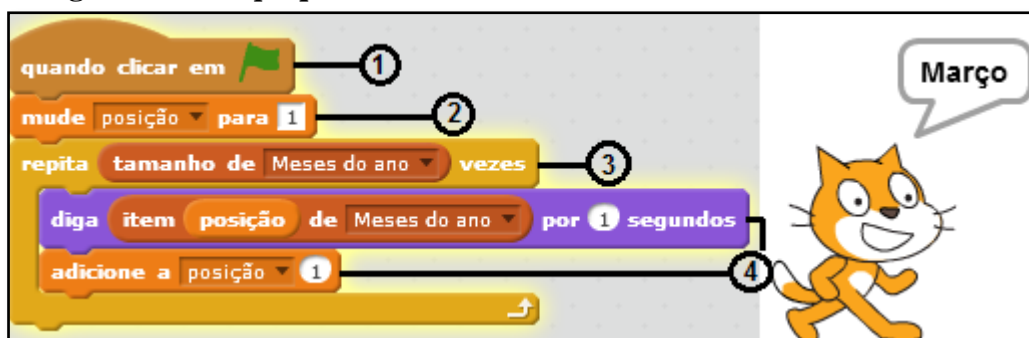
Fonte: elaborado pelos autores, 2017.

Suponha que entrou um aluno novo na turma, “Gilberto”, e seu nome precisa ser acrescentado. Considere, também, que a nota da “Amanda” precisa ser alterada. Essa alteração foi mostrada no *script* da Figura 96.

#### 4. Acessando elementos da lista por meio dos blocos “item de” e “tamanho de”

Como já foi dito anteriormente, qualquer item de uma lista pode ser acessado pelo número do seu índice e o bloco **item de** possibilita esse acesso. A Figura 97 mostra o *script* onde o ator Gato fala todos os itens da lista “Meses do ano”.

Figura 97 – Script que exibe os elementos da lista “Meses do ano”



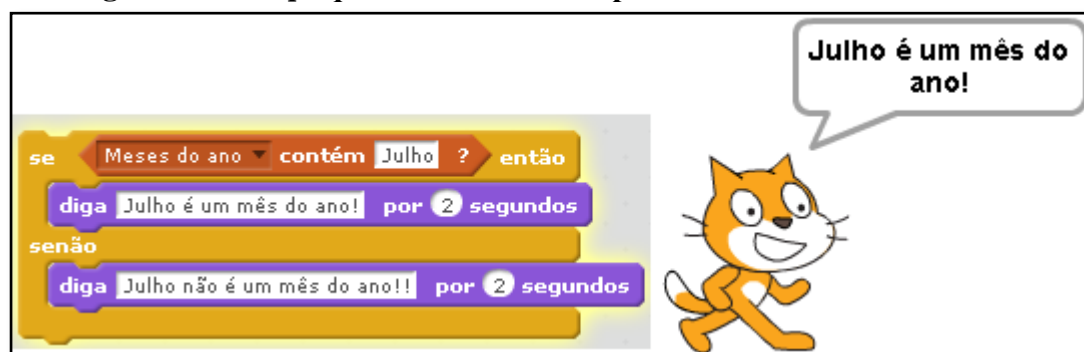
Fonte: elaborado pelos autores, 2017.

Para que os itens da **lista** interajam e sejam exibidos por meio do comando **diga**, o *script* usa uma variável denominada **posição**, a qual é usada como índice para selecionar um elemento específico da lista. Desta forma, quando a bandeira verde for clicada (1), o *script* muda o valor da **posição** para o primeiro item da lista (2), então o bloco **repita** executa um laço com o número de repetições do tamanho da lista “Meses do ano” por meio do bloco **tamanho de** (3). É a variável **posição** que determina cada

índice da **lista** e acrescenta 1 ao valor dessa variável a cada passagem do laço, para que o próximo item seja acessado, e assim, é exibido pelo comando **diga** (4).

5. **O bloco “contém”**: O bloco **contém** verifica se uma *string* está presente em uma lista, retornando um valor booleano, verdadeiro ou falso. A Figura 98 mostra um *script* em que esse comando é executado.

**Figura 98 – Script que verifica se Julho pertence à lista “Meses do ano”**



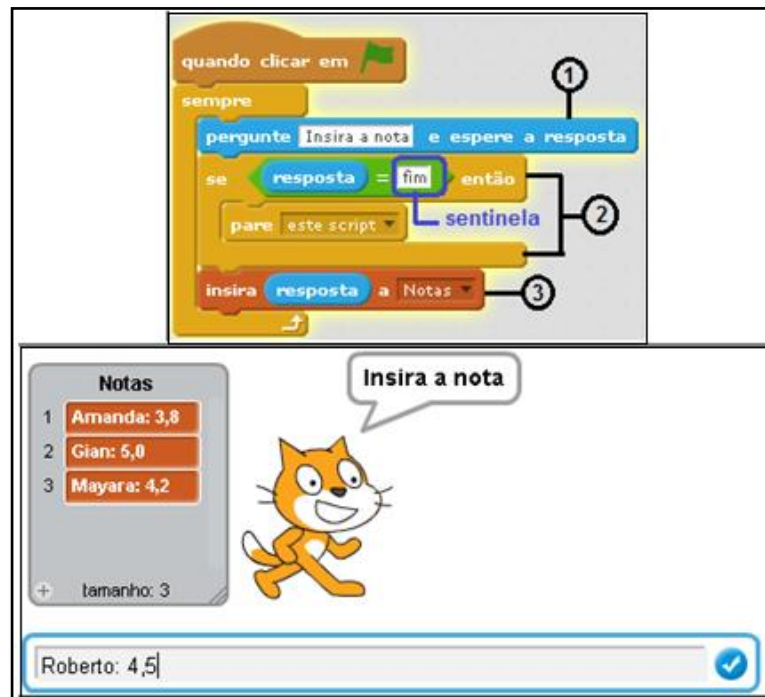
Fonte: elaborado pelos autores, 2017.

Como a *string* **Julho** pertence à **lista** “Meses do ano”, o comando **diga** dentro do bloco **se então, senão** retorna “Julho é um mês do ano!”. Caso contrário, a frase exibida pelo bloco **diga** seria “Julho não é um mês do ano!!”.

### **Preenchendo listas com dados fornecidos pelo usuário – determinando uma “sentinela”**

As listas também podem ser preenchidas com dados fornecidos pelo usuário. A Figura 99 mostra um *script* em que o programa pergunta ao usuário os dados a serem inseridos, vamos supor que sejam as notas da prova de matemática de uma turma. Como são desconhecidos quantos itens comporão a **lista**, precisamos determinar uma “sentinela”, de forma que ao ser inserido pelo usuário no campo de entrada, interromperá o *script*.

Figura 99 – Script que adiciona na lista itens fornecido pelo usuário



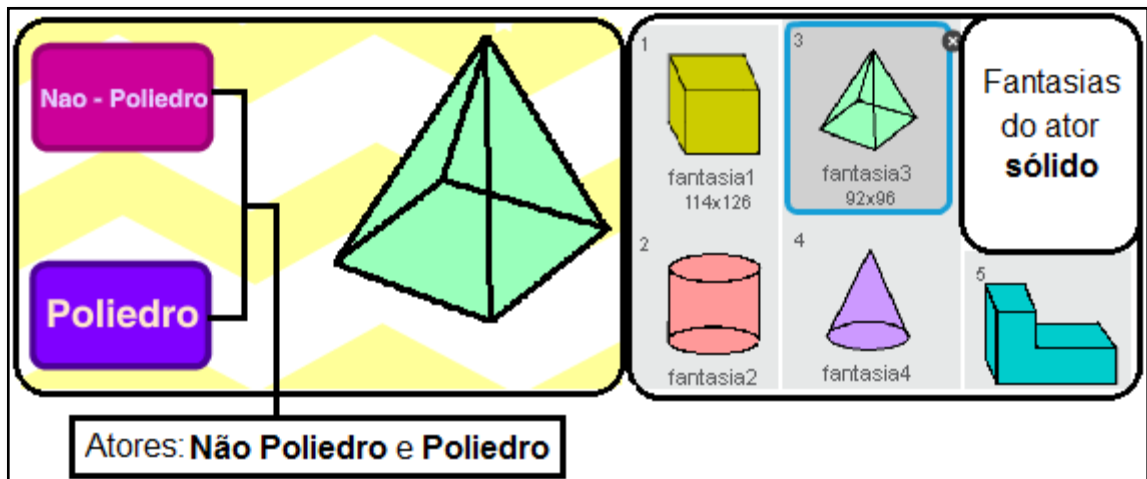
Fonte: elaborado pelos autores, 2017.

O *script* pede que o usuário forneça um item para a lista (inserir uma nota) (1) e compara essa informação com a sentinela (2). Escolhemos a sentinela “**fim**” para esse programa. Se o usuário inserir “**fim**” o *script* será interrompido, do contrário o dado da entrada será adicionado a **lista** e uma nova solicitação de entrada será feita ao usuário.

### Usando e criando com listas: Classificação de Sólidos Geométricos

Este projeto classifica sólidos geométricos em POLIEDROS e NÃO POLIEDROS. Ele mostra aleatoriamente 5 sólidos no Palco e o usuário precisa clicar no botão correto, conforme a Figura 100.

Figura 100 – Tela inicial ao usuário do projeto Classificação de Sólidos Geométricos

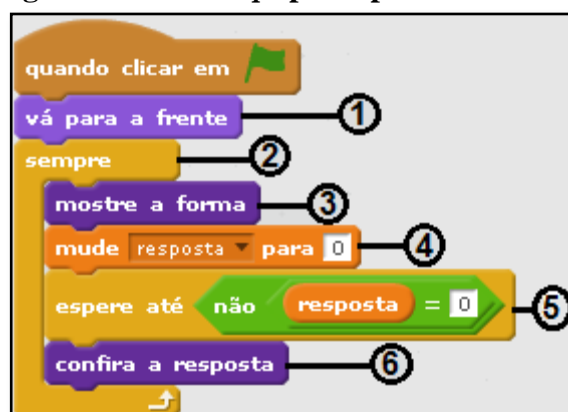


Fonte: elaborado pelos autores, 2017.

O projeto contém três atores: dois para os botões de resposta (**Poliedro** e **Não Poliedro**) e um, chamado **sólido**, que contém o *script* principal e cinco fantasias.

Quando a bandeira verde for clicada, o *script* do ator **sólido** mostrado na Figura 101 será executado.

Figura 101 – O *script* principal do ator sólido



Fonte: elaborado pelos autores, 2017.

Ao ser iniciado o programa, o ator “sólido” vai para frente dos demais atores (1), de maneira a não ficar sobreposto por eles. Em seguida, é executado o laço **sempre** (2), em que o *script* exibe um sólido geométrico aleatório a cada passagem do laço por meio do bloco

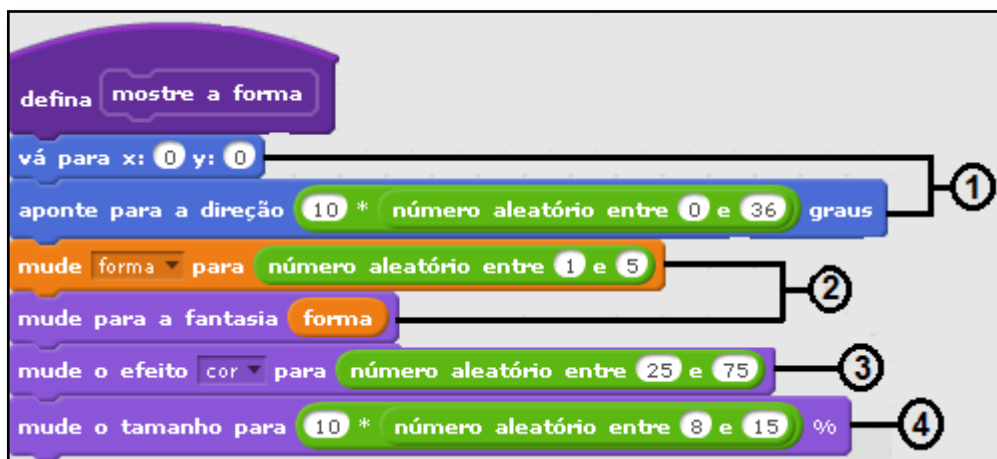


**mostre a forma** (3). Após mostrar o sólido geométrico, o *script* define a variável **resposta** com “0” (zero) (4) para informar que o usuário ainda não respondeu.

Então, o *script* espera até que a variável **resposta** seja alterada para um valor diferente de zero (5), o que acontecerá quando o usuário clicar em um dos botões **Poliedro** ou **Não-Poliedro**. Assim que o usuário selecionar uma das opções, Poliedro ou Não-Poliedro, o *script* chamará o comando **confira a resposta** (6) para dizer ao usuário se a resposta selecionada está correta ou não.

No *script* do ator sólido, apresentado na Figura 101, usamos os blocos personalizados **mostre a forma** e **confira a resposta**. A Figura 102 e Figura 103, respectivamente, mostram a definição de cada bloco criado.

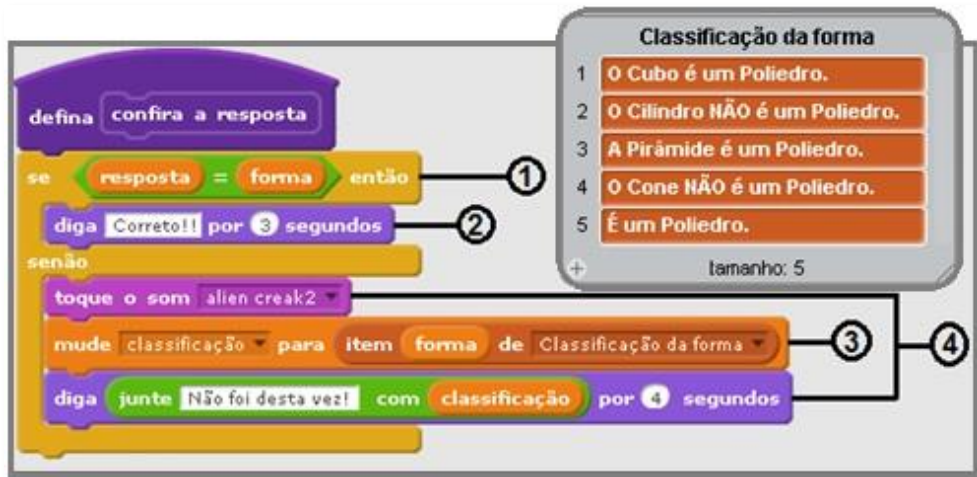
**Figura 102 – O *script* que define o bloco “mostre a forma”**



Fonte: elaborado pelos autores, 2017.

Inicialmente o bloco **mostre a forma** faz o ator sólido se mover para o centro do Palco na coordenada (0, 0) e apontar para uma direção aleatória entre 0° e 360° (1). A variável **forma** recebe um valor aleatório entre 1 e 5 e assim, a fantasia do ator sólido é alterada (2) e um sólido geométrico é exibido no Palco, esperando que o usuário o classifique como Poliedro ou Não Poliedro. E para dar a impressão de que uma nova figura está sendo criada a cada rodada, também é definido o efeito cor com um valor aleatório entre 25 e 75 (3) e o tamanho do ator é redimensionado de 80% a 150% de seu tamanho original (4).

Figura 103 – O script que define o bloco confira a resposta



Fonte: elaborado pelos autores, 2017.

O bloco **confira a resposta** confere a resposta do usuário comparando com o valor armazenado na variável **forma**, que especifica o tipo do sólido geométrico exibido no Palco.

Se os valores armazenados nas variáveis **resposta** e **forma** forem iguais (1), a resposta do usuário está correta (2). Caso contrário, a resposta está errada, um som é emitido e o ator informa a resposta correta (4). O comando **confira a resposta** utiliza a variável **forma** como índice da lista “classificação da forma” para obter a classificação correta do sólido geométrico exibido (3) e armazena esse valor na variável **classificação**.

A seguir, a Figura 104 mostra os *scripts* dos dois atores correspondentes aos botões. Eles são idênticos, exceto pelos valores atribuídos a variável **resposta**.

Figura 104 – O script dos atores correspondentes aos botões



Fonte: elaborado pelos autores, 2017.

Cada um desses *scripts* define o valor da variável **resposta** com um número diferente, de acordo com o botão que o usuário pressionar. Duas listas foram criadas, a lista “Não

Poliedros” contendo os números correspondentes às fantasias do ator sólido que não são poliedros, e uma lista denominada “Poliedro” contendo os números correspondentes às fantasias do ator sólido que são poliedros. A Figura 105 apresenta as listas “Poliedro” e “Não-Poliedros”.

**Figura 105 – As listas “Poliedro” e “Não Poliedros”**



Fonte: elaborado pelos autores, 2017.

Conforma a Figura 104, o *script* verifica por meio do bloco de lista **contém**, se o número da fantasia do ator sólido armazenado na variável **forma** pertence a lista “Poliedro” e/ ou “Não-Poliedro”, de acordo com a que for clicada pelo usuário. Se o número da **forma** pertencer a lista, a variável **resposta** armazenará esse valor, caso contrário assumirá o valor correspondente a lista correta e a resposta do usuário estará errada.

---

## Exercitando 6

1. Construa o projeto “Classificação de Sólidos geométricos”, apresentado anteriormente. Perceba que conforme foi construído, esse jogo será executado indefinidamente. Modifique o programa para adicionar um critério de fim de jogo e que monitore o número de respostas corretas e incorretas do usuário.
2. Crie um programa que peça ao usuário para fornecer nomes e notas de alunos e que armazene esses dados em duas listas. O programa deve parar de coletar os dados quando o usuário inserir “**fim**” na entrada que solicita o nome do aluno.
3. Crie um programa de Classificação de Quadriláteros. A aplicação deverá conter:

- a. Seis atores: cinco para os botões de resposta e um sexto que contenha o *script* principal.
- b. O ator que contém o *script* principal deverá ter as cinco fantasias que correspondem aos cinco quadriláteros do projeto.

## NOTA!

Este guia básico de utilização do software foi desenvolvido de forma a dar uma base para quem quer iniciar a programar em Scratch.

O material desenvolvido teve como base o livro **APRENDA A PROGRAMAR COM SCRATCH**, de Marjed Marji da editora Novatec.

## Referências

MARJI, M. **Aprendendo a programar com Scratch**. São Paulo: Novatec Editora Ltda, 2014.

# RESPOSTAS DO EXERCITANDO

**Apresentamos aqui algumas possíveis soluções para o Exercitando. Ressaltamos que outras soluções também são possíveis.**



# Capítulo 1

---

1.

```
digite A área de um trapézio de altura igual a 2/3, base maior medindo 22/9 e base menor medindo 5/9 é: por 5 segundos
digite 0,5 * ( 5 / 9 + 22 / 9 ) * 2 / 3
```

2.

```
quando clicar em
  apague tudo
  vá para x: 0 y: 0
  aponte para a direção 0 graus
  use a caneta
  repita 4 vezes
    mova 100 passos
    gire 90 graus
```



3.

The image displays four sections of Scratch code blocks, labeled A, B, C, and D, arranged vertically. Section A is a 2x2 grid. Section B is a single column. Section C is a single column. Section D is a single column.

**Section A:**

- quando clicar em [bandeira verde]
- vá para x: 0 y: -120
- sempre
  - mude x para posição x do mouse

**Section B:**

- quando clicar em [bandeira verde]
- vá para x: 0 y: 160
- aponte para a direção número aleatório entre 135 e 225 graus
- sempre
  - mova 10 passos
  - se tocar na borda, volte
  - se tocando em Raquete ? então
    - aponte para a direção número aleatório entre -30 e 30 graus

**Section C:**

- se tocando na cor [vermelha] ? então
  - pare todos

**Section D:**

- quando clicar em [bandeira verde]
- vá para x: 0 y: 160
- aponte para a direção número aleatório entre 135 e 225 graus
- sempre
  - mova 10 passos
  - se tocar na borda, volte
  - se tocando em Raquete ? então
    - aponte para a direção número aleatório entre -30 e 30 graus
    - toque o som [pop]
  - se tocando na cor [vermelha] ? então
    - pare todos



## Capítulo 2

1.

```
quando clicar em   
vá para x: 0 y: 0  
aponte para Rato  
deslize por 3 seg até x: 200 y: 130
```

2.

```
quando clicar em   
vá para x: 0 y: 0  
aponte para a direção 90 graus  
mude a cor da caneta para   
mude o tamanho da caneta para 3  
use a caneta  
apague tudo  
repita 1 vezes — número de lados  
  mova 2 passos — tamanho do lado  
  gire 360 / 3 graus — número de lados
```

3.

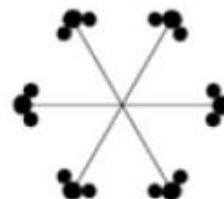
```
mude o estilo de rotação para em todas as direções  
repita 11 vezes  
  carimbe  
  gire 30 graus
```



```
mude o estilo de rotação para em todas as direções  
repita 8 vezes  
  carimbe  
  gire 45 graus
```



```
mude o estilo de rotação para em todas as direções  
repita 6 vezes  
  carimbe  
  gire 60 graus
```



4.

```
quando clicar em [bandeira]
vá para x: 0 y: 0
aponte para a direção 90 graus
mude a cor da caneta para [preto]
mude o tamanho da caneta para 3
use a caneta
apague tudo
repita 8 vezes
  repita 4 vezes
    mova 50 passos
    gire 90 graus
  gire 45 graus
```

5. As Figuras A e B correspondem aos scripts do ator planeta e foguete, respectivamente.

```
quando eu começar como clone
sempre
  adicione -10 a y
  se tocando em [foguete] então
    adicione a pontos 1
    toque o som [zoop]
    apague este clone
  se posição y < -120 então
    toque o som [alien creak2]
    apague este clone

quando clicar em [bandeira]
a) mude pontos para 0
  mostre
  repita 100 vezes
    vá para x: número aleatório entre -210 e 210 y: 180
    crie clone de este ator
    espere número aleatório entre 0.5 e 1.5 seg
  esconda
```

criar um bloco pontos na aba Variáveis

Figura A - resposta das alternativas a e c

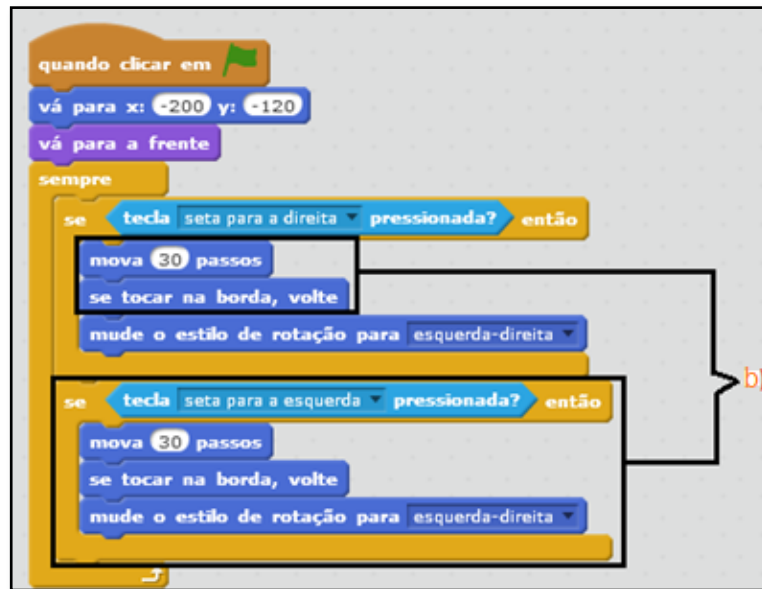


Figura B – resposta da alternativa b



## Capítulo 3

1. O programa simula um homem de palito se levantando de uma cadeira e andando pelo quarto. São dois atores, um que faz o movimento de levantar e o outro que anda pelo quarto. A Figura C e a Figura D mostram os atores e seus *scripts*, respectivamente.

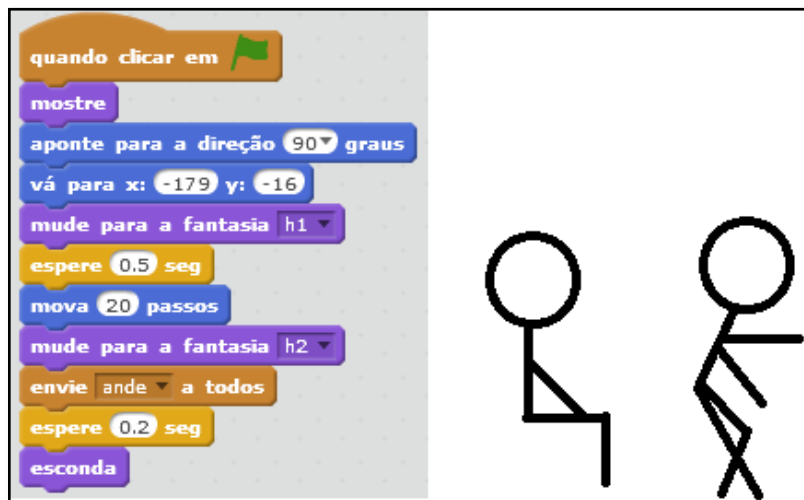


Figura C – *Script* e Fantasias do Ator 1

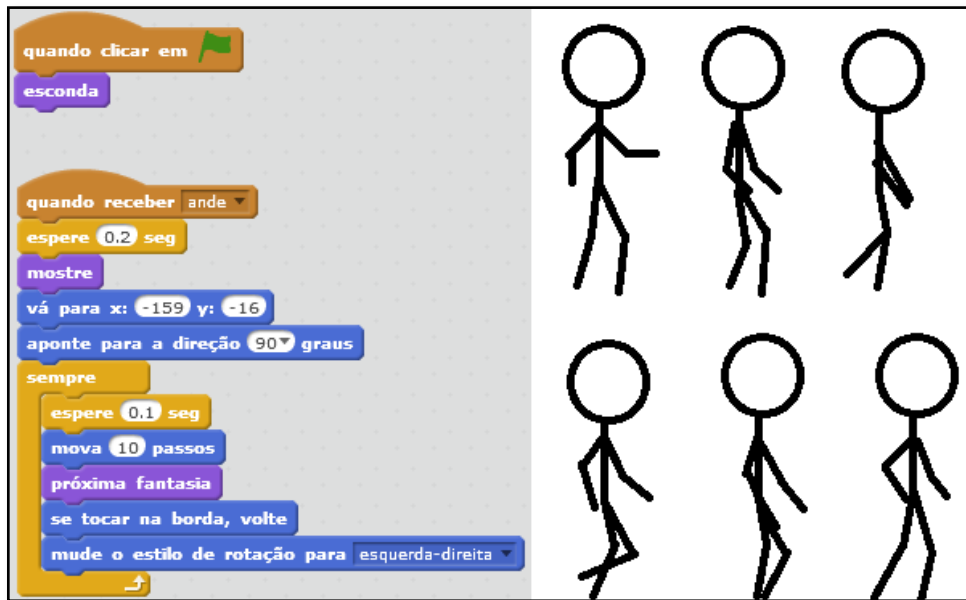


Figura D – Script e Fantasias do Ator 2

2. Os atores **bolhas**, **algas** e **coral** foram feitos a partir do pano de fundo escolhido para o Palco. O ator **polvo** foi escolhido da biblioteca do Scratch e ele possui duas fantasias. As Figuras E e F mostra esses sete atores e seus respectivos *scripts*.



Figura E – Atores e seus respectivos *scripts*



Figura F – Atores e seus respectivos *scripts*

3. São dois atores escolhidos da biblioteca do Scratch, Alex e Avery. Alex possui quatro fantasias e Avery duas fantasias, as quais vão sendo alteradas no decorrer do *script* para que o diálogo fique mais interessante. A Figura G e Figura H mostram os atores com seus respectivos *scripts*.



Figura G – Ator Alex e seu *script*

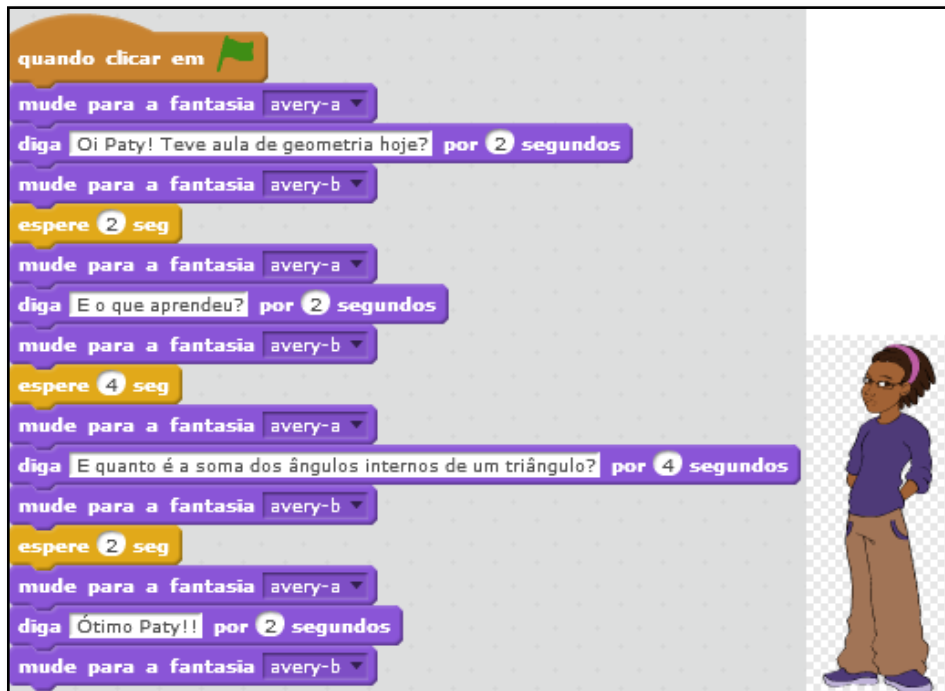
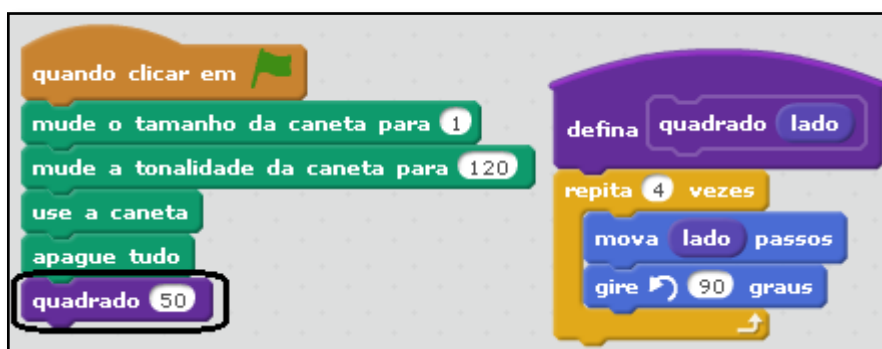


Figura H – Ator Avery e seu *script*



## Capítulo 4

1. A figura a seguir mostra à esquerda o *script* completo, e à direita da figura o procedimento que define o bloco **quadrado**.



2. A Figura I mostra o procedimento que define o bloco **Área do círculo**. Já a Figura J mostra o *script* do ator **círculo**.

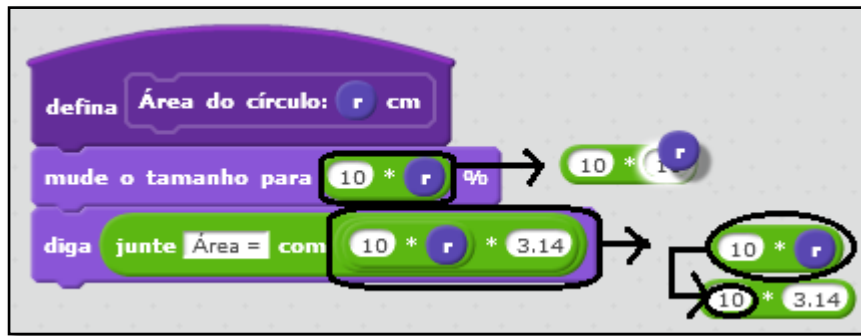


Figura I – Procedimento que define o bloco **Área do círculo**



Figura J – *Script* que determina a área do círculo dado o tamanho do lado

3. A Figura K mostra o procedimento de definição de cada novo bloco criado. Já a Figura L, mostra o *script* base para desenhar cada um dos blocos apresentados da Figura K.

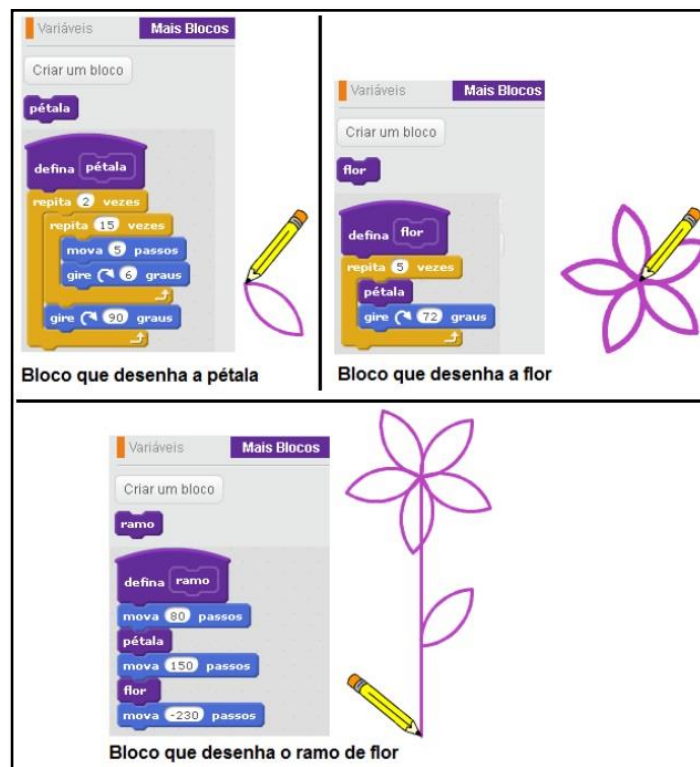


Figura K – Blocos criados na paleta “Mais Blocos”: **pétala**, **flor** e **ramo** respectivamente.



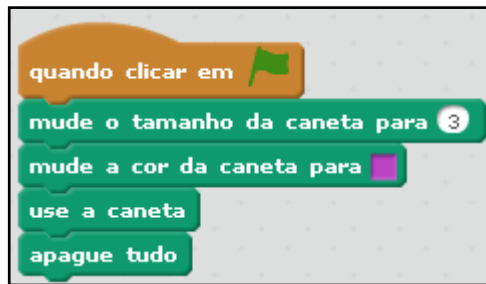
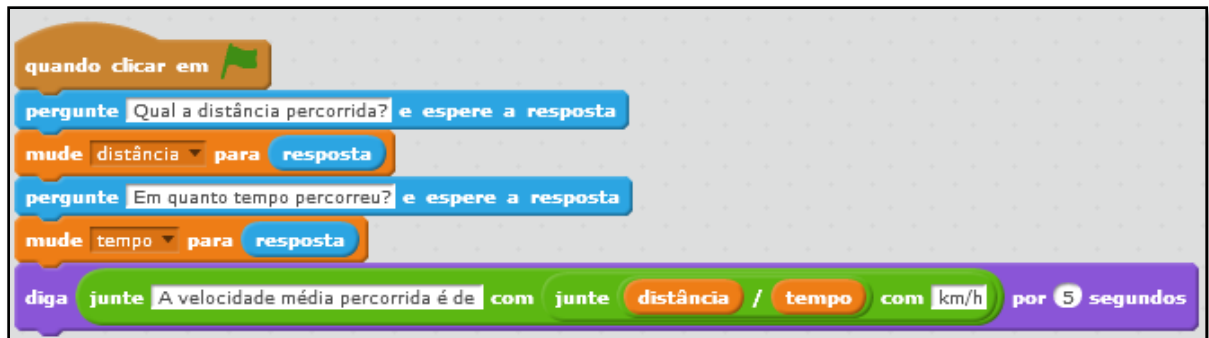


Figura L – *script* base para desenhar



## Capítulo 5

1.



2. A Figura M mostra o *script* que calcula o valor da hipotenusa dado os valores dos catetos pelo usuário. Observe a composição do bloco **diga**, que fornece a frase com o resultado final ao usuário (Figura N).

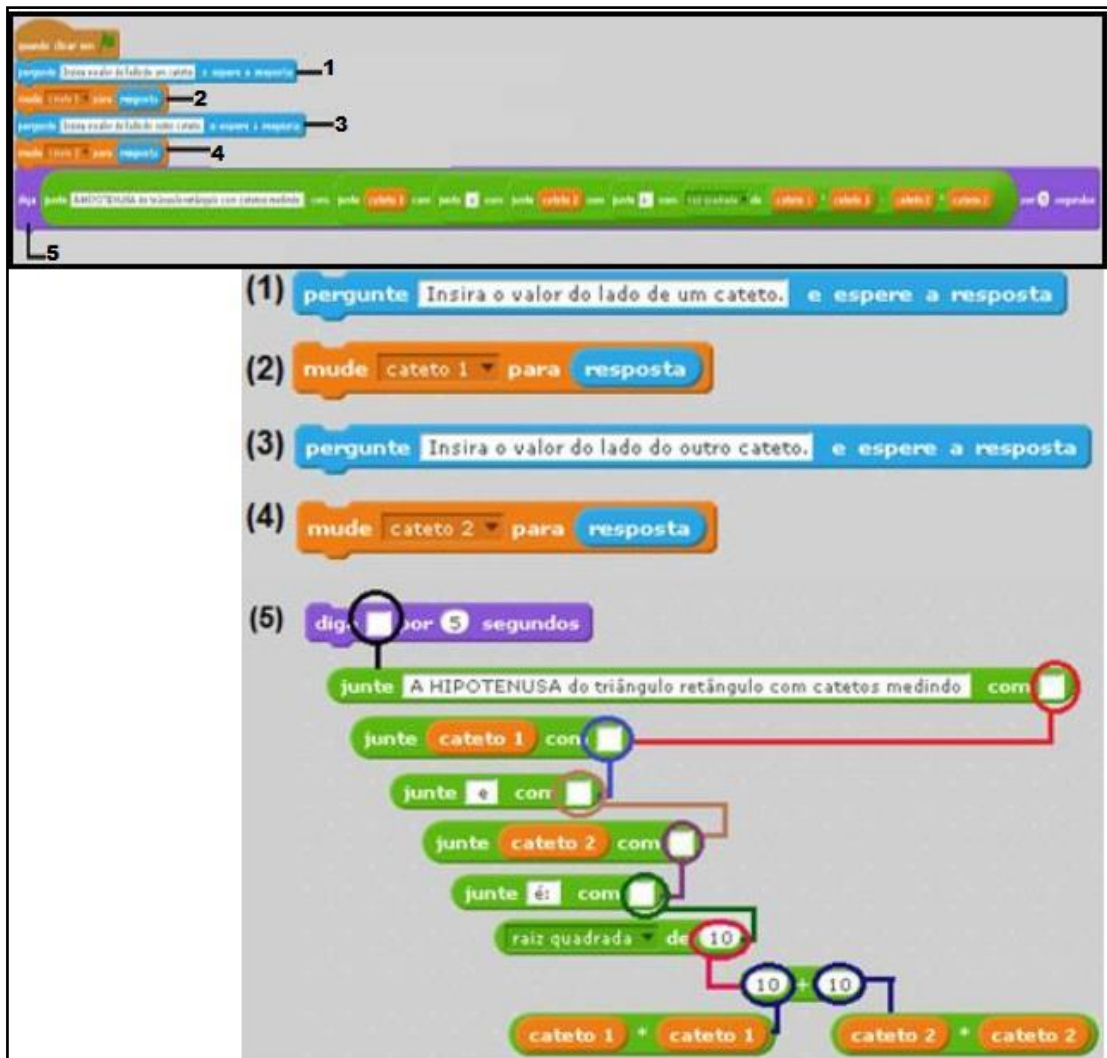


Figura M – Script do ator Gato

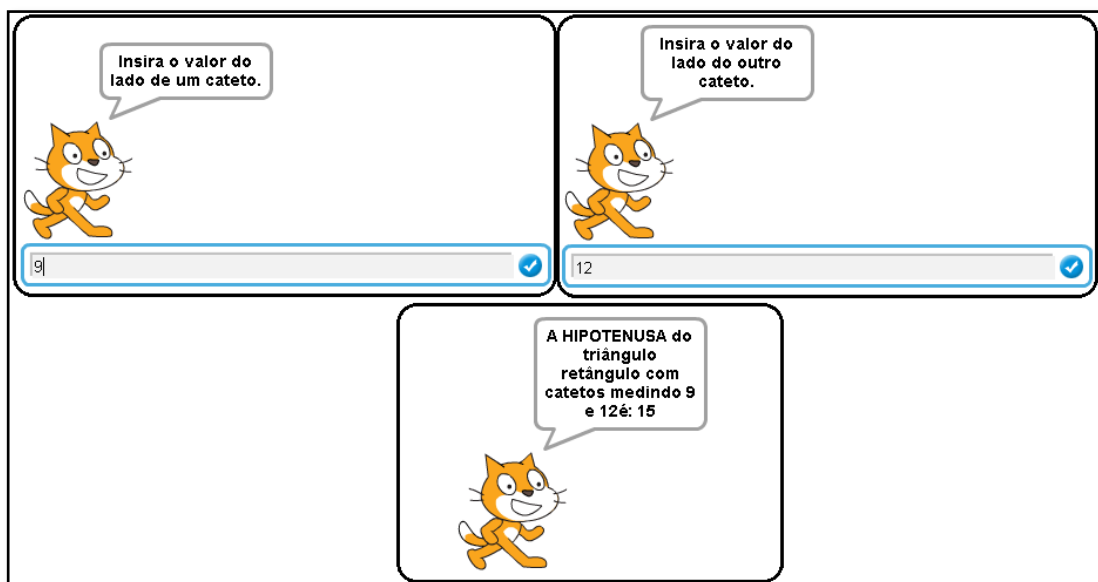


Figura N – Tela do usuário durante a execução do programa

3. As figuras seguintes mostram os três atores do projeto e seus respectivos *scripts*. O *script* da Figura O dá as instruções iniciais ao usuário. O *script* da Figura P desenha o polígono e este ator fica oculto. E a Figura Q mostra o *script* do Ator2, o qual representa o botão a ser clicado após selecionar o número de lados do polígono a ser desenhado.

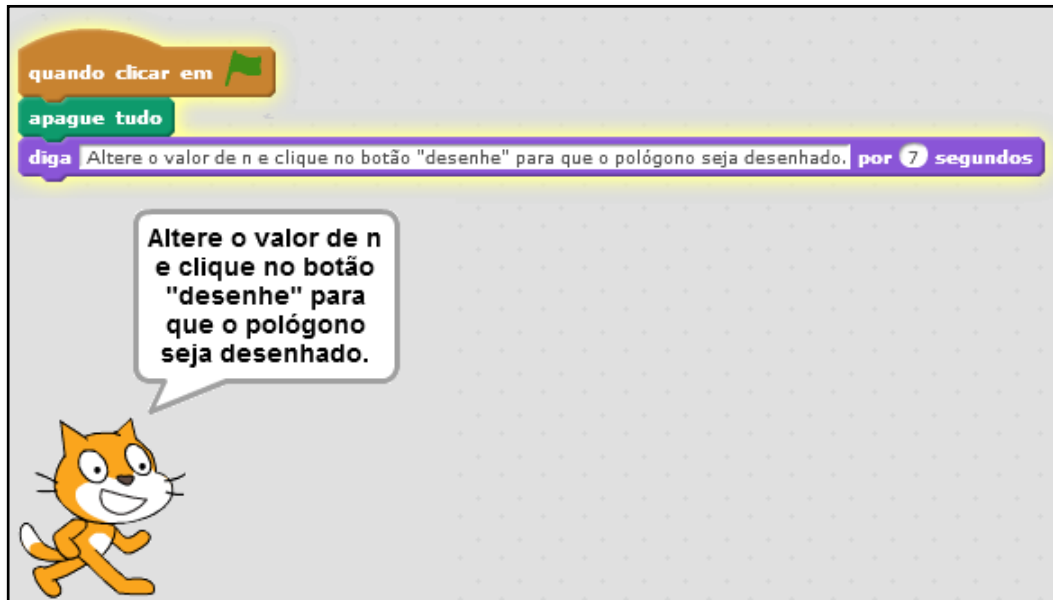


Figura O – Ator Cat1 e seu *script* em execução



Figura P – Ator1 e seu *script*



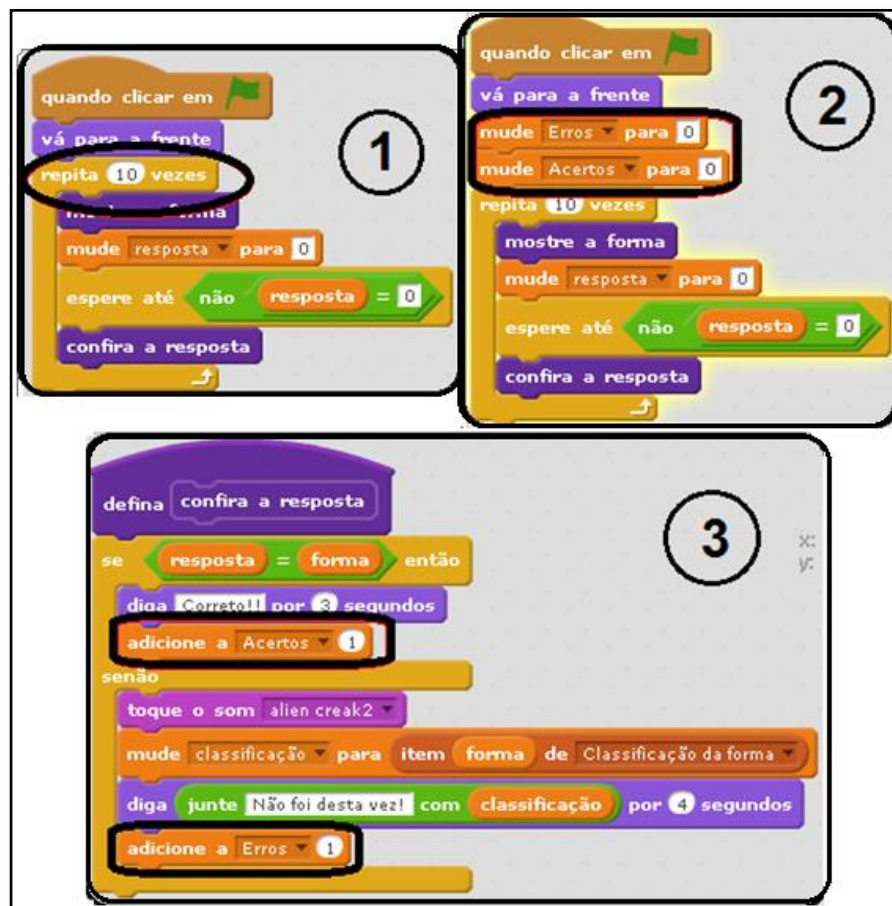
Figura Q – Ator2 e seu *script*



## Capítulo 6

1.

- (i) A figura (1) finaliza a aplicação após 10 repetições;
- (ii) A figura (2) muda os **Erros** e **Acertos** para zero quando a aplicação é iniciada;
- (iii) A figura (3) acrescenta 1 ponto a variável **Acertos** quando o usuário acerta a classificação do sólido geométrico e acrescenta 1 ponto a variável **Erros** quando o usuário erra a classificação do sólido geométrico.



2.

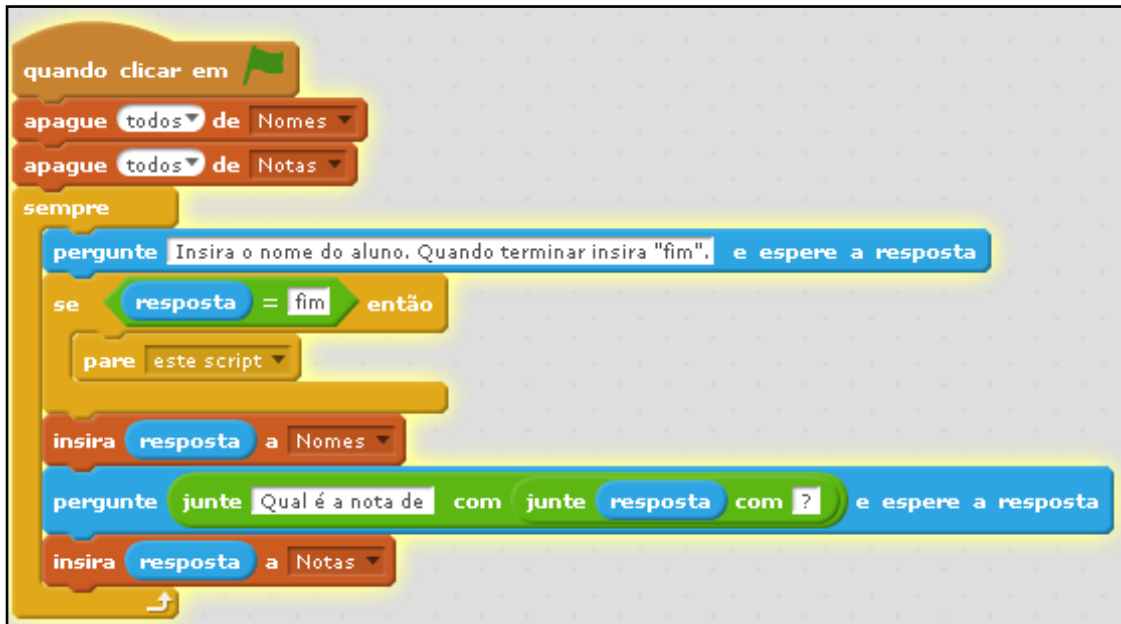


Figura R – Script do ator Gato

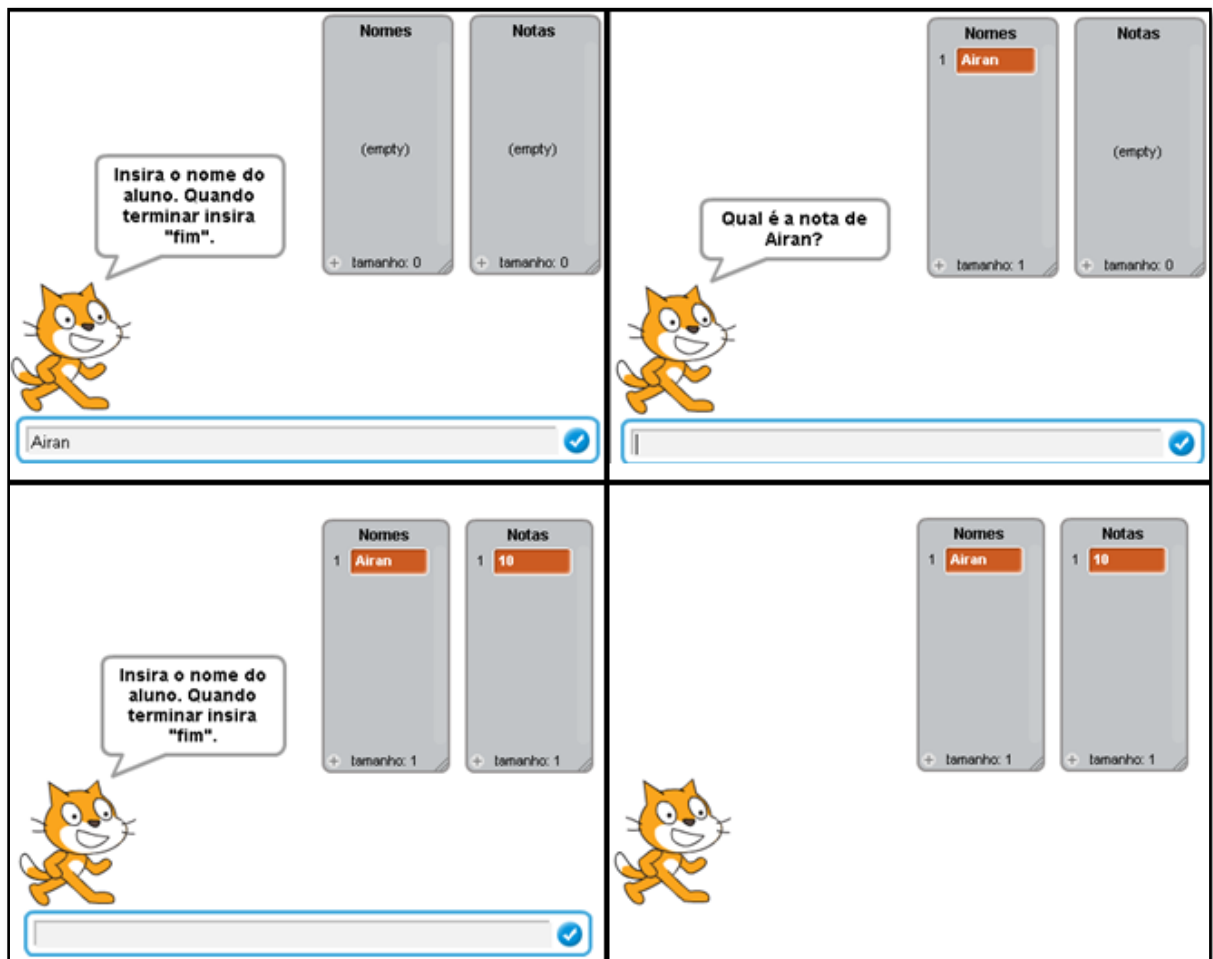


Figura S – Telas que aparecem ao usuário durante a execução do programa

3.

a) Os cinco atores que representam os botões de resposta.

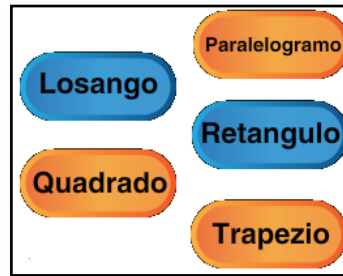


Figura T – Atores dos botões

b) A Figura U mostra o *script* principal do ator quadrilátero e uma sugestão da tela apresentada ao usuário. A Figura V mostra as cinco fantasias do ator **quadrilátero**, as quais foram construídas a partir do *Paint Editor* do Scratch.

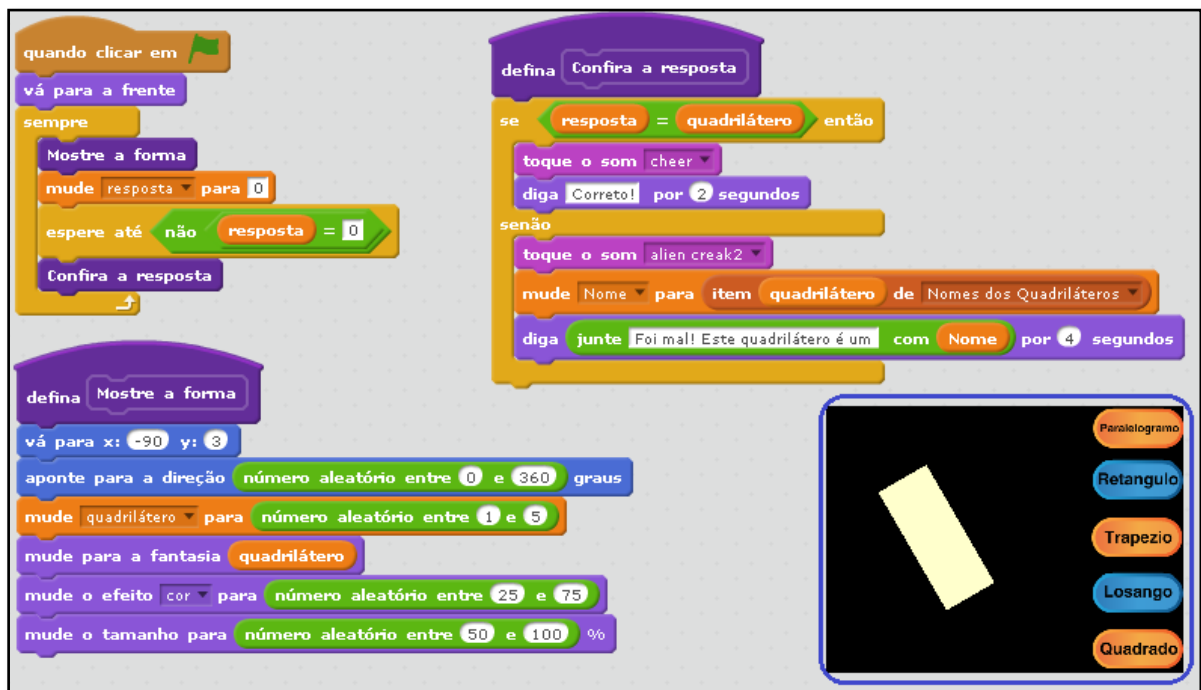


Figura U – *Script* principal do ator **quadrilátero**

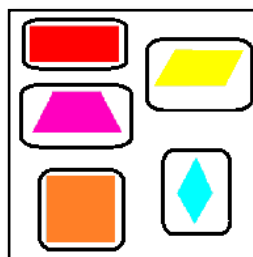


Figura V – Fantasias do ator **quadrilátero**