



2º Simpósio da Formação do  
Professor de Matemática da  
Região Centro-Oeste

# SCRATCH

## COMO USAR A PLATAFORMA NAS AULAS DE MATEMÁTICA

Bruno Hélio dos Santos Ramalho  
Elisa Fonseca Sena e Silva  
Maria Clara Ferreira da Silva Santos



Associação Nacional dos Professores  
de Matemática na Educação Básica



# **SCRATCH**

## **COMO USAR A PLATAFORMA NAS AULAS DE MATEMÁTICA**

**Scratch: como usar a plataforma nas aulas de matemática**

Copyright © 2022 Bruno Hélio dos Santos Ramalho, Elisa Fonseca Sena e Silva e Maria Clara Ferreira da Silva Santos

Direitos reservados pela Associação Nacional dos Professores de Matemática na Educação Básica.

A reprodução não autorizada desta publicação, no todo ou em parte, constitui violação de direitos autorais. (Lei 9.610/98)

**Associação Nacional dos Professores de Matemática na Educação Básica**

**Presidente:** Marcela Luciano Vilela de Souza

**Vice-Presidente:** Sérgio Augusto Amaral Lopes

**Diretores:** Ana Luiza de Freitas Kessler

Gilmar José Fava

Renata Magarinus

Sumaia Almeida Ramos

**2º Simpósio da Formação do Professor de Matemática da Região Centro-Oeste**

**Comissão Organizadora:**

Alex Ferreira Rossini

Ana Luiza de Freitas Kessler

Claudemir Aniz

Edson Rodrigues Carvalho

Elen Viviani Pereira Spreafico

Elisabete Sousa Freitas

Fernando Pereira de Souza

Gilmar Fava

Graziele Souza Mózer

Leandro Bezerra de Lima

Lilian Milena Ramos Carvalho

Marcela Luciano Vilela de Souza

Maria Botelho – Rede Estadual

Mateus Gianni Fonseca

Mustapha Rachidi

Raquel Bodart

Renata Magarinus

Rúbia Mara de Oliveira Santos

Sérgio Augusto Amaral Lopes

Sumaia Almeida Ramos

**Comitê Editorial:**

Ana Luiza de Freitas Kessler

Gilmar José Fava

Marcela Luciano Vilela de Souza,

Renata Magarinus

Sérgio Augusto Amaral Lopes

Sumaia Almeida Ramos

**Comitê Científico:**

Alex Ferreira Rossini

Edilene Simões Costa dos Santos

Marcela Luciano Vilela de Souza

Sérgio Augusto Amaral Lopes

Sumaia Almeida Ramos

Willy Alves de Oliveira

**Projeto gráfico e capa:** Gabriel Brasil Nepomuceno

**Diagramação e Assessoria Editorial:** Yunelsy Nápoles Alvarez

**ISBN:** 978-65-88013-20-5

**Distribuição**

Associação Nacional dos Professores de Matemática na Educação Básica

<http://www.anpmat.org.br> / email: [editoraanpmat@anpmat.org.br](mailto:editoraanpmat@anpmat.org.br)



2º Simpósio da Formação do  
Professor de Matemática da  
Região Centro-Oeste

# SCRATCH

## COMO USAR A PLATAFORMA NAS AULAS DE MATEMÁTICA

Bruno Hélio dos Santos Ramalho  
Elisa Fonseca Sena e Silva  
Maria Clara Ferreira da Silva Santos

1ª edição

2022

Rio de Janeiro



Associação Nacional dos Professores  
de Matemática na Educação Básica



# Sobre os autores



Natural de Maceió, AL, com imenso interesse em matemática e suas aplicações, principalmente as que se destinam à computação. Formado em licenciatura em Matemática pela Universidade Federal de Alagoas. O interesse por tal licenciatura veio aos 11 anos, nutrido pelos diversos professores de matemática com que lidei ao longo dos anos. Atualmente sou professor de Matemática e do itinerário de Ciência e Tecnologia na Escola Sesi de Educação Básica Industrial Abelardo Lopes, onde busco alimentar minha paixão pela matemática e suas aplicações.



**Bruno Helio dos Santos Ramalho**

profbhramalho@gmail.com



**Elisa Fonseca Sena e Silva**

elisa.silva@im.ufal.br

Mentalidades Matemáticas, Ensino de Matemática e Formação de Professores de Matemática.

Sou estudante de licenciatura em Matemática na Universidade Federal de Alagoas, e também voluntária no projeto de extensão Sem Mais Nem Menos e ex bolsista do Pibid. Escolhi a matemática como minha futura profissão porque compreendo a contribuição que essa ciência tem para a sociedade, almejando, assim, de forma direta ou indireta, contribuir para o ensino da matemática.



**Maria Clara Ferreira da Silva**

maria.ferreira@im.ufal.br





# Sumário



---

Sobre os autores	vi
Prefácio	xv
Introdução	2
1 A plataforma <i>Scratch</i>	4
2 Criando com o <i>Scratch</i>	7
2.1 Apresentando a interface . . . . .	8
2.2 Primeiro dia: <i>Flappy Bird</i> . . . . .	17
2.3 Segundo dia: <i>Snake</i> com operações . . . . .	29
3 Considerações finais	32
Referências Bibliográficas	34

# Lista de Figuras



---

---

1	Linguagem de Programação Visual: <i>Scratch</i> . . . . .	5
2	Interface do <i>Scratch</i> . . . . .	8
3	Tela de apresentação . . . . .	10
4	Janela dos atores . . . . .	11
5	Janela de programação: Aba Código . . . . .	11
6	Janela de programação: Aba Fantasias . . . . .	12
7	Janela de programação: Aba Sons . . . . .	13
8	Categoria dos blocos: Eventos, Controle e Sensores . . . . .	15
9	Categoria dos blocos: Operadores, Variáveis e Movimento . . . . .	16
10	Categoria dos blocos: Aparência, Som e Meus Blocos . . . . .	17
11	Flappy Bird . . . . .	18
12	Adicionando o ator principal . . . . .	19
13	Criando as variáveis . . . . .	19
14	Algoritmo do <i>Parrot</i> : Movimentação . . . . .	21
15	Algoritmos do <i>Parrot</i> : Efeitos visuais e sonoros: . . . . .	22
16	Selecionando o cenário . . . . .	23
17	Botão “Pintar” . . . . .	24
18	Desenhando os Canos . . . . .	24
19	Algoritmo de clonagem . . . . .	25
20	Algoritmo de movimento para os <i>clones</i> . . . . .	27
21	Algoritmo de Fim de jogo . . . . .	28
22	<i>Snake</i> com operações . . . . .	29
23	<i>Snake</i> com operações: Atriz Avery . . . . .	30

# Lista de Tabelas



1	Funções presentes em cada seção . . . . .	9
2	Categoria dos blocos de programação . . . . .	14

# Prefácio





Este *e-book* é baseado no minicurso homônimo que foi ministrado no 2º Simpósio da Formação do Professor de Matemática da Região Centro-Oeste. Temos como objetivo apresentar a plataforma de programação visual em blocos *Scratch*, mostrando como a mesma pode ser utilizada para aprender e ensinar programação e para construir jogos matemáticos educativos.

Assim como no evento, escrevemos para licenciandos em Matemática e professores formados na área que, seguindo os tutoriais que aqui se encontram, podem aprender não só a utilizar a plataforma *Scratch* como a criar seus próprios jogos educativos. Para tanto, assim como nas oficinas, abordaremos dois jogos: um *clone* do jogo “*Angry Bird*” e o jogo “*Snake* com operações”, uma variação do conhecido ‘jogo da cobrinha’.

Ao apresentar o primeiro jogo, temos o intuito de mostrar como a plataforma funciona e como a programação relaciona-se com a matemática. Já o segundo jogo retrata como é possível adaptar um jogo para inserir problemas de matemática mais direcionados como, por exemplo, abordar as quatro operações básicas. Com isso, esperamos contribuir para que os leitores utilizem no futuro o *Scratch* tanto para ensinar lógica de programação como para fazer seus próprios jogos de matemática.

Os autores



# Introdução



A programação, com ênfase na linguagem visual, traz diversos benefícios para o ensino de matemática na educação básica, sendo um deles a aproximação da matemática com o cotidiano dos estudantes. Nesse sentido, [Conceição, Mendes e Borges \(2015\)](#) em seu artigo acerca dos fatores que desmotivam os alunos com relação à matemática, dizem que:

O ensino tradicional da matemática não vem despertando tanto interesse nos alunos atualmente. Com tantos meios de comunicações e tecnologias à disposição desses alunos não basta apenas os professores dominarem o conteúdo da disciplina ou apenas as técnicas de educar; o professor precisa estar atualizado tanto com a matemática específica quanto com os meios educacionais atuais, interagindo com alguns meios tecnológicos e novos métodos de ensino em suas aulas. ([CONCEIÇÃO; MENDES; BORGES, 2015](#), p. 2).

A tecnologia vem aproximando-se cada vez mais do cotidiano dos brasileiros e não é diferente no âmbito escolar. Depois da pandemia de Covid-19, houve um crescimento ao acesso e uso de tecnologia pelos brasileiros, além do acesso habitual como redes sociais, jogos eletrônicos e mídias por *streaming*. Trabalho, educação e lazer passaram a se desenvolver majoritariamente no meio virtual.

A idade do público brasileiro que possui acesso aos celulares vem diminuindo. A pesquisa “Crianças e *smartphones* no Brasil 2020” aponta que 61% das crianças de 0 a 3 anos têm acesso a *smartphone*, seja próprio ou dos pais. Essa proporção sobe conforme a idade, e alcança 95% na faixa de 10 a 12 anos.

O interesse por jogos digitais no cotidiano brasileiro também vem crescendo. A pesquisa “Game Brasil 2021” revela que 72% dos brasileiros têm costume de jogar jogos digitais, e, dentre esses, 40,8% costumam usar o *smartphone* ou *tablet* todos os dias para jogar. Além disso, a pesquisa “Uso de apps no Brasil 2021” ressalta que 59% dos brasileiros com *smartphone* declaram jogar *games* no aparelho.

Posto isso, estratégias de ensino de matemática que abranjam a tecnologia como tópico, destacando as tecnologias digitais relacionadas a jogos, que é um interesse relacionado ao lazer ou prazer pessoal de muito jovens e adultos brasileiros, podem e devem ser usadas para atrair o interesse do aluno e conectar o ensino com o seu cotidiano. Dessa forma, pode-se contextualizar o ensino da matemática com o atual cotidiano de diversos alunos brasileiros. Segundo os Parâmetros Curriculares Nacionais para o Ensino Médio - PCNEM ([BRASIL, 2020](#), p. 78):

O tratamento contextualizado do conhecimento é o recurso que a escola tem para retirar o aluno da condição de espectador passivo. Se bem trabalhado permite que, ao longo da transposição didática, o conteúdo do ensino provoque aprendizagens significativas que mobilizem o aluno e estabeleçam entre ele e o objeto do conhecimento uma relação de reciprocidade. A contextualização evoca por isso áreas, âmbitos ou dimensões presentes na vida pessoal, social e cultural, e mobiliza competências cognitivas já adquiridas.

A própria Base Nacional Comum Curricular – BNCC ([BRASIL, 2018](#)) ressalta que jogos, planilhas eletrônicas, *softwares* de geometria dinâmica, dentre outros recursos didáticos, têm papel essencial para a compreensão e utilização das noções matemáticas no

ensino fundamental, tanto nos anos iniciais, quanto nos anos finais. Entretanto, o objetivo de ensinar programação visual na educação de matemática vai além de aliar a ludicidade dos jogos com o ensino de matemática. A ideia central é tornar o aluno construtor do seu conhecimento através da elaboração dos seus jogos favoritos ou seus próprios jogos com suas próprias histórias.

Nesse sentido, a plataforma *Scratch* apresenta-se como uma ótima ferramenta para os docentes abordarem tanto a programação em sala de aula como para elaborar seus próprios jogos educativos. Sendo assim, este *e-book*, que tem como objetivo ensinar a utilizar a plataforma a partir da criação de dois jogos, está dividido da seguinte forma: no próximo capítulo mostramos resultados de pesquisas que utilizaram o *Scratch* para fins educativos. Em seguida, ensinamos como mexer na plataforma para, em seguida, darmos o passo a passo de como construir dois jogos: um *clone* do “*Angry Birds*” e o “*Snake* com operações”, uma variação do ‘jogo da cobrinha’ com conteúdos matemáticos.

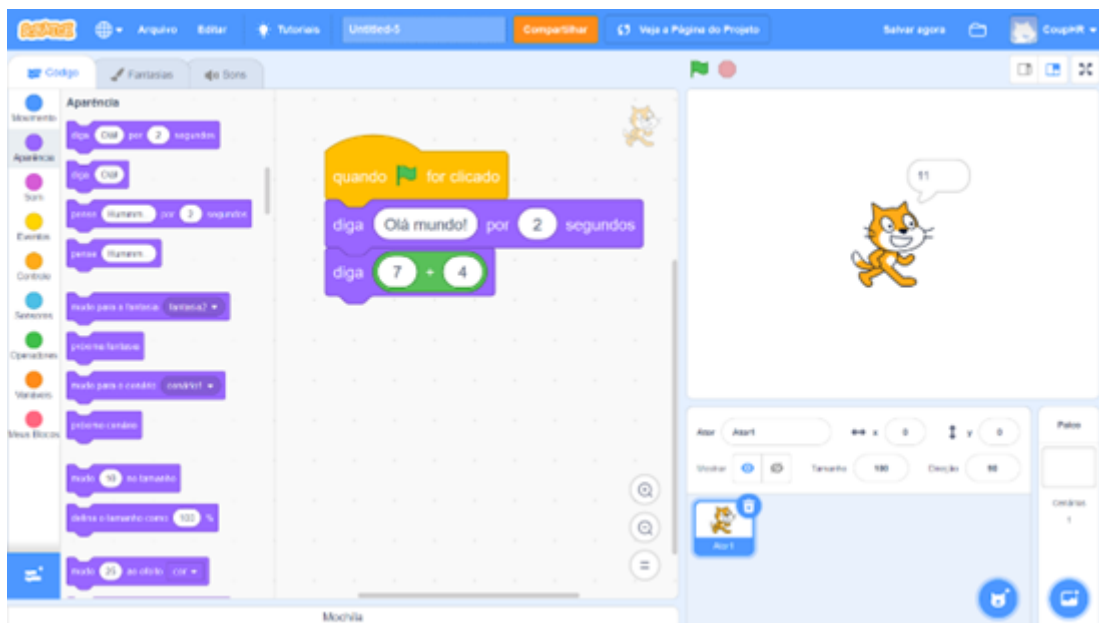
# Capítulo 1

## A plataforma *Scratch*



O *Scratch* é uma linguagem de programação visual e comunidade *on-line* semelhante a uma rede social, mas com foco em programação, em que os usuários podem programar usando blocos lógicos de encaixe (Figura 1) e compartilhar seus projetos como histórias, jogos e animações. Enquanto cria com o *Scratch*, o usuário aprende a pensar de forma criativa, trabalhar em colaboração e exercitar o seu pensamento computacional. O estudante desenvolve seus programas manipulando elementos visuais, por meio de caixas com comandos preestabelecidos, diferente das linguagens de programação textuais que necessitam especificar os comandos textualmente, como podemos ver abaixo na Figura 1.

Figura 1: Linguagem de Programação Visual: *Scratch*



Fonte: Autores, 2021.

De acordo com o relatório anual de 2019, o *Scratch* é utilizado em mais de 150 países e atualmente 1,51% (865.319) dos usuários é de brasileiros e no mesmo ano chegou ao marco de 170 milhões de visitantes únicos, 60 milhões de projetos criados e 20 milhões de pessoas que criaram projetos pela plataforma.

Shimohara e Sobreira (2015) relataram a experiência de um trabalho desenvolvido com alunos do 5º ano do Ensino Fundamental I, que tinha como objetivo utilizar programação para a criação de jogos digitais com elementos matemáticos, desenvolvendo, assim, habilidades de argumentação, observação, dedução, criação de roteiros e estratégias, incluindo desafios de matemática, para criação de jogos digitais. Em suas discussões, as autoras evidenciam:

A utilização do *Scratch* como recurso pedagógico propiciou o envolvimento dos alunos no desenvolvimento do conteúdo curricular. A programação com *Scratch* promove o desenvolvimento de conceitos matemáticos, pois envolve conhecimentos de localização no plano cartesiano (posição x, posição y), a lógica das variáveis e o uso de condições (se e senão). (SHIMOHARA; SOBREIRA, 2015, p. 79).

Chegaram à conclusão de que após a criação de seus jogos, os alunos ficaram empolgados em jogar e compartilhar suas produções, pois tiveram a oportunidade de criar ao invés de apenas utilizar.

Mota *et al.* (2014) usaram o *Scratch* para o ensino de lógica da computação com alunos do Ensino Médio junto às disciplinas de física e matemática. Os alunos que realizaram a oficina trabalharam problemas relacionados aos conteúdos vistos em aula. Após o término da oficina, além do interesse em lógica de programação, os alunos também continuaram desenvolvendo jogos para a resolução de problemas de matemática que foram resolvidos anteriormente de forma tradicional.

Para que o ensino não fique defasado, cabe aos professores buscar por novas estratégias de ensino que motivem e agreguem conhecimento matemático junto às experiências cotidianas de seus alunos. Tendo em vista que o ensino de programação, assim como o uso de tecnologias digitais para o ensino de matemática, estão obtendo resultados positivos como ferramenta ou estratégia de ensino, propusemos este minicurso para ensinar como utilizar a plataforma *Scratch* para o ensino da matemática.



# Capítulo 2

## Criando com o *Scratch*

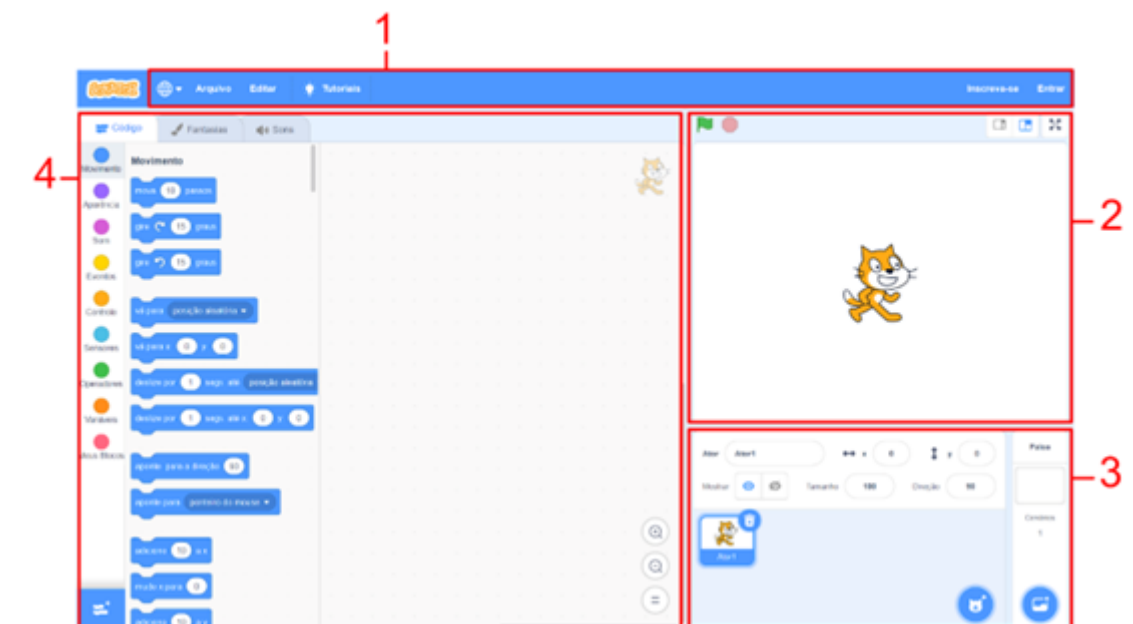


O minicurso ocorreu em dois dias por meio de videoconferência. No primeiro dia, foi apresentada a plataforma para os participantes e elaborado o projeto “Flappy Bird”. O objetivo do primeiro dia foi mostrar como a programação utiliza diversos conhecimentos matemáticos como lógica, variáveis, funções etc. Seja qual for o projeto que esteja sendo desenvolvido através da programação, irá necessitar de conhecimentos matemáticos. Através do *Scratch* podemos desenvolver diversos projetos e trabalhar conceitos matemáticos com os alunos da educação básica. No segundo dia, foi elaborado o jogo “Snake com operações”, que é o jogo da Serpente adicionando um elemento matemático ativo, em que, ao jogar, o aluno estará também trabalhando algum tópico de matemática. O objetivo do segundo dia foi demonstrar que a partir da programação podemos desenvolver jogos (ferramentas) lúdicos para serem usados em nossas aulas. Esse tipo de ferramenta é extremamente poderoso, pois ao jogar um jogo com elementos matemáticos, o aluno está trabalhando o conhecimento de maneira passiva.

## 2.1 Apresentando a interface

Essa é uma breve apresentação do ambiente de programação presente no *Scratch*. Após baixar<sup>1</sup> o aplicativo e instalar, ou acessar o *site*<sup>2</sup> e clicar na aba “criar”, será apresentada a seguinte interface:

Figura 2: Interface do *Scratch*



Fonte: Autores, 2021.

Para melhor compreensão, podemos dividir a interface em 4 seções, detalhadas no

<sup>1</sup>Disponível em: <<https://Scratch.mit.edu/download>>.

<sup>2</sup>Disponível em: <<https://Scratch.mit.edu/>>

Quadro 1, abaixo.

Quadro 1: Funções presentes em cada seção

### 1. Barra de menu:

Localizada na parte superior, aqui teremos as seguintes opções:

- Mudar a linguagem clicando no ícone “globo”;
- “Arquivo” terá as opções de criar um novo projeto, salvá-lo, carregar um projeto do computador ou baixar o projeto para o computador;
- “Editar” dará a opção do modo turbo, onde as ações ocorrerão de maneira acelerada;
- “Tutoriais” dará acesso à biblioteca com diversos tutoriais de miniprojetos com o intuito de auxiliar a criação com o *Scratch*.

### 2. Tela de apresentação (que pode ser vista com mais detalhes na figura 3):

Localizada na parte superior direita. Aqui teremos uma visualização prévia do projeto.

- 2.1. Bandeira verde: inicia/reinicia a execução do projeto;
- 2.2. Octógono vermelho: cessa a execução do projeto;
- 2.3. No canto superior direito da tela de apresentação encontram-se três botões que permitem alterar o tamanho da tela de apresentação.

### 3. Janela dos atores (que pode ser vista com mais detalhes na figura 4):

Localizada na parte inferior direita. Local reservado para os objetos de animação do projeto.

- 3.1. Configuração do ator: pode-se renomear o ator, alterar sua posição no plano cartesiano, esconder, mudar o tamanho e sua direção;
- 3.2. Janela dos atores: local onde todos os atores utilizados no projeto ficarão;
- 3.3. Palco: local onde ficam os cenários do projeto;
- 3.4. Adicionar ator ou cenário: possibilita adicionar novos atores ou cenários diretamente do computador, da biblioteca do próprio *Scratch*, além da opção de pintar seu próprio ator.

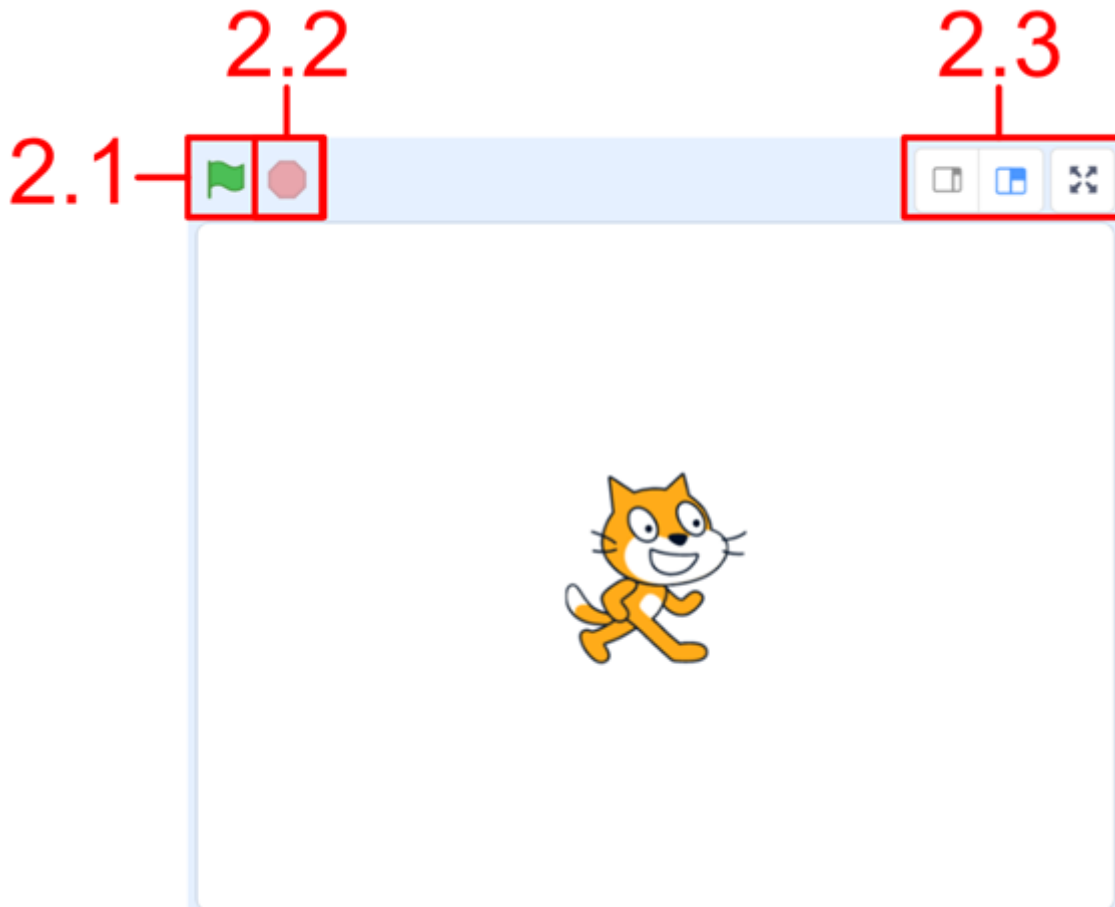
#### 4. Janela de programação:

Aqui é onde o *Scratch* brilha. Dividida em 3 abas (Código, Fantasias e Sons). Todos os algoritmos do projeto serão feitos através dos blocos lógicos predefinidos.

- 4.1. Código (visto em mais detalhes na figura 5): aba onde encontram-se todos os blocos de programação; os blocos estão divididos em categorias e cores distintas com funcionalidades diferentes;
- 4.2. Fantasias (vistas em mais detalhes na figura 6): aba onde é possível adicionar e editar visuais para o ator;
- 4.3. Sons (vistos em mais detalhes na figura ??): aba onde é possível adicionar e editar sons para utilizar com o ator. Há a opção de carregar sons do computador, gravar ou utilizar a biblioteca do *Scratch*.

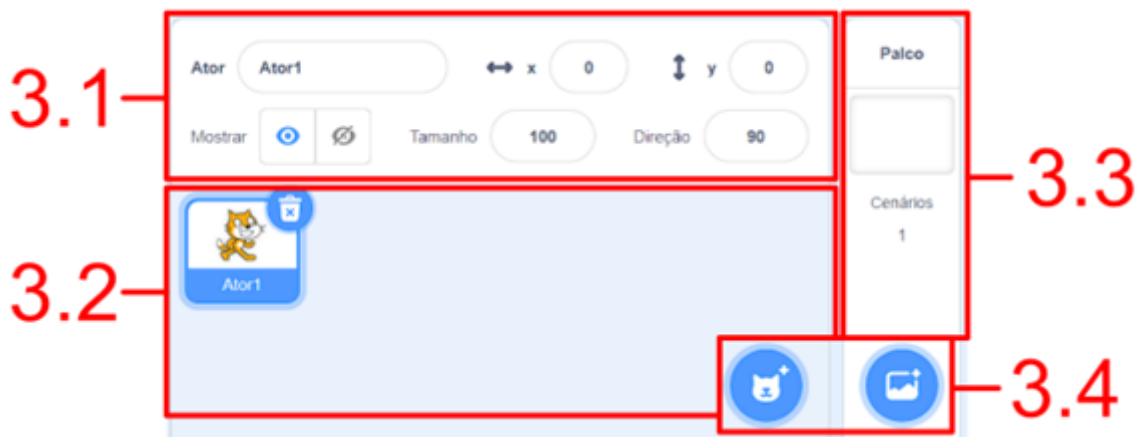
Fonte: Autores, 2021

Figura 3: Tela de apresentação



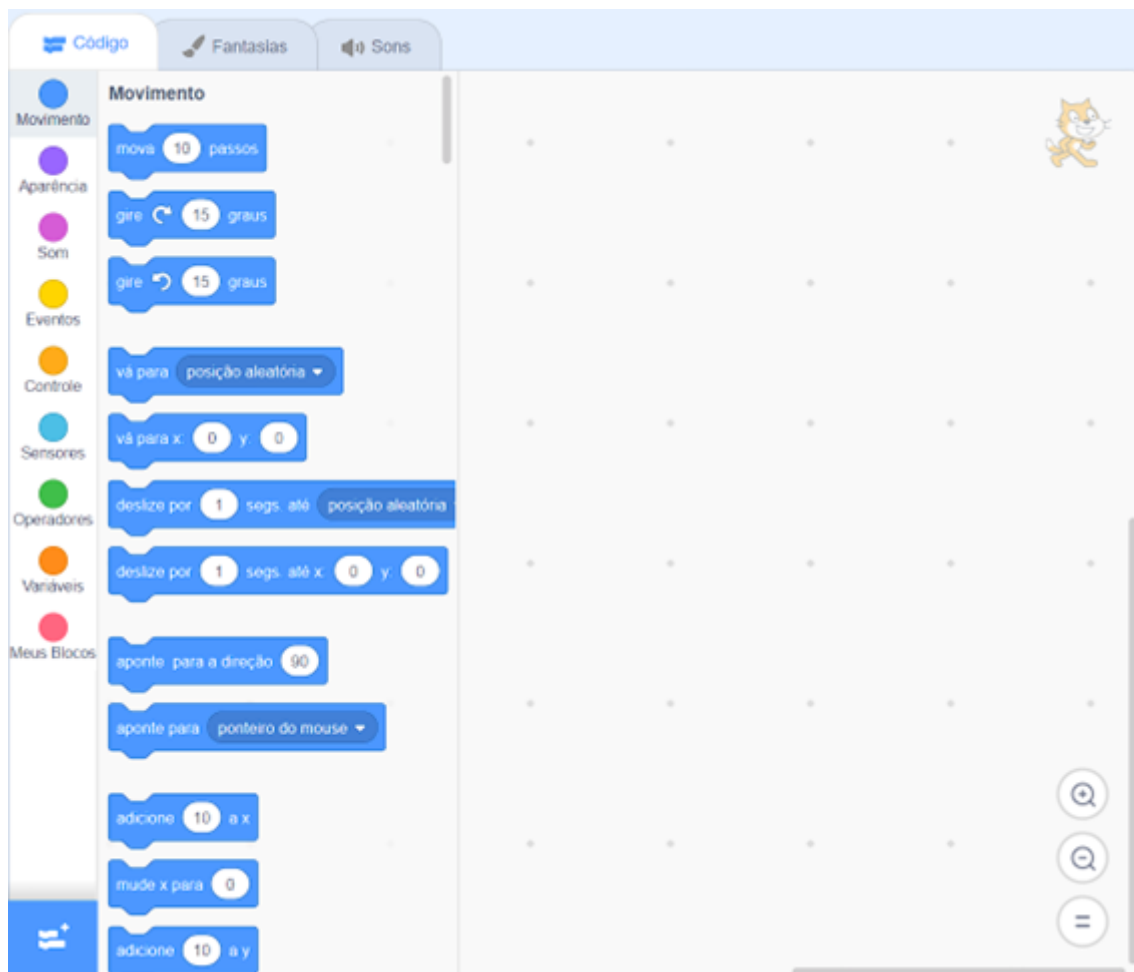
Fonte: Autores, 2021.

Figura 4: Janela dos atores



Fonte: Autores, 2021.

Figura 5: Janela de programação: Aba Código



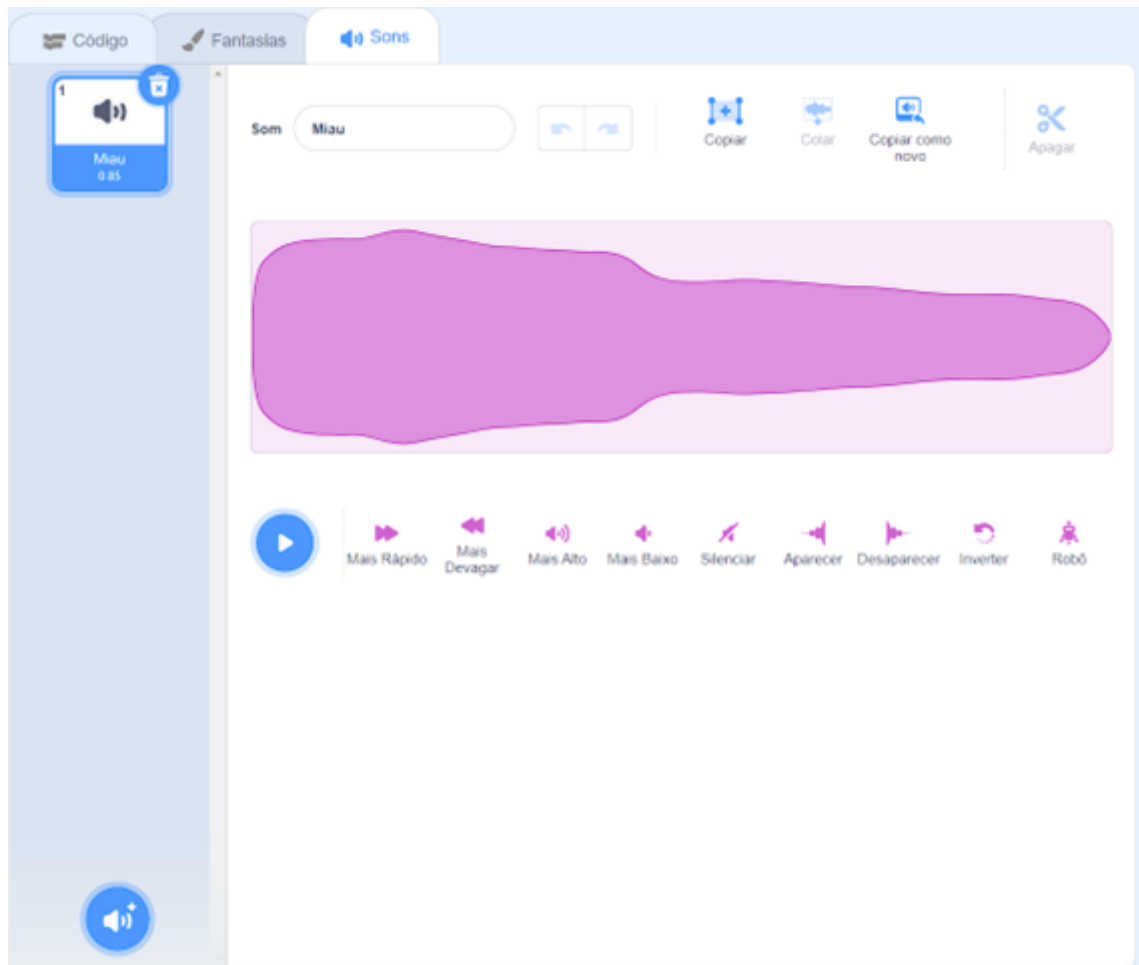
Fonte: Autores, 2021.

Figura 6: Janela de programação: Aba Fantasias



Fonte: Autores, 2021.

Figura 7: Janela de programação: Aba Sons



Fonte: Autores, 2021.

Com uma interface muito amigável, o processo de programação com o *Scratch* é bastante intuitivo. Os blocos de programação são divididos em nove categorias que possuem cores distintas e formatos peculiares, além de possuir 11 extensões que adicionam funções e possibilidades mais complexas como detecção de vídeo, narrador de texto, interação com robótica etc. Os blocos do *Scratch* servem justamente para a programação dos algoritmos dos atores do projeto. Cada ator terá seu próprio *script* de programação. Para criar um algoritmo, basta clicar, arrastar e empilhar os blocos para o centro da tela da aba código (figura 10).

Abordaremos a seguir um pouco de cada categoria principal em uma ordem lógica. As principais categorias são: movimento, aparência, som, eventos, controle, sensores, operadores, variáveis e meus blocos, conforme quadro 2 a seguir.

Quadro 2: Categoria dos blocos de programação

<p><b>Eventos (veja mais detalhes na Figura 8):</b> Categoria destinada a blocos circunstanciais. Todo algoritmo da programação irá iniciar com algum bloco dessa categoria. Além da cor amarela, esse bloco também se diferencia pela parte superior arredondada, impedindo de se colocar qualquer outro bloco acima.</p>
<p><b>Controle (veja mais detalhes na Figura 8):</b> Categoria destinada a blocos de controle. São os blocos responsáveis por manipular as ações do ator. Além da cor laranja, os blocos de controle destacam-se pela abertura contida neles. Essa abertura serve justamente para o encaixe de ações que se deseje manipular através dos blocos “sensores” ou “operadores”.</p>
<p><b>Sensores (veja mais detalhes na Figura 8):</b> Categoria destinada a auxiliar os blocos de controle. Estes blocos servirão como parâmetro para os blocos de controle. Além da cor azul, os blocos de sensores apresentam modelos de blocos que terminam com cantos arredondados ou triangulares. Os blocos com cantos arredondados são destinados a variáveis, e os triangulares ao situacional.</p>
<p><b>Operadores (veja mais detalhes na Figura 9):</b> Categoria destinada a auxiliar os blocos de controle, porém com intuito operatório matemático. Essa categoria serve justamente para as operações matemáticas, seja auxiliando blocos de controle ou efetuando uma operação com variáveis. São todos blocos das cores verde, e é com eles que criamos nossas funções matemáticas para todos os projetos.</p>
<p><b>Movimento (veja mais detalhes na Figura 9):</b> Junto com as categorias aparência e som, é responsável pelas ações do ator; para ser mais claro, as ações relacionadas à locomoção e rotação. Os blocos são da cor azul-marinho com aberturas circulares para o uso de variáveis e blocos auxiliares.</p>
<p><b>Aparência (veja mais detalhes na Figura 9):</b> Junto com as categorias movimento e som, é responsável pelas ações do ator; para ser mais claro, as ações relacionadas à aparência do ator, podendo alterar a fantasia do ator, ou até mesmo seu tamanho e cores. Os blocos são da cor roxa com aberturas circulares para o uso de variáveis e blocos auxiliares.</p>
<p><b>Som (veja mais detalhes na Figura 10):</b> Junto com as categorias movimento e aparência, é responsável pelas ações do ator, para ser mais claro, as ações relacionadas aos sons do ator, podendo tocar um som específico, carregar da memória do dispositivo, gravar e ainda aumentar ou diminuir o volume do som escolhido. Os blocos são da cor rosa com aberturas circulares para o uso de variáveis e blocos auxiliares.</p>



**Variáveis (veja mais detalhes na Figura 10):** Categoria responsável pela criação de variáveis, assim como suas alterações.

As variáveis são responsáveis pela versatilidade do nosso projeto. Com elas são possíveis manipular ações de forma dinâmica, ao contrário do uso de constante que não possibilitaria.

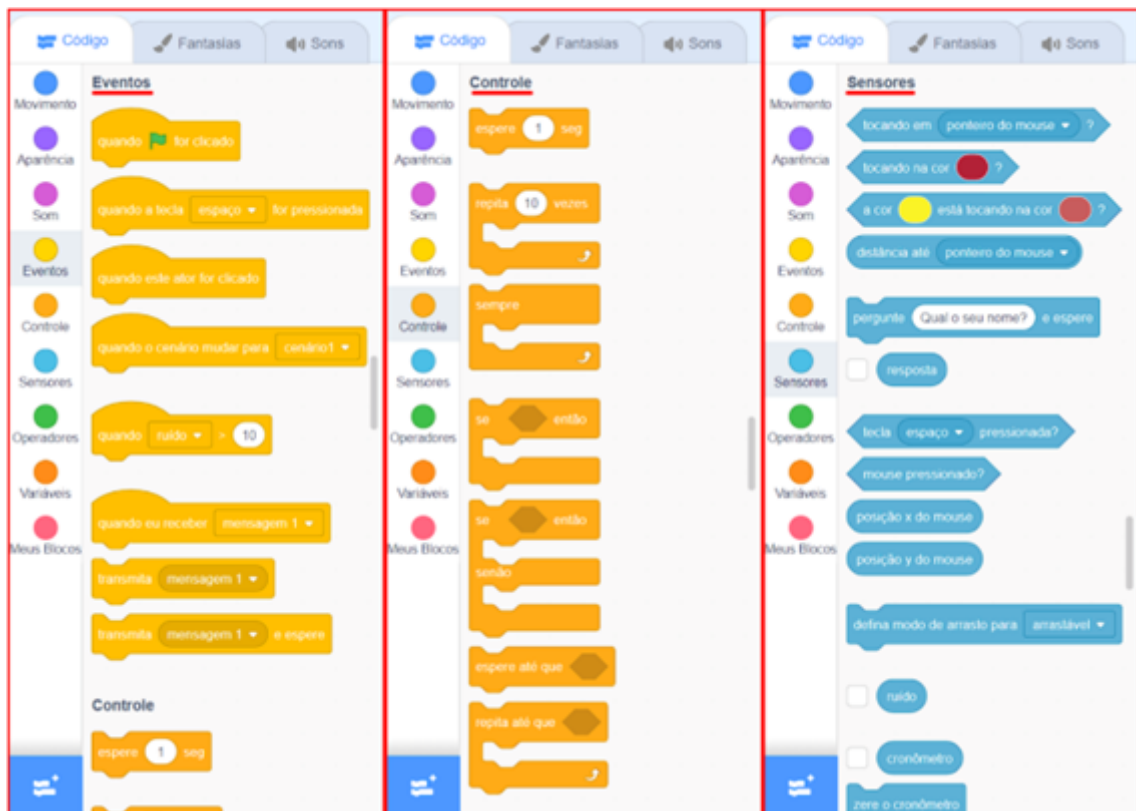
Os blocos dessa categoria são da cor laranja.

**Meus Blocos (veja mais detalhes na Figura 10):** Categoria responsável pela criação de novos blocos a partir dos anteriores. Basta nomear um bloco e adicionar os blocos desejados das outras categorias à sua definição. Sempre que você usar o seu bloco criado, ele irá executar todos os blocos que foram adicionados à sua definição.

Os blocos criados serão todos da cor vermelha

Fonte: Autores, 2021

Figura 8: Categoria dos blocos: Eventos, Controle e Sensores



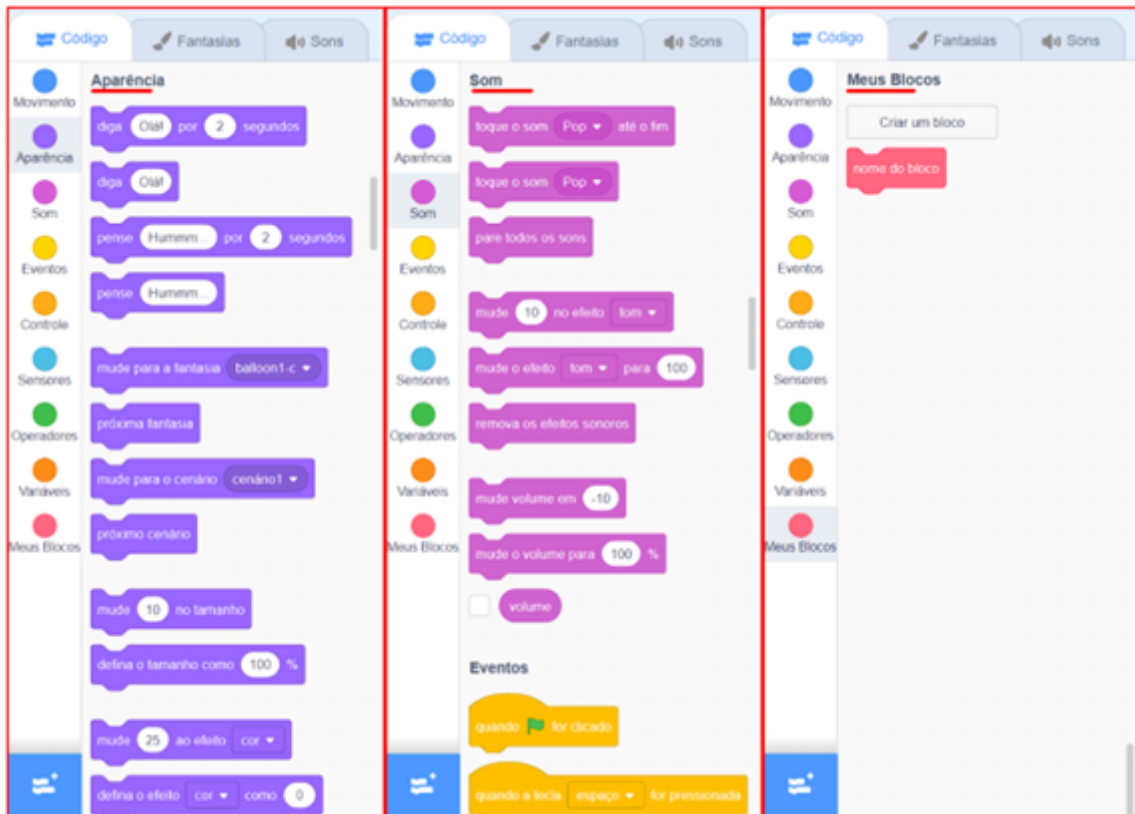
Fonte: Autores, 2021.

Figura 9: Categoria dos blocos: Operadores, Variáveis e Movimento



Fonte: Autores, 2021.

Figura 10: Categoria dos blocos: Aparência, Som e Meus Blocos



Fonte: Autores, 2021.

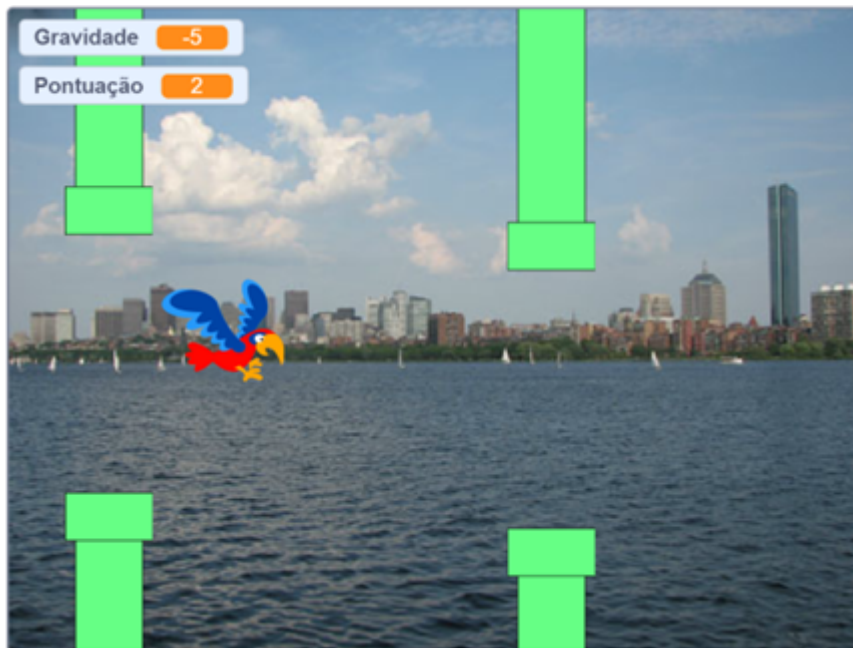
Como todos os blocos são coloridos e em forma de quebra-cabeça, pouco tempo de uso é suficiente para entender como as peças encaixam-se e relacionam-se, prevenindo o erro de sintaxe que ocorre frequentemente em linguagem de programação textual. Isso é um ponto extremamente positivo para a criação dos projetos no *Scratch*, é o que o diferencia de uma linguagem convencional e o que o torna tão atrativo. Por todas essas características, o *Scratch* foi a opção escolhida para trabalhar no minicurso.

## 2.2 Primeiro dia: *Flappy Bird*<sup>3</sup>

*Flappy Bird* é um jogo mobile de premissa simples com o objetivo de desviar o maior número de canos (obstáculos) sem deixar o pássaro cair. Para tal feito, basta tocar na tela ou, no nosso caso, clicar com o *mouse* para o pássaro bater as asas e voar.

<sup>3</sup>Disponível em: <<https://Scratchmit.edu/projects/557149293>>

Figura 11: Flappy Bird



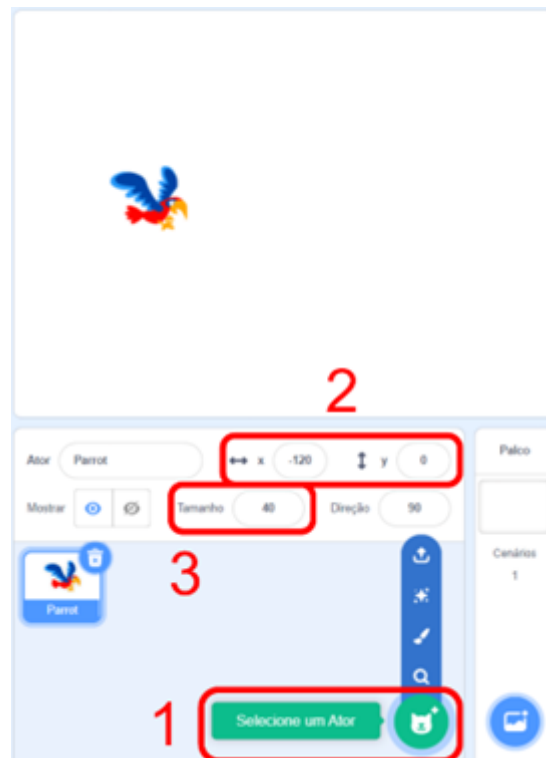
Fonte: Autores, 2021.

### Passo a passo da criação:

1. Adicionar o ator principal:

Nesse caso foi usado o ator “Parrot” disponível na própria biblioteca do Scratch e, em seguida, editamos suas coordenadas para  $(x = -120, y = 0)$  e seu tamanho para (40).

Figura 12: Adicionando o ator principal



Fonte: Autores, 2021.

## 2. Criando as variáveis:

Primeiramente é preciso criar duas variáveis. Uma chamaremos de “Gravidade”, que será responsável pela movimentação vertical com aspecto natural do “*Parrot*”, e a outra será a “Pontuação”.

Figura 13: Criando as variáveis



Fonte: Autores, 2021.

### 3. Criando os algoritmos do *Parrot*:

#### 3.1. Algoritmo do *Parrot*: Movimentação:

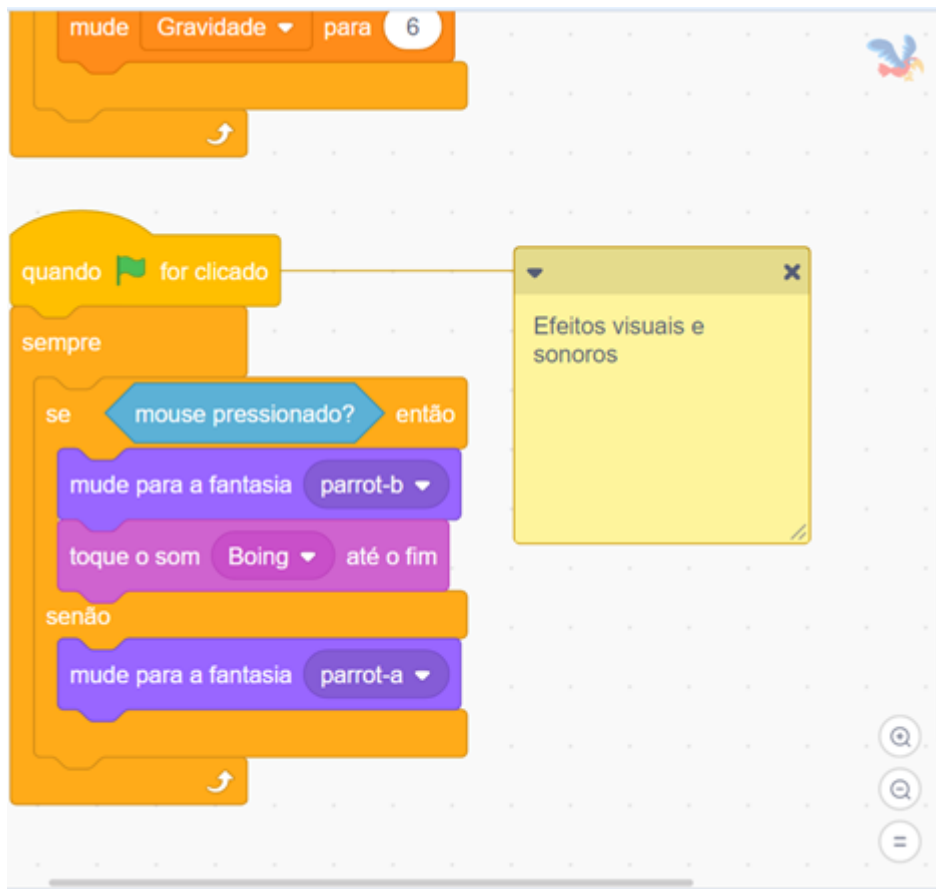
- Como todo início de algoritmo, iremos começar com um bloco da categoria “Evento”, no nosso caso, o bloco “quando bandeira verde for clicado”;
- Logo em seguida, iremos usar os blocos “mude Gravidade para (0)” e “mude Pontuação para (0)”; esse comando serve para resetar as variáveis sempre que o jogo for iniciado ou reiniciado;
- Para pôr nosso ator na posição inicial sempre que iniciar ou reiniciar o jogo, usaremos o bloco “vá para x: (-120) y: (0)”;
- Em seguida iniciaremos com o bloco “Sempre” da categoria “Controle”, responsável pela repetição dos próximos blocos;
- Para efeito de queda do *Parrot*, usaremos o bloco “adicionar (Gravidade) a y”;
- Em seguida, “adicionar (-0.5) a Gravidade”;
- Usaremos agora mais um bloco da categoria “Controle” junto de um bloco da categoria “Sensores”, o bloco “se (*mouse* pressionado?) então”;
- Dentro do “se (*mouse* pressionado?) então”, usaremos o bloco “mude Gravidade para (6)”.

Figura 14: Algoritmo do *Parrot*: Movimentação

Fonte: Autores, 2021.

### 3.2. Algoritmo do *Parrot*: Efeitos visuais e sonoros:

- Vamos iniciar novamente com o bloco “quando bandeira verde for clicado”;
- Usaremos o bloco de Controle “sempre”;
- Dentro do “sempre” usaremos o bloco “se (*mouse* pressionado?) então, senão”;
- No bloco “se (*mouse* pressionado?) então, senão” usaremos dentro do então, os blocos “mude para a fantasia (*Parrot-b*)” e “toque o som (*Boing*) até o fim” e dentro do senão. O som “Boing” foi adicionado da própria biblioteca do *Scratch*.

Figura 15: Algoritmos do *Parrot*: Efeitos visuais e sonoros:

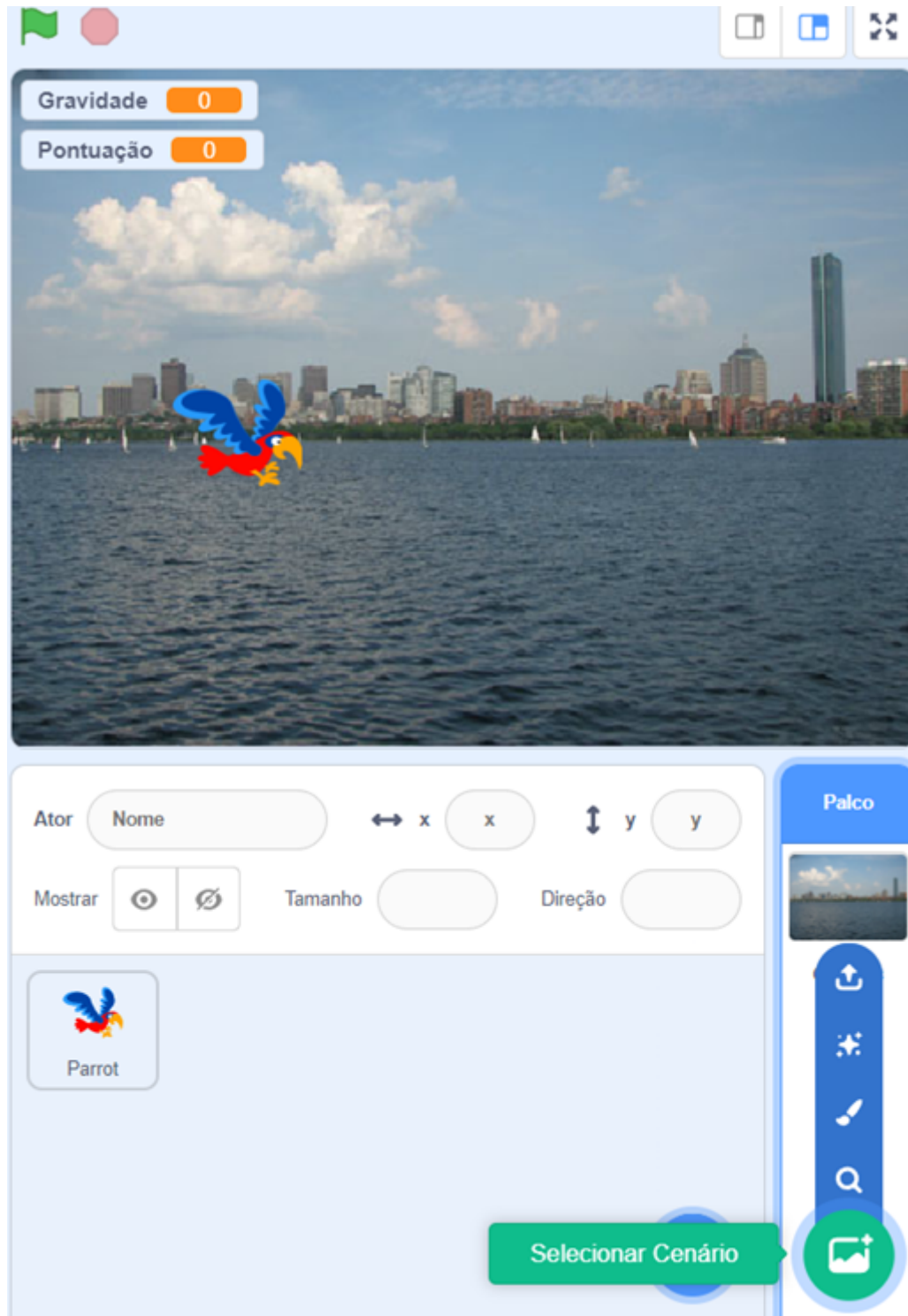
Fonte: Autores, 2021.

#### 4. Criando os obstáculos:

4.1. Antes de criar os obstáculos, colocaremos um plano de fundo, para isso, basta clicar no botão “Selecionar Cenário” e escolher um dos cenários presentes na biblioteca do *Scratch*, nesse caso, foi usado o cenário “City With Water”.



Figura 16: Selecionando o cenário

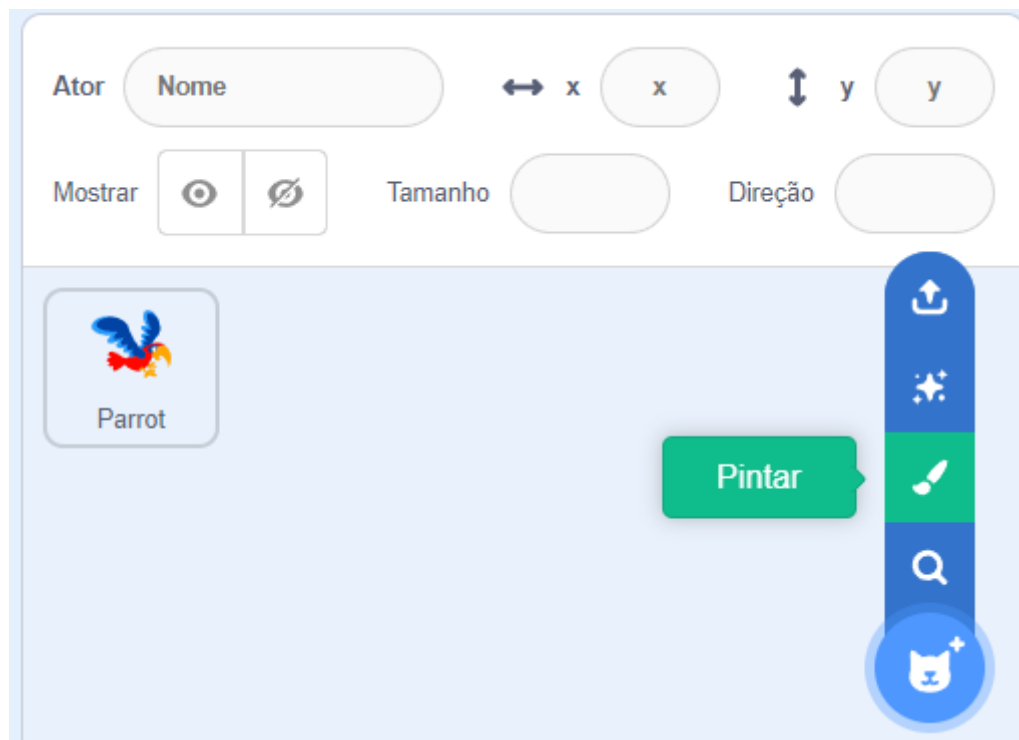


Fonte: Autores, 2021.

#### 4.2. Criando o obstáculo Cano:

- Primeiro clicamos no botão “pintar”;

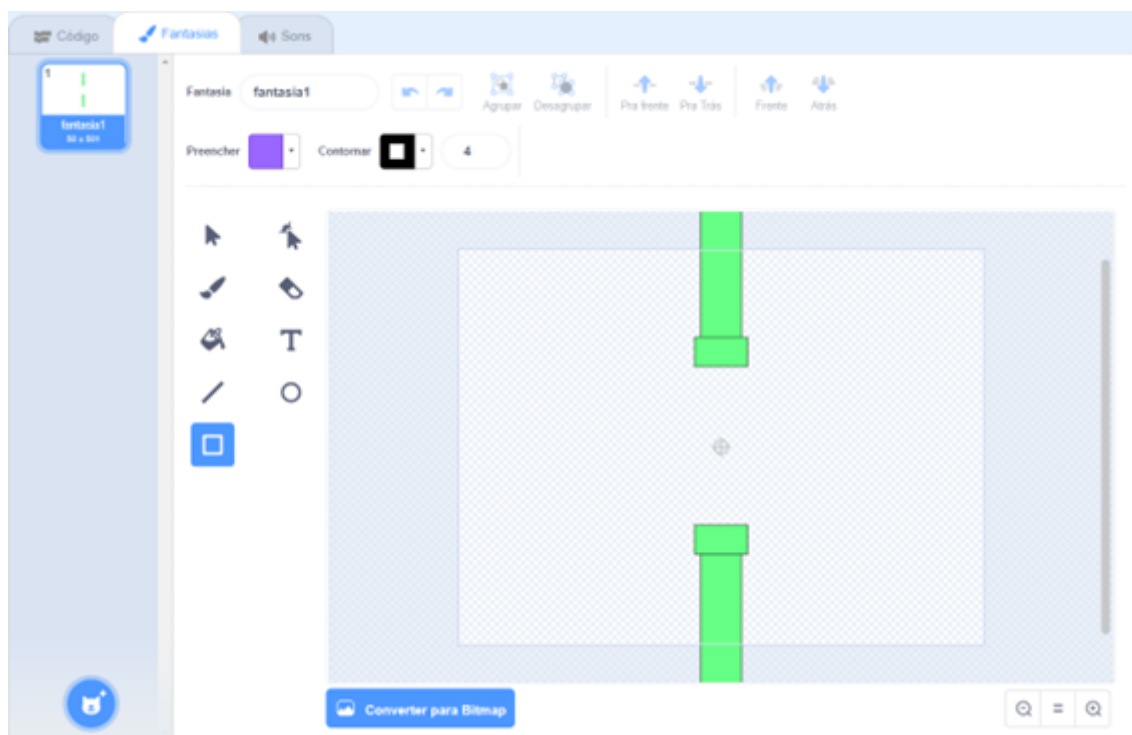
Figura 17: Botão “Pintar”



Fonte: Autores, 2021.

- Na aba “Fantasias”, usaremos a ferramenta “Retângulo” para desenhar os canos. Nesse caso foram criados 4 retângulos como mostra a imagem a seguir.

Figura 18: Desenhando os Canos



Fonte: Autores, 2021.

#### 4.3. Criando os algoritmos do ator “Cano”:

Primeiro criaremos o algoritmo de clonagem, responsável pela criação de infinitos canos:

- O algoritmo começa com o bloco “quando bandeira verde for clicado”;
- Logo após o bloco “vá para x: (0) y: (0)” para sempre iniciar na mesma posição;
- Em seguida, o bloco “esconda”, pois não usaremos o ator em si, apenas iremos usar os seus *clones*;
- Para criar infinitos *clones*, iremos utilizar o bloco “sempre”;
- Dentro do bloco “sempre”, iremos utilizar os blocos “espere (2) seg” e “crie clone de (este ator)”. O “espere (2) seg” é a frequência com que os canos irão surgir; nesse caso, de 2 em 2 segundos.

Figura 19: Algoritmo de clonagem

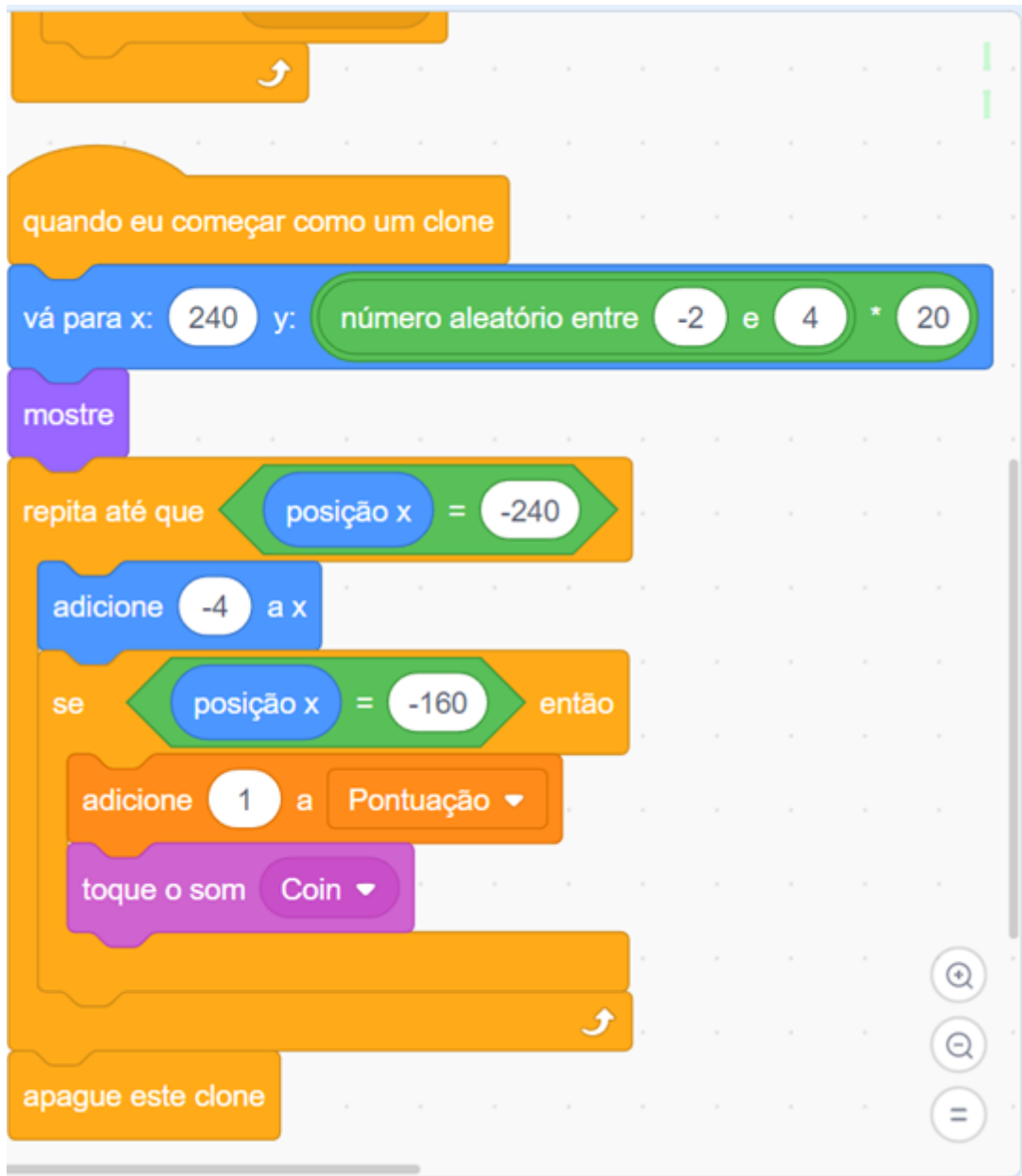


Fonte: Autores, 2021.

Em seguida, criaremos o algoritmo de movimento para os canos:

- Iremos começar com o bloco “quando eu começar como um *clone*”, que define as ações que os *clones* devem executar;
- Usaremos o bloco “vá para x: (240) y: ((número aleatório entre (-2) e (3)) \* (20))”. Esse bloco é responsável pelo *clone* começar no canto direito da tela e também em uma posição y aleatória entre -40, -20, 0, 20, 40, 60;
- Após pôr o *clone* no canto direito, usaremos o bloco “mostre” para que o *clone* do “Cano” possa ser visto;
- Usaremos o bloco “repita até que ((posição x) = (-240))” para fazer a movimentação do nosso personagem;
- Dentro do bloco “repita até que ((posição x) = (-240))” usaremos o bloco “adicione (-4) a x”, ou seja, o *clone* irá se movimentar de -4 em -4 em relação a posição x, até chegar na posição -240, que se refere ao canto direito da tela;
- Ainda dentro do bloco “repita até que ((posição x) = (-240))” colocaremos também o bloco “se ((posição x) = (-160)) então” e dentro dele os blocos “adicione (1) a (Pontuação)” e “toque o som (Coin)”, esses blocos adicionarão 1 ponto e tocarão o som “Coin” cada vez que o *clone* ultrapasse o ator *Parrot*.

Figura 20: Algoritmo de movimento para os clones



Fonte: Autores, 2021.

##### 5. Algoritmo de Fim de Jogo:

Para finalizar o jogo, iremos voltar ao ator *Parrot* e criaremos mais um algoritmo para ele, o algoritmo responsável pelo fim do jogo:

- Começaremos com o bloco “quando bandeira verde for clicado”;
- E logo após o bloco “sempre”;
- Dentro do bloco “sempre”, colocaremos o bloco “se ((tocando em (borda)) ou (tocando em (Cano))) então” e dentro do bloco “se ((tocando em (borda)) ou (tocando em (Cano))) então” colocaremos os blocos “diga (Fim de jogo)”, depois o bloco “to-

que o som (Bonk) até o fim” e por fim o bloco “pare (todos)” que é responsável por todos os outros algoritmos, assim finalizando o jogo e só permitindo jogar novamente quando clicado na bandeira verde.

Figura 21: Algoritmo de Fim de jogo



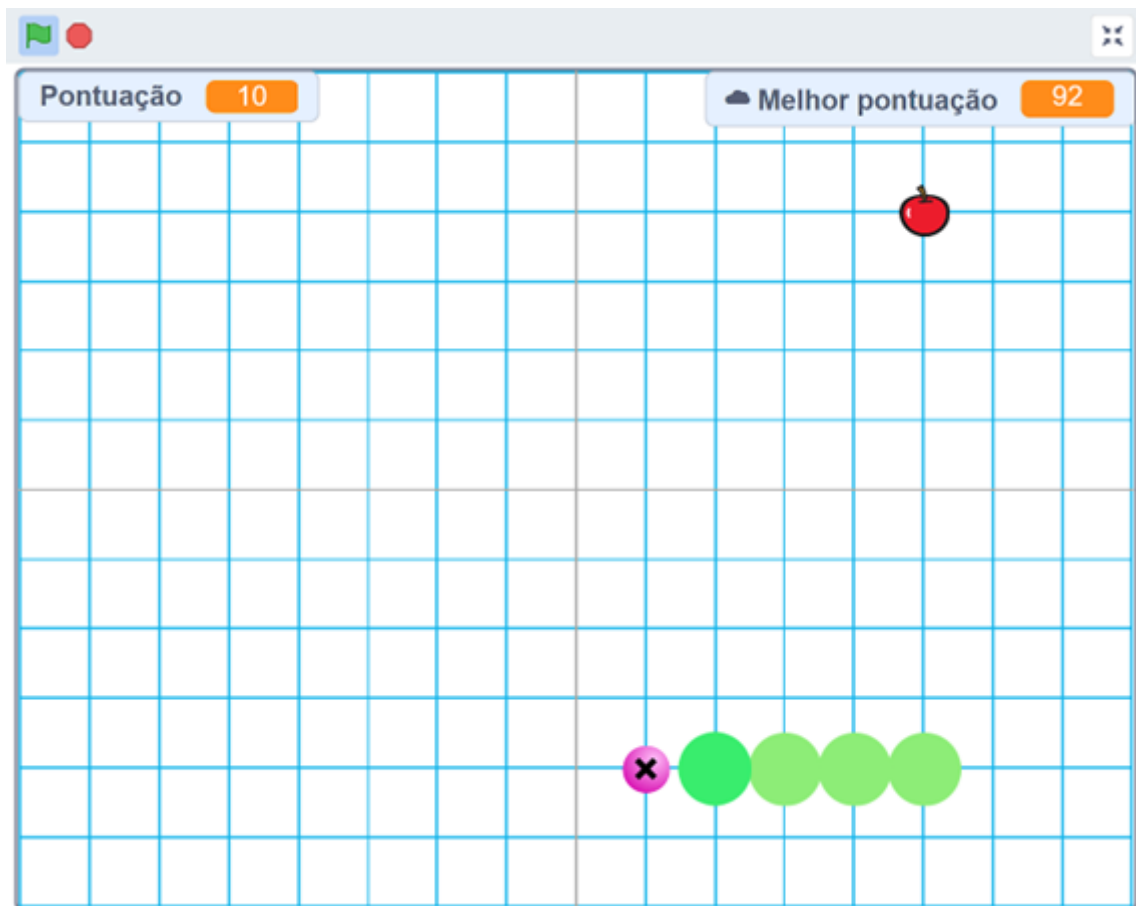
Fonte: Autores, 2021.

Como o *Scratch* é uma plataforma voltada para crianças e adolescentes, sua linguagem (uso de blocos predefinidos) é extremamente simples e atrativa, podendo com isso ser utilizada com alunos da educação básica, seja para o desenvolvimento de projetos junto dos alunos ou de projetos próprios que possam ser utilizados em sala de aula, como jogos matemáticos.

## 2.3 Segundo dia: *Snake* com operações<sup>4</sup>

A ideia do segundo dia foi trazer um jogo bastante conhecido por todos, popularmente conhecido no Brasil como “jogo da cobrinha”. O objetivo do jogo é comer o maior número de maçãs para obter a maior pontuação enquanto a cobra aumenta de tamanho sem bater nas bordas ou no próprio corpo. Apesar da programação desse jogo envolver bastante conhecimento matemático, a ideia nesse segundo momento foi demonstrar como modificar os jogos para incluir elementos matemáticos ativos, em que, ao jogar, o aluno se veja desafiado também em relação ao conhecimento matemático.

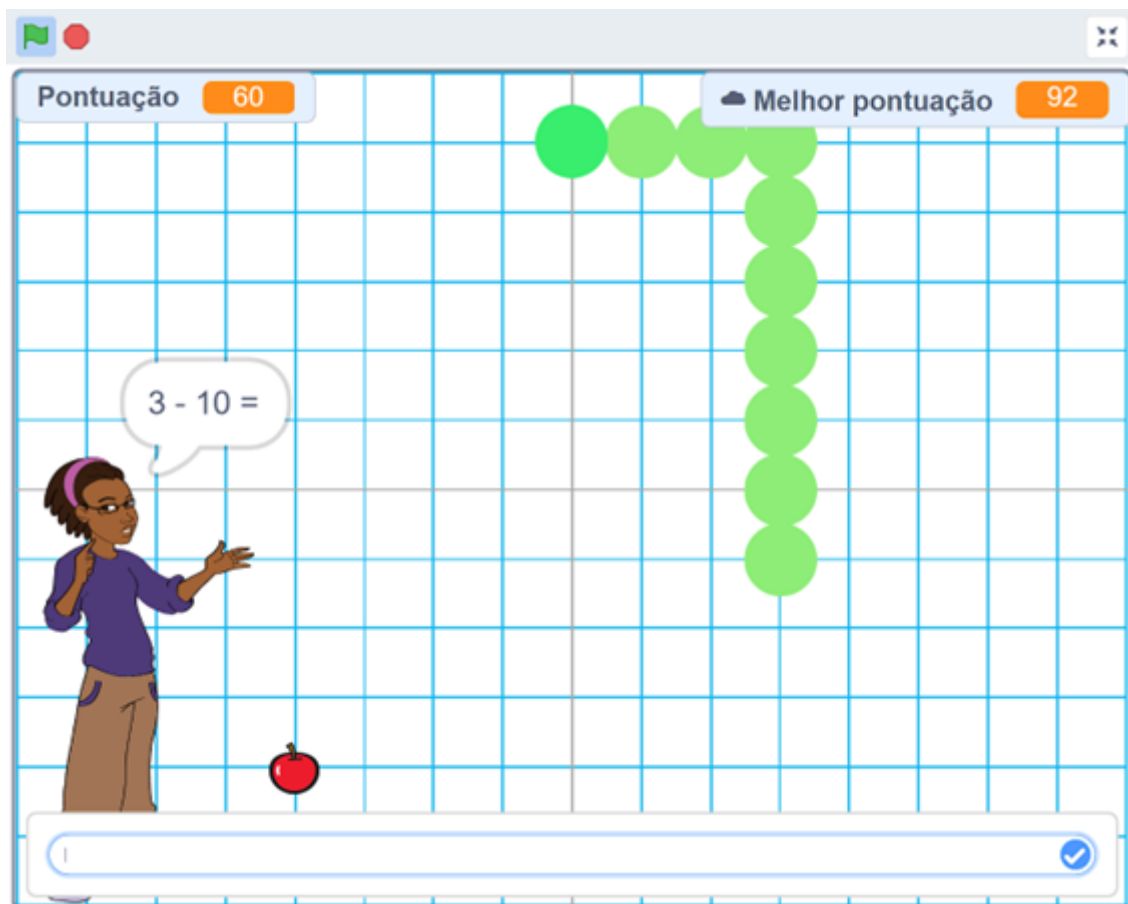
Figura 22: *Snake* com operações



Fonte: Autores, 2021.

Nesse caso, foi acrescentada uma esfera especial que aparece toda vez que o jogador coleta 5 maçãs. Caso o jogador colete a esfera, irá aparecer uma pergunta em relação à operação presente na esfera, podendo ser uma soma, subtração ou multiplicação. Sempre que o jogador acerta a pergunta, uma pontuação extra é adicionada à sua pontuação.

<sup>4</sup>Disponível em: <<https://Scratchmit.edu/projects/558493247>>.

Figura 23: *Snake* com operações: Atriz Avery

Fonte: Autores, 2021.

As etapas para a programação do projeto “*Snake* com operações” através do *Scratch* serão as seguintes:

- Criar o ator principal (a serpente): o ator principal será dividido em dois atores, a “Cabeça” e o “Corpo”;
- Criar o ator Maçã;
- Criar o ator de ligação com elemento matemático (Esfera);
- Criar a atriz de exposição do elemento matemático (Avery);
- Criar os algoritmos da Cabeça;
- Criar os algoritmos do Corpo;
- Criar os algoritmos da Maçã;
- Criar os algoritmos da Esfera;
- Criar os algoritmos da Avery;



Todos os recursos necessários para os projetos foram tirados da própria biblioteca disponível no *Scratch* ou criados com a própria ferramenta de criação do *Scratch*.

# Capítulo 3

## Considerações finais



---

A partir da experiência do minicurso, foi possível perceber que a programação com linguagem visual se mostrou extremamente proveitosa no que se refere ao ensino de matemática, capaz até mesmo de atrair e encantar os adultos. Foi nítido o interesse assim como os ganhos que os participantes tiveram pelo uso da programação como ferramenta de ensino, ainda que os encontros do minicurso tenham ocorrido por videoconferência.

Para que mais docentes tenham acesso a esse conhecimento e aos conteúdos que trabalhamos no minicurso, escrevemos este *e-book*. Esperamos contribuir para que mais professores possam utilizar a plataforma *Scratch* em suas aulas, colhendo os benefícios que a mesma proporciona não apenas para a aprendizagem de conteúdos matemáticos como também para a aprendizagem da lógica de programação. Os jogos digitais e a programação estão cada vez mais presentes no cotidiano dos nossos estudantes e é importante que tenhamos recursos para utilizar esse interesse a nosso favor nas salas de aula.

# Referências Bibliográficas



BRASIL (Ministério da Educação). **Base Nacional Comum Curricular**: educação é a base. Brasília, 2018. Disponível em: <<http://basenacionalcomum.mec.gov.br>>. Acesso em: 23 de Jul. 2021. [2](#)

BRASIL (Ministério da Educação). **Parâmetros Curriculares Nacionais do Ensino Médio**. Brasília, 2020. [2](#)

CONCEIÇÃO, D. B. d.; MENDES, A. A.; BORGES, L. H. d. F. Análise dos fatores que desmotivam/desinteressam os alunos com relação à matemática. In: SEMINÁRIO CIENTÍFICO DO UNIFACIG, 1., 2015, Manhuaçu. **Anais [...]**. Manhuaçu: UNIFACIG, 2015. [2](#)

MOTA, F. P.; RIBEIRO, N. F. A.; EMMENDORFER, L.; BUTZEN, P.; MACHADO, K. S.; ADAMATTI, D. F. Desenvolvendo o Raciocínio Lógico no Ensino Médio: uma proposta utilizando a ferramenta *Scratch*. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 25., 2014. **Anais [...]**. Porto Alegre: SBC, 2014. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/2964>>. Acesso em: 20 de jul de 2021. [6](#)

PANORAMA. **Crianças e smartphones no Brasil**. 2020. Disponível em: <<https://www.mobiletime.com.br/pesquisas/criancas-e-smartphones-no-brasil-outubro-de-2020/>>. Acesso em: 24 de Jul. 2021.

PANORAMA. **Uso de Apps no Brasil**. 2021. Disponível em: <<https://www.mobiletime.com.br/pesquisas/uso-de-apps-no-brasil-junho-de-2021//>>. Acesso em: 24 de Jul. 2021.

SHIMOHARA, C.; SOBREIRA, E. Criando Jogos Digitais para a aprendizagem de matemática no ensino fundamental I. In: WORKSHOP DE INFORMÁTICA NA ESCOLA, 21., 2015. **Anais [...]**. Porto Alegre: SBC, 2015. [5](#)



## 2º Simpósio da Formação do Professor de Matemática da Região Centro-Oeste

Realização e Organização



Associação Nacional dos Professores de Matemática na Educação Básica

ISBN: 978-65-88013-20-5

CRL



9 786588 013205