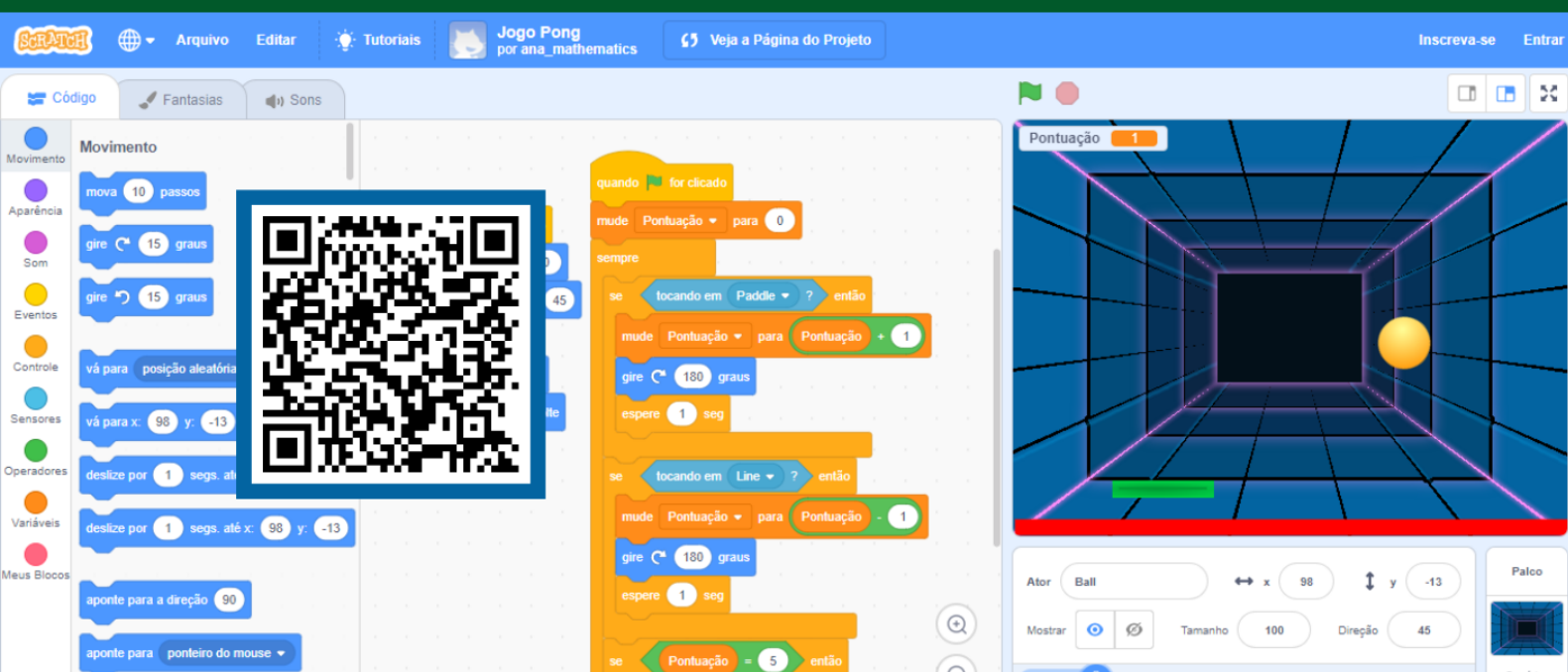


PRODUTO EDUCACIONAL

GUIA DIDÁTICO-PEDAGÓGICO

ANA CLARA DIAS BERNARDINO
ROSANA MARIA MENDES

A UTILIZAÇÃO DO SOFTWARE SCRATCH NA PERSPECTIVA DA EDUCAÇÃO MATEMÁTICA INCLUSIVA



ppgecem

PROGRAMA DE PÓS-GRADUAÇÃO EM ENSINO DE CIÊNCIAS E EDUCAÇÃO MATEMÁTICA



UNIVERSIDADE FEDERAL DE LAVRAS

**A UTILIZAÇÃO DO *SOFTWARE*
SCRATCH NA PERSPECTIVA DA
EDUCAÇÃO MATEMÁTICA INCLUSIVA**

A UTILIZAÇÃO DO *SOFTWARE* SCRATCH NA EDUCAÇÃO MATEMÁTICA

Ana Clara Dias Bernardino
Rosana Maria Mendes

Copyright © dos autores

Todos os direitos garantidos. Qualquer parte desta obra pode ser reproduzida, transmitida ou arquivada desde que levados em conta os direitos dos autores.

Ficha catalográfica elaborada pela Coordenadoria de Desenvolvimento do Acervo da Biblioteca Universitária da UFLA

Bernardino, Ana Clara Dias.

A utilização do software scratch na perspectiva da educação matemática inclusiva [recurso eletrônico] / Ana Clara Dias Bernardino, Rosana Maria Mendes. – Lavras: PPGECEM/UFLA, 2024.

1 recurso online (66 p.) : il. color.

Modo de acesso: <http://repositorio.ufla.br/handle/1/56524>

Publicação digital (e-book) no formato PDF.

ISBN:978-65-84982-12-3

1. Programação em blocos. 2. Desenvolvimento de jogos. 3. Inclusão. 4. Educação - Matemática Inclusiva. 5. Game development. I. Mendes, Rosana Maria. II. Título.

CDD - 370.71

Bibliotecária: Defátima Aparecida Silva Pessoa - CRB6/1496

Coordenador da Coleção de e-books *Práticas Pedagógicas e Formação Docente*:

José Antônio Araújo Andrade

Editor responsável:

José Antônio Araújo Andrade

Revisão:

Natália Rodrigues Silva do Nascimento

Capa:

Ana Clara Dias Bernardino, Rosana Maria Mendes e José Antônio Araújo Andrade

Diagramação:

José Antônio Araújo Andrade



Coleção de e-books Práticas Pedagógicas e Formação Docente

José Antônio Araújo Andrade

Marianna Meirelles Junqueira

Iraziet da Cunha Charret

Conselho Editorial

Dra. Adair Mendes Nacarato – Universidade São Francisco – Brasil

Dra. Adriana Aparecida Molina Gomes – Universidade Federal do Mato Grosso do Sul – Brasil

Dra. Adriana Correia de Almeida – Instituto Federal do Sul de Minas – Brasil

Dra. Cármen Lúcia Brancaglioni Passos – Universidade Federal de São Carlos – Brasil

Dra. Cristina Carvalho de Almeida – Instituto Federal do Sul de Minas – Brasil

Dr. Evandro Fortes Rozentalski – Universidade Federal de Itajubá – Brasil

Dra. Flávia Cristina Figueiredo Coura – Universidade Federal de São João Del Rei – Brasil

Dra. Francine de Paulo Martins Lima – Universidade Federal de Lavras – Brasil

Dr. Frederico Augusto Totti – Universidade Federal de Alfenas – Brasil

Dr. Gildo Giroto Junior – Universidade Estadual de Campinas – Brasil

Dra. Iraziet da Cunha Charret – Universidade Federal de Lavras – Brasil

Dr. João Pedro da Ponte – Universidade de Lisboa – Portugal

Dr. José Antônio Araújo Andrade – Universidade Federal de Lavras – Brasil

Dra. Leonor Santos – Universidade de Lisboa – Portugal

Dr. Luciano Fernandes Silva – Universidade Federal de Itajubá – Brasil

Dra. Maria do Carmo de Sousa – Universidade Federal de São Carlos – Brasil

Dra. Marianna Meirelles Junqueira – Universidade Federal de Lavras – Brasil

Dr. Regilson Maciel Borges – Universidade Federal de Lavras – Brasil

Dra. Regina Célia Grando – Universidade Federal de Santa Catarina – Brasil

Dr. Ronei Ximenes Martins – Universidade Federal de Lavras – Brasil

Dr. Vitor Fabrício Machado Souza – Universidade Federal do Paraná – Brasil

Dr. Wilson Elmer Nascimento – Universidade Federal do Rio Grande do Norte – Brasil

SUMÁRIO

1	APRESENTAÇÃO.....	9
2	EDUCAÇÃO MATEMÁTICA INCLUSIVA.....	13
	2.1 Cenário para investigação inclusivo	14
3	CONHECENDO O SOFTWARE SCRATCH	17
	3.1 Primeiros passos	17
	3.2 Como utilizar o Scratch on-line	18
	3.3 Como fazer o download do Scratch	19
	3.4 Ambiente de programação Scratch.....	20
	3.5 Palco	22
	3.6 Atores.....	25
4	O USO DO SCRATCH NAS AULAS DE MATEMÁTICA.....	31
	4.1 Exemplos para familiarização com o Scratch	31
	4.2 Exemplos de construção de jogos no Scratch	43
	Jogo de Capturar	43
	Jogo do Tubarão.....	44
	Jogo do Balão	49
	Jogo Poluição do Ar	52
	4.3 Exemplos de jogos para praticar.....	54
	Jogo Pong	54
	Jogo “Labirinto: Recolhendo o lixo”	55
5	COMO UTILIZAR O SCRATCH NA PERSPECTIVA INCLUSIVA.....	57
	5.1 Estudantes de diferentes níveis de ensino.....	57
	5.2 Estudantes Surdas e Surdos	58
	5.3 Estudantes com Deficiência intelectual.....	58
	5.4 Estudantes Cegas e Cegos	59
6	CONSIDERAÇÕES FINAIS	63
	REFERÊNCIAS BIBLIOGRÁFICAS	65



APRESENTAÇÃO

Prezada professora e prezado professor,

Neste Produto Educacional, você encontrará sugestões de atividades práticas que podem auxiliar no desenvolvimento de jogos, histórias e animações em suas aulas utilizando o *software* Scratch. Essas sugestões são fruto de estudos e diálogos entre docentes que planejaram, experimentaram, vivenciaram e refletiram sobre a utilização do Scratch para o processo de ensinar e aprender Matemática na perspectiva inclusiva.

O Scratch é um ambiente de programação que permite criar histórias interativas, jogos e animações, de maneira simples, lúdica e intuitiva. No processo de aprender a programar, além de aprender ideias matemáticas e computacionais, também podemos aprender estratégias para a resolução de problemas, a elaboração de projetos e a comunicação de ideias, que são habilidades úteis não apenas para cientistas da computação, mas para todas as pessoas, independentemente da idade, interesses ou ocupação (RESNICK, 2013). Como utilizar esse *software* em sala de aula do ponto de vista da Educação Matemática Inclusiva? Pensando nessas questões, elaboramos este material de apoio destinado às professoras e aos professores que pretendem utilizar o Scratch em suas aulas em uma perspectiva inclusiva, buscando contribuir para que esse *software* seja acessível a um público cada vez mais amplo.

Este Produto Educacional foi desenvolvido a partir dos resultados da dissertação de mestrado intitulada “A Educação Matemática Inclusiva e o *software* Scratch: percepções de docentes em uma prática formativa”, no âmbito do Programa de Pós-Graduação em Ensino de Ciências e Educação Matemática, da Universidade Federal de Lavras. A pesquisa foi realizada com um enfoque qualitativo, utilizando a metodologia de Análise de Conteúdo, cujos dados foram construídos em um curso

denominado “Introdução ao *software* Scratch na perspectiva da Educação Matemática Inclusiva”. Esse curso ocorreu exclusivamente de forma remota e teve como participantes docentes que lecionavam Matemática em escolas públicas de cidades do sul de Minas Gerais, no Ensino Fundamental e Médio.

Naquele curso, criamos um espaço de formação compartilhada e discutimos o uso desse *software* para mobilizar conhecimentos matemáticos e computacionais utilizando o raciocínio lógico para criar estratégias de resolução de problemas de maneira autônoma, colaborativa e criativa, na perspectiva inclusiva. Diversas pesquisas apontaram as potencialidades desse *software* para o processo de ensino e aprendizagem de Matemática e de outras disciplinas. Porém, identificamos a necessidade de investigar sobre o uso do Scratch na Educação Matemática Inclusiva, devido à escassez de trabalhos com essa temática. Nesse contexto, durante o curso, buscamos respostas para questionamentos como estes: *É possível utilizar o Scratch em sala de aula na perspectiva da Educação Matemática Inclusiva? De que maneiras? Que adaptações são necessárias?* Trazemos, neste material, resultados dessa investigação.

O que entendemos por Educação Matemática Inclusiva? Orientamo-nos por ambas as interpretações apontadas por Skovsmose (2019) para a Educação Inclusiva e Educação Matemática Inclusiva, considerando que existem conexões estreitas entre elas. Uma delas é bastante específica e predominante no Brasil: refere-se aos desafios de incluir na sala de aula convencional estudantes com deficiência visual, Cegas e Cegos, deficiência auditiva, Surdas e Surdos, com transtorno do espectro autista, com deficiência intelectual, entre outras deficiências. A outra trata-se de uma interpretação mais geral, por meio da qual pode-se considerar que a Educação Inclusiva e Educação Matemática Inclusiva se referem, por exemplo, à inclusão de estudantes com diferentes origens culturais no mesmo contexto educacional. Nessa perspectiva, concebemos a Educação Matemática Inclusiva num sentido amplo, ao considerarmos que todas as estudantes e todos os estudantes são capazes de aprender, embora aprendam de maneiras diferentes, sejam com, sejam sem deficiências.

Neste material, apresentamos as principais ferramentas do Scratch, para que você conheça a estrutura do *software* e possa aprender a utilizá-lo. Em seguida, apresentamos tarefas que podem ser realizadas para praticar os conhecimentos

relacionados à programação, enquanto reconhece as situações-problema e elabora estratégias buscando resolvê-las. Ressaltamos que, embora haja alguns comandos em comum, as tarefas propostas poderão ser resolvidas de maneiras diferentes e cada pessoa poderá utilizar suas próprias estratégias para atingir os objetivos. Apresentamos tarefas mais específicas com objetivos pré-determinados e tarefas que possibilitam que cada estudante tenha liberdade para escolher que tipo de projeto (jogos, histórias interativas e animações) irá desenvolver. Desse modo, sugerimos que essa abordagem também seja utilizada por docentes em suas aulas, proporcionando oportunidades para que estudantes sejam criativos e tenham uma postura mais ativa no processo de construção do conhecimento.

Trazemos, ainda, sugestões que consideramos importantes para auxiliar a utilizar esse *software* com o objetivo de possibilitar a inclusão de todas as estudantes e todos os estudantes durante as aulas. Não há uma fórmula única para utilizar o Scratch na perspectiva inclusiva e ainda há desafios a serem superados para atingir esse objetivo, mas compartilhamos, neste Produto Educacional, possíveis caminhos que podem servir de inspiração para professoras e professores.

As tarefas e os jogos aqui apresentados poderão ser adaptados por você, de acordo com seus objetivos e considerando as características, potencialidades, limitações e necessidades das estudantes e dos estudantes que irão utilizar o Scratch em suas aulas. Nesse sentido, para planejar atividades na perspectiva inclusiva é necessário reconhecer que todas as pessoas possuem capacidades e aptidões e são capazes de aprender, mas não aprendem da mesma maneira.

Nós, docentes, como mediadoras e mediadores do processo de ensino e de aprendizagem, podemos avaliar e estabelecer estratégias, atentando-nos às especificidades de cada estudante, com o objetivo de criar condições para que todas as pessoas aprendam. Damos ênfase à expressão *“criar condições para que todas as pessoas aprendam”*, ressaltando que não é possível garantir que as pessoas irão aprender, sejam estudantes com, sejam sem deficiência, sejam estudantes com, sejam sem dificuldades de aprendizagem, mas é possível encontrar meios para tornar o conhecimento acessível a um público amplo, considerando as limitações sensoriais ou físicas das estudantes e dos estudantes, considerando as diferentes formas de aprendizagem e a natureza do conteúdo que será ensinado. Afinal, essas questões

podem ser consideradas durante o planejamento das tarefas e podem auxiliar nas intervenções que serão realizadas durante as aulas.

Esperamos que este Produto Educacional contribua para sua prática e sirva de inspiração para utilizar o *software* Scratch em suas aulas com o olhar voltado para a Educação Matemática Inclusiva.

Boa leitura!



EDUCAÇÃO MATEMÁTICA INCLUSIVA

Na concepção inclusiva, estudantes com e sem deficiência permanecem na mesma sala de aula. A escola inclusiva busca a participação e o progresso de todas as pessoas, adotando novas práticas pedagógicas. Além disso, reconhece as diferenças das estudantes e dos estudantes diante do processo educativo.

Respeitar a diferença significa não subestimar as possibilidades, nem superestimar as dificuldades, mas sim adotar novas práticas pedagógicas para minimizar as desvantagens das estudantes e dos estudantes. Precisamos investir na equiparação de oportunidades e, assim, ampliar as possibilidades para que estudantes com e sem deficiência se desenvolvam, aprendam e conquistem autonomia.

Quer saber mais informações sobre o assunto? Veja as sugestões de leitura sobre Educação Inclusiva e Educação Matemática Inclusiva:

 SKOVSMOSE, Ole. Inclusões, encontros e cenários. **Educação Matemática em Revista**, Brasília, v. 24, n. 64, p. 16-32, set./dez. 2019.

 FERNANDES, Solange Hassan Ahmad Ali. Educação Matemática Inclusiva: adaptação x construção. **Revista Educação Inclusiva** - REIN, Campina Grande, PB, v. 1. 01, n. 01, p.78-95, julho/dezembro-2017.

 BRASIL. Ministério da Educação. Secretaria de Educação Especial. Política Nacional de Educação Especial na Perspectiva da Educação Inclusiva. Inclusão: revista da educação especial, v. 4, n 1, janeiro/junho 2008. Brasília: MEC/SEESP, 2008.

Compreendemos a educação inclusiva não como uma educação que tenta incluir as pessoas com deficiência na “normalidade”, mas sim como uma educação que busca ir além das diferenças e tenta estabelecer *encontros entre diferenças*¹.

Para ilustrar o que esta interpretação pode significar com relação à educação matemática inclusiva, Skovsmose (2019) menciona a possibilidade de construir materiais didáticos apropriados para cegas e cegos, que possibilitam o acesso de estudantes à matemática, mas que também podem ser utilizados por estudantes videntes, de modo que possam trabalhar em conjunto nas mesmas tarefas.

Você quer conhecer estudos que têm sido realizados no âmbito da Educação Matemática Inclusiva no Brasil?

O GT13 - Diferença, Inclusão e Educação Matemática, por exemplo, é um Grupo de Trabalho da Sociedade Brasileira de Educação Matemática (SBEM) que agrega pesquisadores cujos estudos contemplam as diferentes questões da Educação Matemática Inclusiva.

As informações referentes ao GT13 estão disponíveis no endereço eletrônico:

 [GT13 - Diferença, Inclusão e Educação Matemática](#)

2.1 Cenário para investigação inclusivo

Já ouviu falar de “cenários para investigação inclusivos”? Podemos nos fundamentar nessa perspectiva para pensar em aulas de Matemática inclusivas, utilizando o *software* Scratch ou outros recursos.

De acordo com Skovsmose (2019), cenários para investigação inclusivos possuem as seguintes características:

(1) Abrem espaço para investigações. Ao invés de especificar sequências de exercícios ou problemas a serem resolvidos, podemos convidar as estudantes e os estudantes a participarem do processo de investigação, fazendo perguntas,

¹ O termo “encontros entre diferenças” foi sugerido por Lulu Healy (FIGUEIRAS; HEALY; SKOVSMOSE, 2016).

formulando hipóteses, experimentando argumentos e ouvindo outros argumentos e ideias.

(2) Compartilham a ideia principal do design universal: viabilizar um ambiente que seja acessível a todas as pessoas. Nesses cenários, é possível encontrar uma variedade de desafios que reconhecem as diferenças entre as estudantes e os estudantes.

(3) Facilitam colaborações. As diferenças entre as estudantes e os estudantes ajudam a estabelecer processos de igualdade que assumem a forma de diálogos. Nesses processos, as próprias noções de capacidade e incapacidade ou normais e não-normais perdem significância.

Nos próximos tópicos, apresentaremos informações básicas sobre o Scratch e mostraremos sugestões de como usar esse *software* em aulas de Matemática com inspiração nas características dos cenários para investigação inclusivos.



CONHECENDO O SOFTWARE SCRATCH

O Scratch é um *software* de programação em que a digitação do código característica das linguagens de programação mais complexas é substituída pela ação de clicar, arrastar e encaixar os blocos de comando para uma área específica do programa. Com o Scratch, é possível criar histórias interativas, jogos e animações, em uma linguagem simples, lúdica e intuitiva.

A primeira versão do Scratch foi lançada em maio de 2007 e, desde então, foram realizadas diversas atualizações, correção de erros e implementação de novas funcionalidades. O Scratch é gratuito e está disponível em diversos idiomas, tendo, portanto, usuárias e usuários de diversos países.

3.1 Primeiros passos

O *software* Scratch pode ser instalado no computador (Scratch *Desktop*), o que possibilita criar e salvar projetos sem conexão com a internet. A versão mais recente do *software*, o Scratch 3.0, é compatível com os sistemas operacionais Windows 10+, macOS 10.13+, ChromeOS e Android 6.0+². Atualmente, não está disponível para Linux, mas estão verificando a possibilidade de, no futuro, ser disponibilizado, conforme informado na comunidade *on-line* do Scratch³.

O Scratch também pode ser utilizado *on-line*, por meio do acesso à comunidade do Scratch, sendo necessária a conexão com a internet para criar, salvar e compartilhar projetos.

² A versão atual do Scratch não pode ser utilizada em celulares Android, funciona apenas em *tablets* Android.

³ A comunidade *on-line* do Scratch está disponível em: <https://scratch.mit.edu/>. Acesso em: jul. 2020.

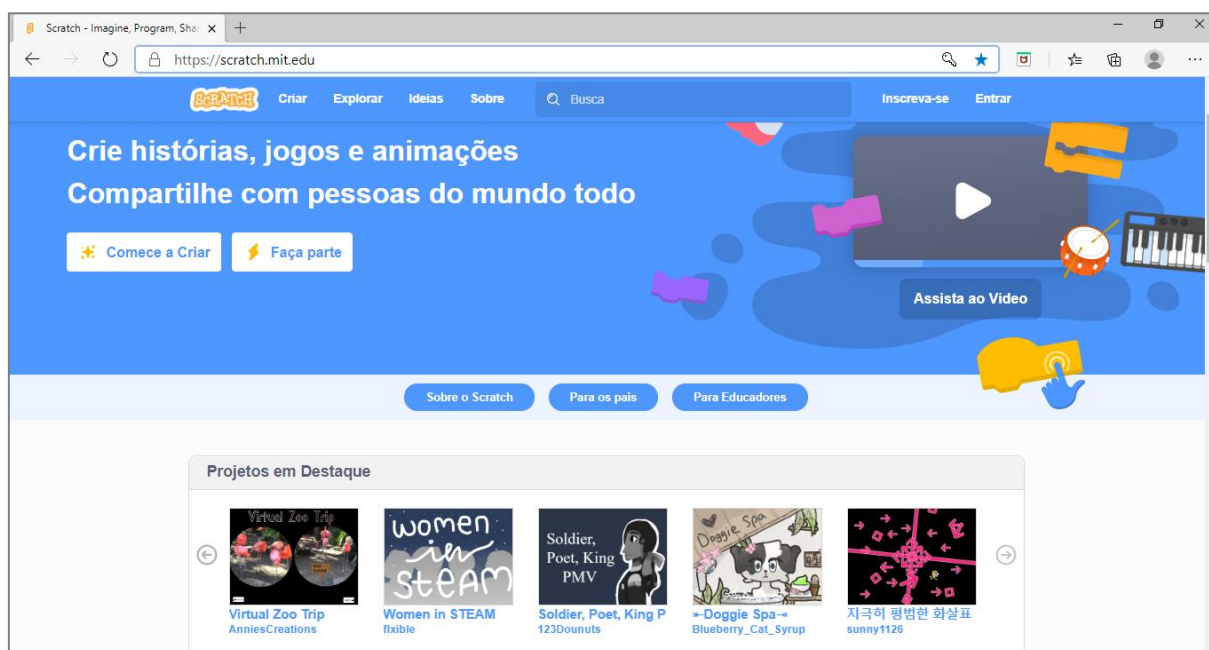
Os projetos criados no Scratch podem ser compartilhados na comunidade *on-line*. Para isso, é necessário cadastrar uma conta no *site*, clicando em “Inscreva-se” na tela inicial do *site* ou na tela do Scratch *on-line*.

A seguir, descreveremos os procedimentos necessários para a utilização do Scratch *Desktop* e do Scratch *on-line*.

3.2 Como utilizar o Scratch *on-line*

Para utilizar o Scratch *on-line*, basta acessar a comunidade do Scratch por meio do endereço <https://scratch.mit.edu/> e clicar em “Criar”:

Figura 3.1 – Captura de tela da página inicial da comunidade *on-line* do Scratch.



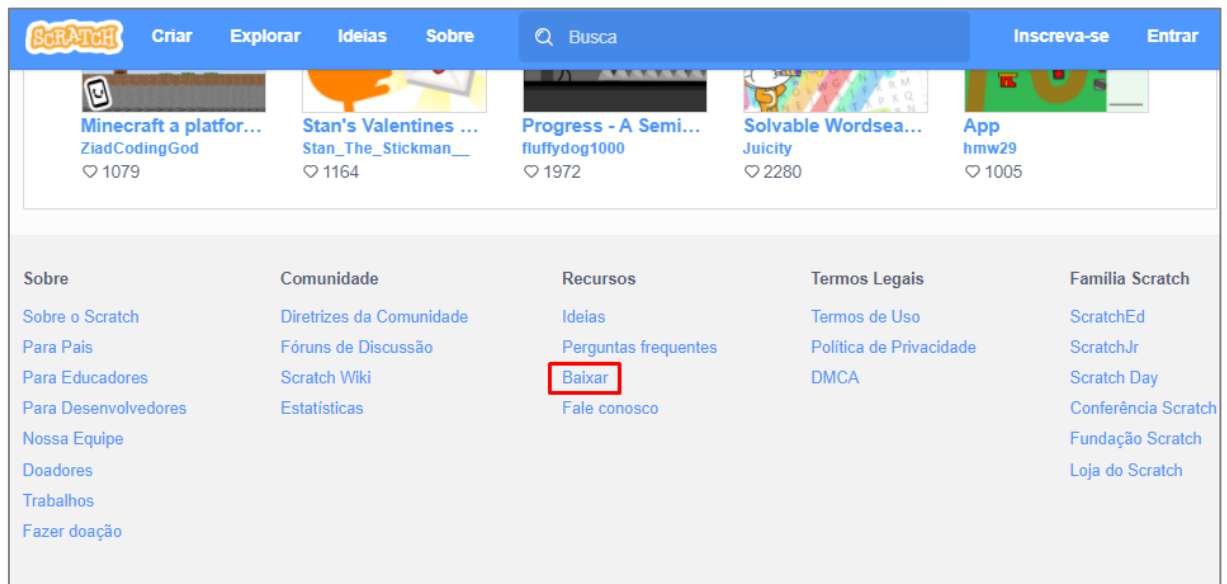
Fonte: Das autoras (2024).

Em seguida, será aberta a página do Scratch *on-line*, onde será possível criar projetos, desde que possua acesso à internet. A versão *on-line* do Scratch pode ser utilizada em qualquer sistema operacional.

3.3 Como fazer o *download* do Scratch

Para fazer o *download* do Scratch, o primeiro passo é acessar a comunidade do Scratch no endereço: <https://scratch.mit.edu/>. Em seguida, ao final da página, basta selecionar a opção “Baixar”:

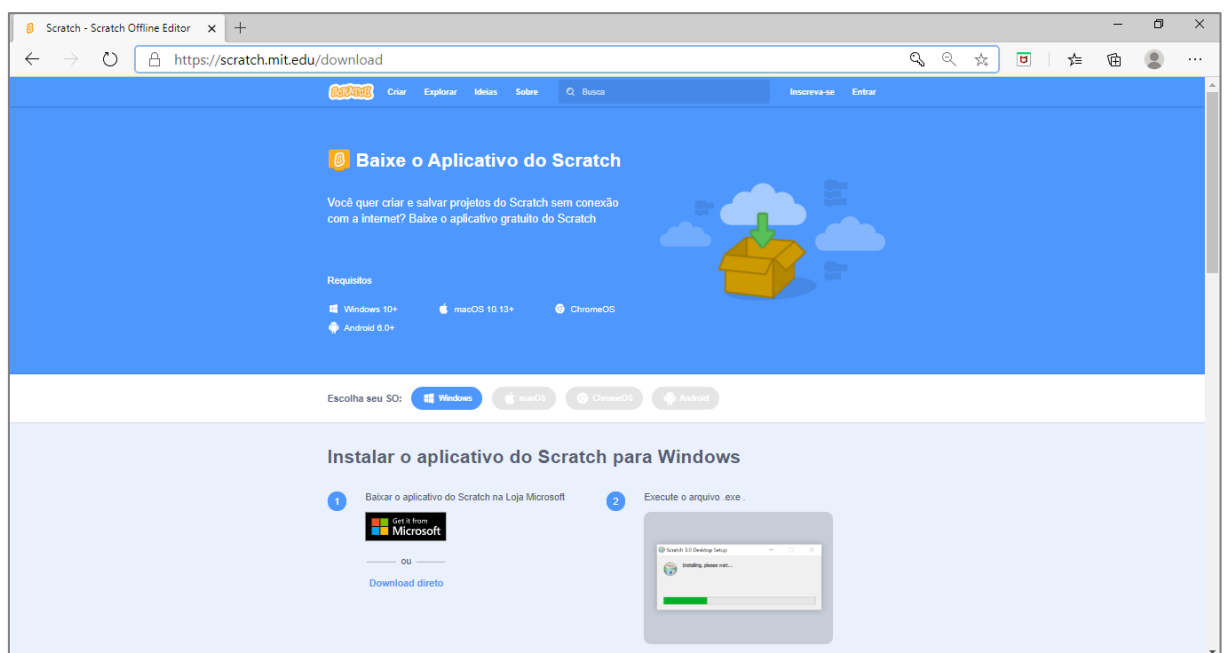
Figura 3.2 – Captura de tela da página inicial da comunidade *on-line* do Scratch para *download*.



Fonte: Das autoras (2022).

Após clicar em “baixar”, será necessário selecionar o Sistema Operacional do seu computador ou *tablet*, como mostramos na Figura 3.3:

Figura 3.3 – Captura de tela da página da comunidade *on-line* do Scratch: como baixar o aplicativo



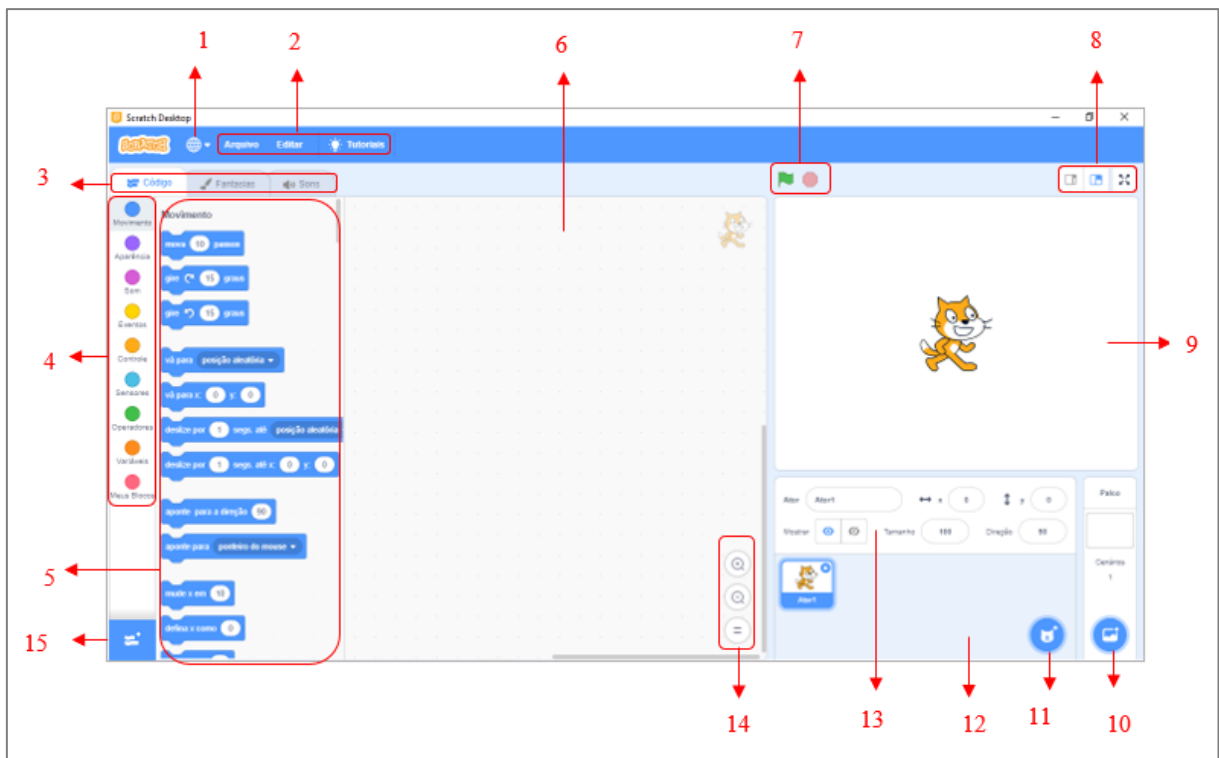
Fonte: Das autoras (2022).

Ao clicar em “Baixar o aplicativo do Scratch na Loja Microsoft”, será aberta a página da Microsoft para realizar o *download*. Após a instalação do *software*, basta começar a criar os projetos!

3.4 Ambiente de programação Scratch

Na Figura 3.4, apresentamos a tela inicial do Scratch 3.0 e, em seguida, descrevemos sucintamente seus principais elementos.

Figura 3.4 – Tela inicial do Scratch 3.0



Fonte: Das autoras (2022).

- 1) Seleção de idioma: alterar para o idioma de preferência do usuário;
- 2) Menu principal: Arquivo (novo, carregar do seu computador, baixar para o seu computador), Editar, Tutoriais. Os tutoriais do Scratch 3.0 foram disponibilizados em vídeo no próprio *software*.
- 3) Abas para as opções: Código, Fantasias e Sons;
- 4) Categorias de comandos: Movimento, Aparência, Som, Eventos, Controle, Sensores, Operadores, Variáveis, Mais blocos;

- 5) Blocos de código: cada categoria de comando possui vários blocos de código;
- 6) Área de edição: depende da aba selecionada (ver item 3).
 - Quando a opção “Código” for selecionada, nesta área é feita a programação, unindo-se os blocos de código;
 - Quando a opção “Fantasias” for selecionada, será possível visualizar as fantasias disponíveis para o ator selecionado. Também é possível editar as fantasias, acrescentar caixas de texto e cores, por exemplo, bem como duplicá-las;
 - Quando a opção “Sons” for selecionada, será possível editar os sons disponíveis para o ator selecionado (cortar trecho, aumentar ou diminuir a velocidade do som do ator).
- 7) Iniciar e Parar: botões para iniciar e parar *Scripts* (Um *script* é o conjunto de blocos de comando utilizados para que uma função seja executada);
- 8) Selecionar modo de exibição: alternar entre modos de edição e modo de apresentação;
- 9) Palco: local de visualização dos atores, cenários e do resultado da sequência de comandos utilizada. Ao iniciar o programa, o objeto que aparece no palco é a imagem do gato;
- 10) Selecionar cenário: é possível escolher um cenário da biblioteca do Scratch, pintar novo cenário, carregar cenário do arquivo do usuário ou selecionar a opção “Surpresa”, que carrega aleatoriamente um cenário da biblioteca;
- 11) Selecione um ator: é possível escolher um ator da biblioteca do Scratch, pintar novo ator, carregar ator do arquivo do usuário ou selecionar a opção “Surpresa”, que carrega aleatoriamente um ator da biblioteca;
- 12) Edição do ator selecionado: nessa área, são exibidos os atores utilizados no projeto e, ao clicar com o botão direito do *mouse* sobre cada ator, é possível duplicar, apagar e exportar. Ao clicar com o botão esquerdo do *mouse* sobre um ator, a programação de cada ator é feita separadamente;
- 13) Nessa área, são exibidas as informações do ator selecionado: nome do ator, coordenadas da posição do ator, tamanho, direção, mostrar e ocultar ator no palco;
- 14) Aumentar e diminuir os blocos de código utilizados na área de edição (item 6);

15) Adicionar uma extensão: Música (sons de instrumentos musicais), Caneta (desenhar com os atores), Detecção de vídeo (detecta movimento com a câmera), Texto para Fala, Traduzir (traduz texto para várias línguas), Makey Makey, micro:bit, Lego Mindstorms EV3, Lego WeDo 2.0.

No tópico a seguir, detalharemos alguns dos itens listados na Figura 3.4, que consideramos fundamentais para entender o funcionamento do Scratch. Mais adiante, iremos sugerir materiais que podem ser utilizados para consultar mais detalhes sobre as funcionalidades do *software*.

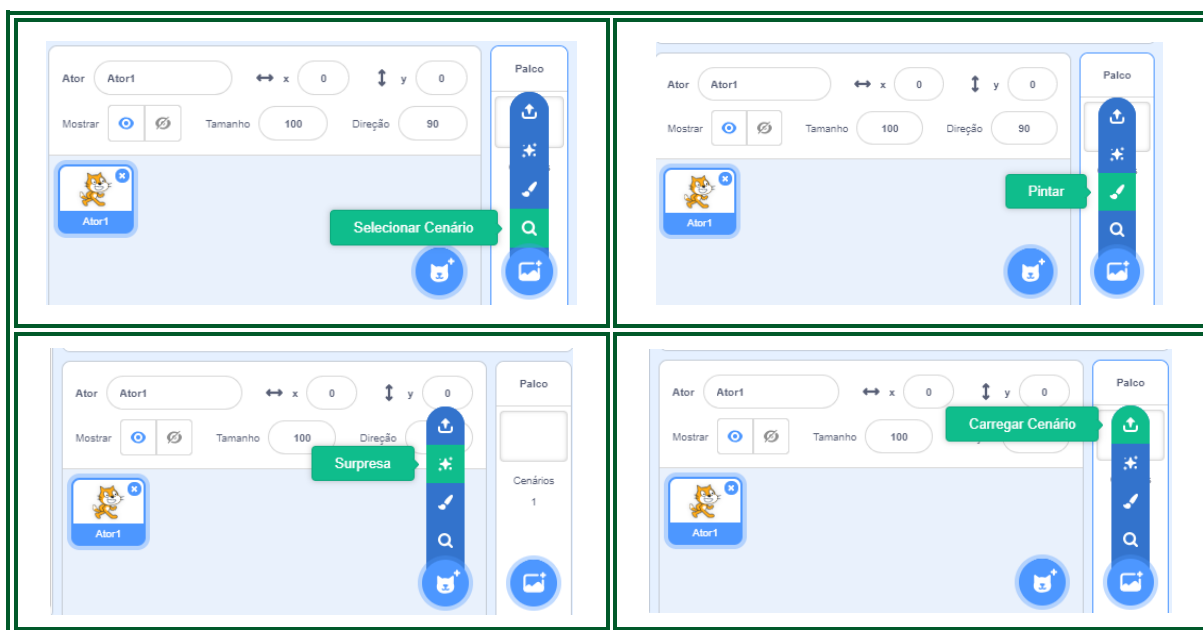
3.5 Palco

Como mencionamos no item 9 da Figura 3.4, o palco é o local de visualização dos atores, cenários e do resultado da sequência de códigos utilizada, ou seja, onde pode ser visualizado o resultado da programação de jogos, histórias interativas ou animações.

Ao iniciar o Scratch, o objeto que aparece no palco é a imagem do gato, o cenário e a área de edição aparecem em branco. Podemos selecionar outros atores e cenários disponibilizados pelo Scratch ou carregar imagens salvas no computador ou tablet em que estivermos utilizando o *software*.

Para alterar o cenário, na tela inicial do Scratch, clique em “Selecionar Cenário” (item 10 da Figura 3.4) e serão exibidos quatro ícones: “Selecionar Cenário”, “Pintar”, “Surpresa” e “Carregar Cenário”, conforme apresentamos na Figura 3.5.

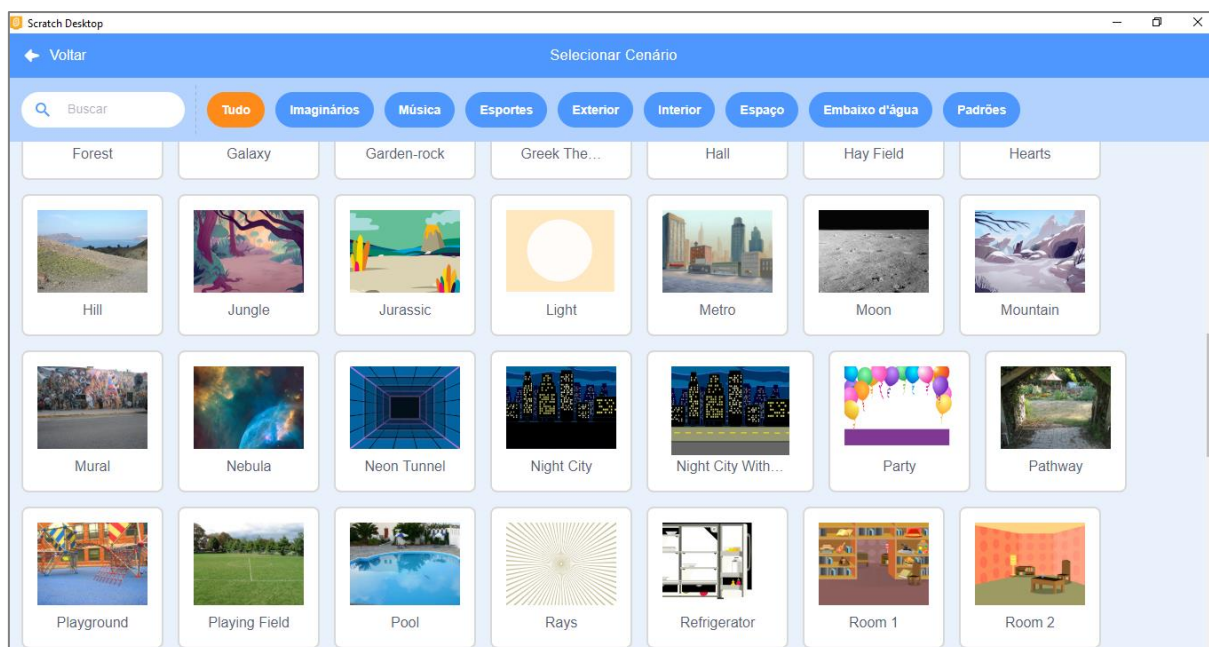
Figura 3.5 – Opções de seleção de cenários do Scratch.



Fonte: Das autoras (2022).

Ao clicar em “Selecionar Cenário”, serão exibidas diversas imagens, como na Figura 3.6.

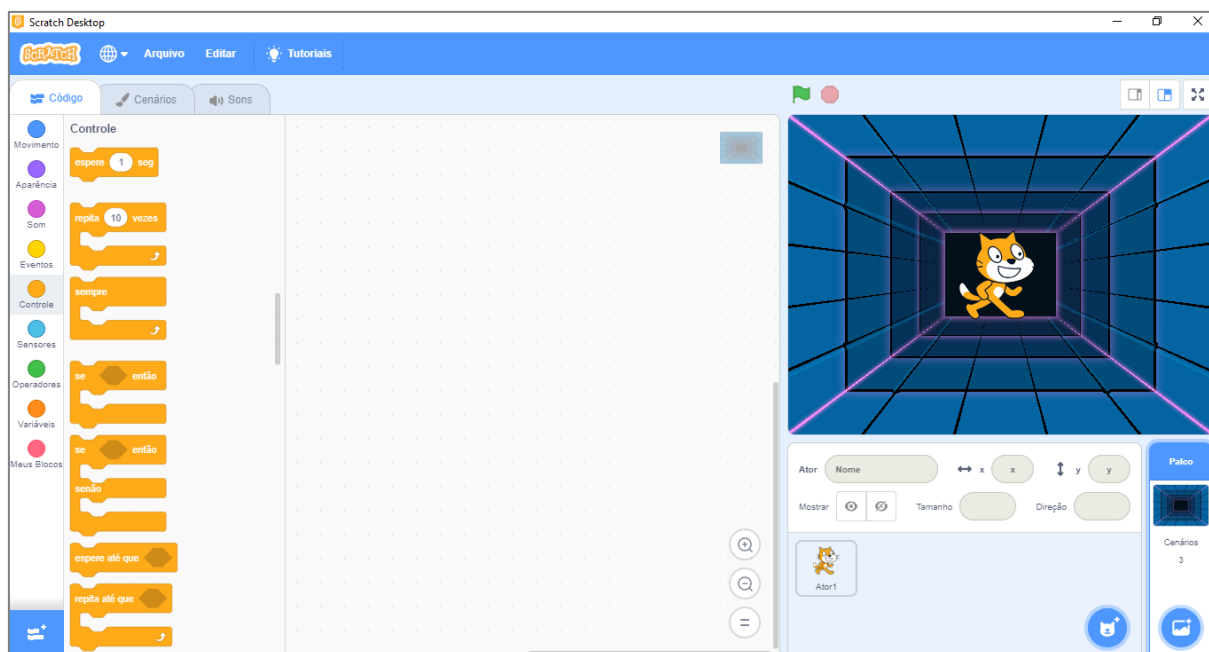
Figura 3.6 – Imagens disponíveis no Scratch para selecionar como Cenário.



Fonte: Das autoras (2022).

Em seguida, você poderá clicar em uma dessas imagens e o palco será modificado, como ilustrado na Figura 3.7.

Figura 3.7 – Tela do Scratch com exemplo de cenário selecionado.

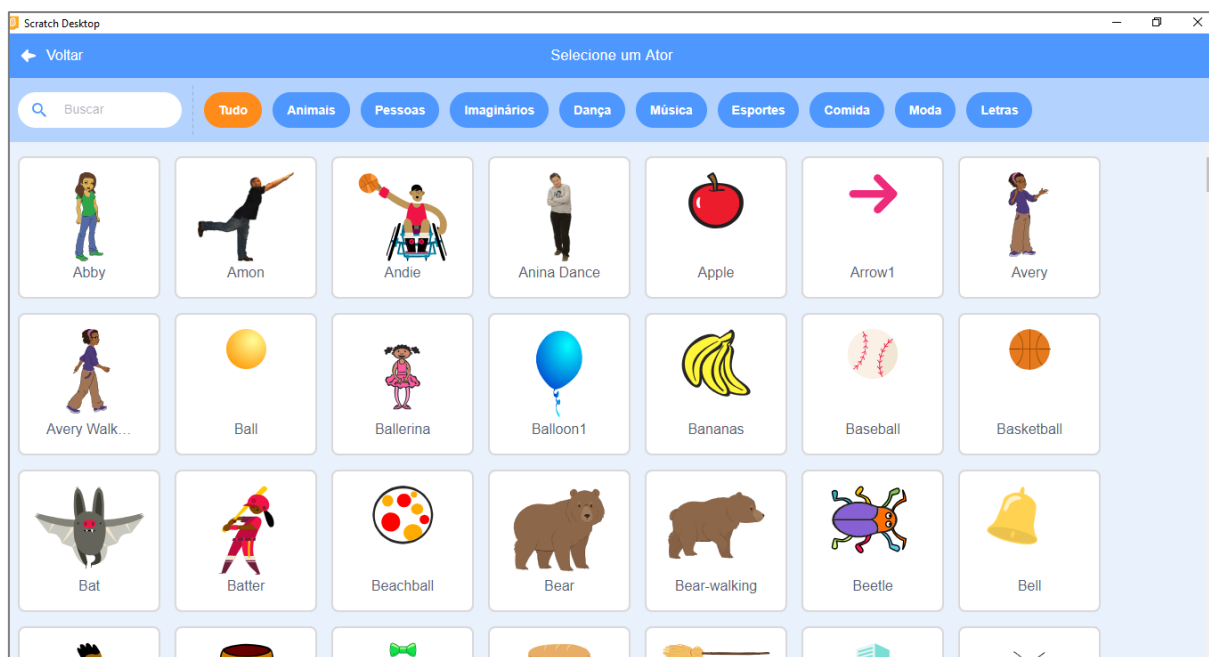


Fonte: Das autoras (2022).

Caso deseje acrescentar imagens salvas no computador ou *tablet*, basta selecionar o ícone “Carregar Cenário” e escolher a imagem desejada.

De modo análogo, para acrescentar outro ator na tela inicial do Scratch, clique em “Selecione um Ator” (item 11 da Figura 3.4) e serão exibidas diversas imagens, como na Figura 3.8.

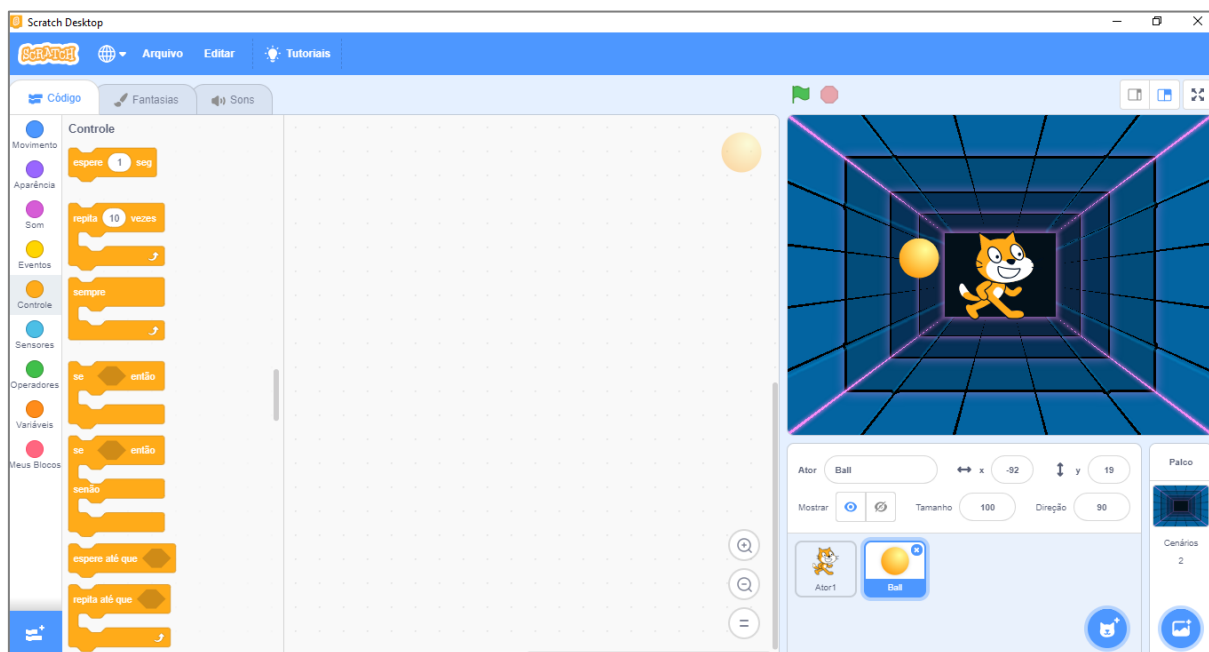
Figura 3.8 - Imagens disponíveis no Scratch para selecionar como Ator.



Fonte: Das autoras (2022).

Em seguida, clique em uma das imagens e será acrescentado o ator no palco, como ilustrado na Figura 3.9, onde foi acrescentada a imagem de uma bola.

Figura 3.9 – Tela do Scratch com exemplo de Ator selecionado.



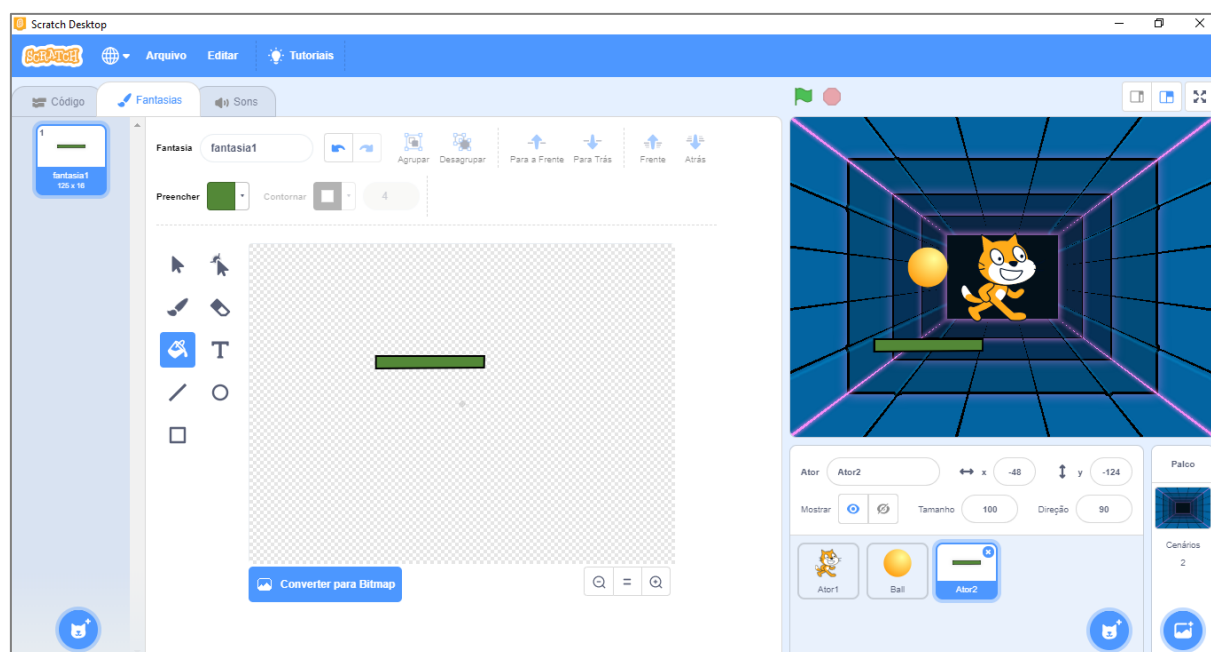
Fonte: Das autoras (2022).

Podem ser acrescentados diversos atores em um mesmo projeto e cada um deles pode ser editado separadamente.

3.6 Atores

É possível apagar um dos atores ou duplicar os atores, clicando com o botão direito do *mouse* sobre eles na área de edição do ator selecionado (item 12 da Figura 3.4). Também é possível selecionar ou pintar outros atores. Na Figura 3.10, apresentamos um ator (barra) que foi desenhado no próprio *software*, por ser uma imagem simples.

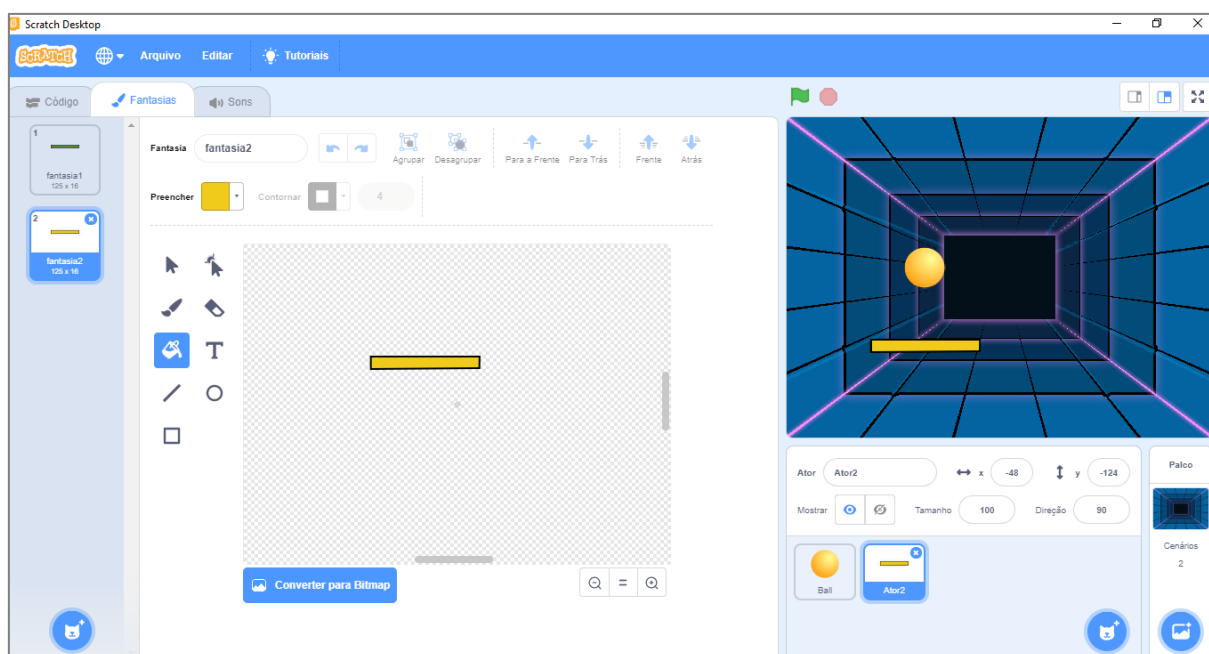
Figura 3.10 – Aba “Fantasias” para edição de atores no Scratch.



Fonte: Das autoras (2022).

Também é possível criar diferentes fantasias para o mesmo ator, que podem ser utilizadas em diferentes momentos da programação de um jogo. Podemos selecionar duas cores diferentes para o mesmo ator, por exemplo, ao duplicar a fantasia e selecionar uma cor para uma delas, conforme mostra a Figura 3.11, em que a fantasia 1 possui cor verde e a fantasia 2 possui cor amarela. Também foi excluída a imagem do gato, ao clicar no x sobre a imagem do gato.

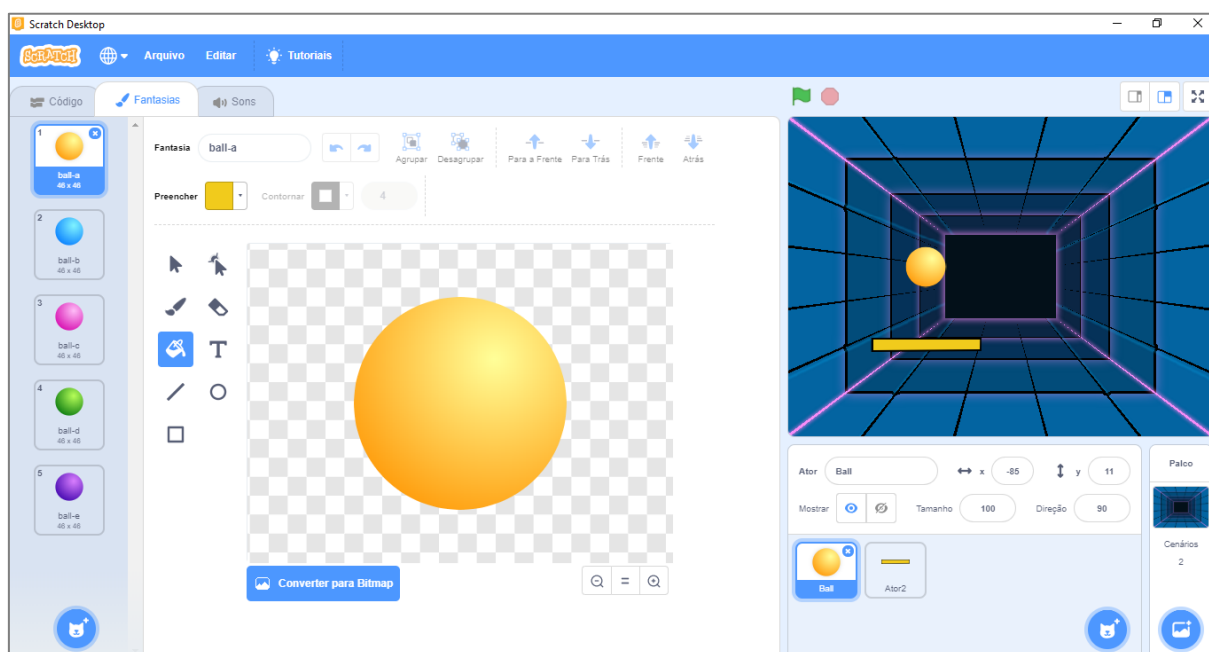
Figura 3.11 – Exemplo de como alterar as cores do ator na aba “Fantasias”.



Fonte: Das autoras (2022).

Os atores disponibilizados no Scratch também possuem diferentes fantasias. A seguir, na Figura 3.12, apresentamos um exemplo de ator com as fantasias que foram disponibilizadas pelo próprio *software*.

Figura 3.12 – Exemplo de ator com cinco fantasias disponibilizadas no Scratch.

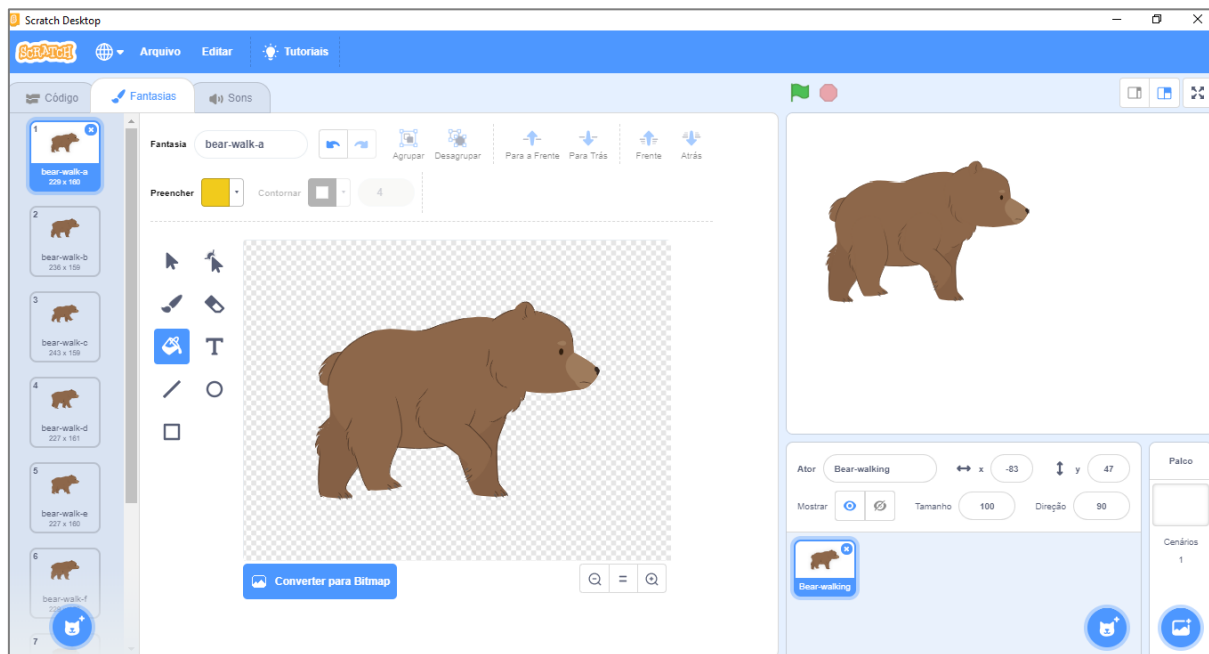


Fonte: Das autoras (2022).

Além de diferentes cores, alguns atores disponibilizados pelo *software* apresentam fantasias úteis para dar a ideia de movimento, como mostra a Figura 3.13,

em que as fantasias simulam o movimento do urso caminhando, dependendo da programação realizada utilizando a mudança de fantasias.

Figura 3.13 – Fantasias do ator denominado “urso” no Scratch.

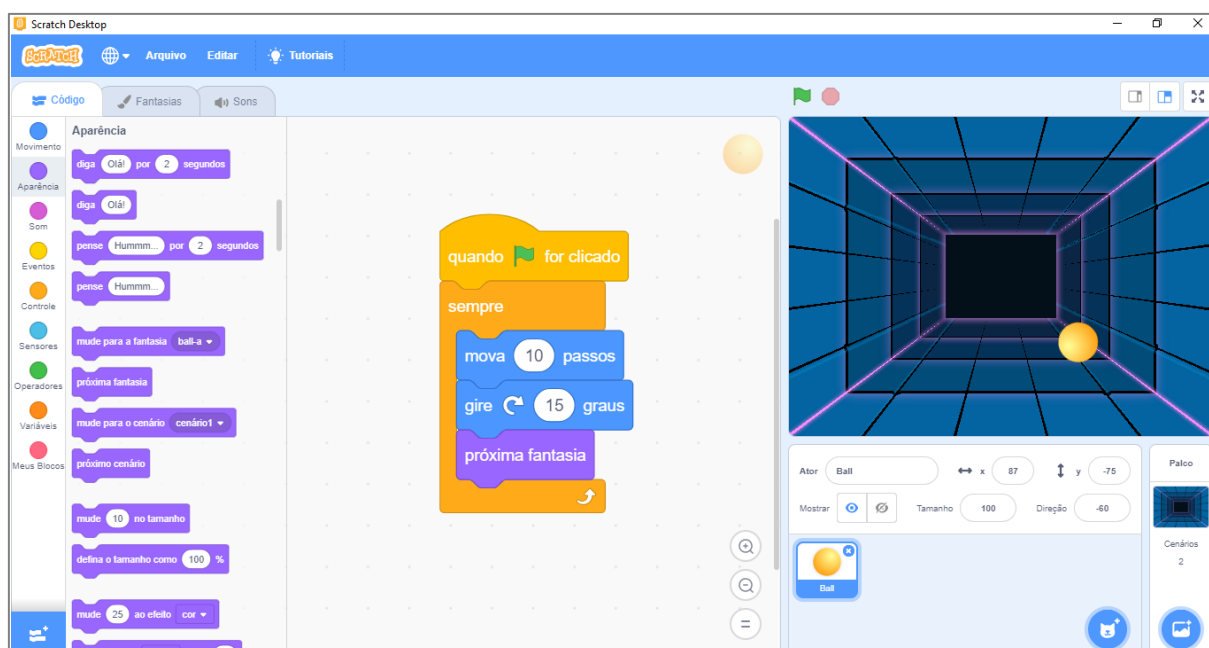


Fonte: Das autoras (2022).

Para programar o comportamento de cada ator escolhido, será necessário escolher blocos de comandos e encaixá-los na área de edição. Podem ser escolhidos comandos de diferentes categorias, como os de movimento, aparência, som e eventos, por exemplo. Podemos clicar em um bloco na área de edição para testar o resultado ou encaixar vários blocos e clicar sobre eles para visualizar o resultado da combinação desses blocos. Também é possível excluir ou reordenar os blocos a qualquer momento.

Na Figura 3.14, apresentamos um exemplo de combinação de blocos de comando para o ator “Ball”, em que estão presentes comandos das categorias Eventos, Controle, Movimento e Aparência. O resultado dessa programação é o movimento circular do ator, devido ao uso do bloco de movimento que faz com que ele gire 15 graus para a direita, bem como a mudança de cor devido ao uso do bloco de aparência “próxima fantasia”. Além disso, o movimento se repete de maneira ininterrupta devido ao uso do bloco de controle “sempre”. O bloco de evento utilizado faz com que a animação se inicie quando a bandeira verde for clicada e, para parar, basta clicar no ícone vermelho localizado ao lado da bandeira verde.

Figura 3.14 – Exemplo de combinação de blocos de comando para o ator “Ball”.



Fonte: Das autoras (2022).

Dica: Em sala de aula, sugerimos que, no primeiro momento de contato com o *software*, sejam apresentados os seus elementos básicos e um exemplo simples de como unir os blocos, como o da figura anterior, por exemplo, evitando mostrar todas as funcionalidades do *software* de maneira extensa e expositiva. Após essa breve explicação, pode ser proposto que as estudantes e os estudantes explorem o Scratch para conhecer os blocos de todas as categorias de comando e experimentar diferentes combinações de comandos. Consideramos que dessa maneira poderão usar a curiosidade e a criatividade, tanto para aprenderem a utilizar o Scratch, quanto para criarem seus primeiros projetos, assumindo uma postura mais ativa e autônoma.

Esperamos que, já nesse primeiro contato com o Scratch, as estudantes e os estudantes encontrem situações-problema e elaborarem estratégias buscando resolvê-las. Desse modo, cada estudante pode compartilhar suas descobertas com a turma, mostrar suas primeiras criações e falar sobre suas primeiras impressões e expectativas.

Dentro do próprio *software* Scratch, há vídeos tutoriais disponíveis com legendas e áudio em português e em outros idiomas.

Para consultar detalhadamente mais funcionalidades do *software* Scratch, sugerimos:

- Utilizar como material de apoio o Produto Educacional intitulado “Trabalhando Matemática com o Scratch”, que contém explicações mais detalhadas das principais funcionalidades do Scratch.



Referência: SILVA, Admilson Iaresk da. Trabalhando Matemática com o Scratch. 2020. 55 p. Produto Educacional (Mestrado em Formação Científica, Educacional e Tecnológica) - Universidade Tecnológica Federal do Paraná, Curitiba, 2020.



Acessar a comunidade *on-line* do Scratch para explorar os projetos compartilhados, tutorias e materiais de apoio. Disponível em: <https://scratch.mit.edu/>. Acesso em: 10 set. 2020.



O USO DO SCRATCH NAS AULAS DE MATEMÁTICA

Neste tópico, apresentamos sugestões de jogos que podem ser construídos no Scratch por estudantes em sala de aula, com o auxílio das professoras e dos professores. Também sugerimos estratégias que podem auxiliar a prática docente no contexto da utilização do Scratch na perspectiva inclusiva.

Essas sugestões são inspiradas em estudos e diálogos entre docentes que participaram do curso em que realizamos a constituição de dados de nossa pesquisa do Mestrado. As pessoas participantes do curso planejaram, experimentaram, vivenciaram, produziram jogos e refletiram sobre a utilização do Scratch para o processo de ensinar e aprender Matemática na perspectiva inclusiva.

Entre os conceitos matemáticos e computacionais que poderão ser mobilizados durante a construção desses jogos, citamos: raciocínio lógico, variáveis, números negativos, localização e deslocamento no referencial cartesiano, manipulação e controle de eventos, entre outros.

As estratégias e os jogos aqui apresentados poderão ser utilizados e adaptados por você, de acordo com seus objetivos e considerando as características, potencialidades, limitações e necessidades das estudantes e dos estudantes que irão utilizar o Scratch em suas aulas.

4.1 Exemplos para familiarização com o Scratch

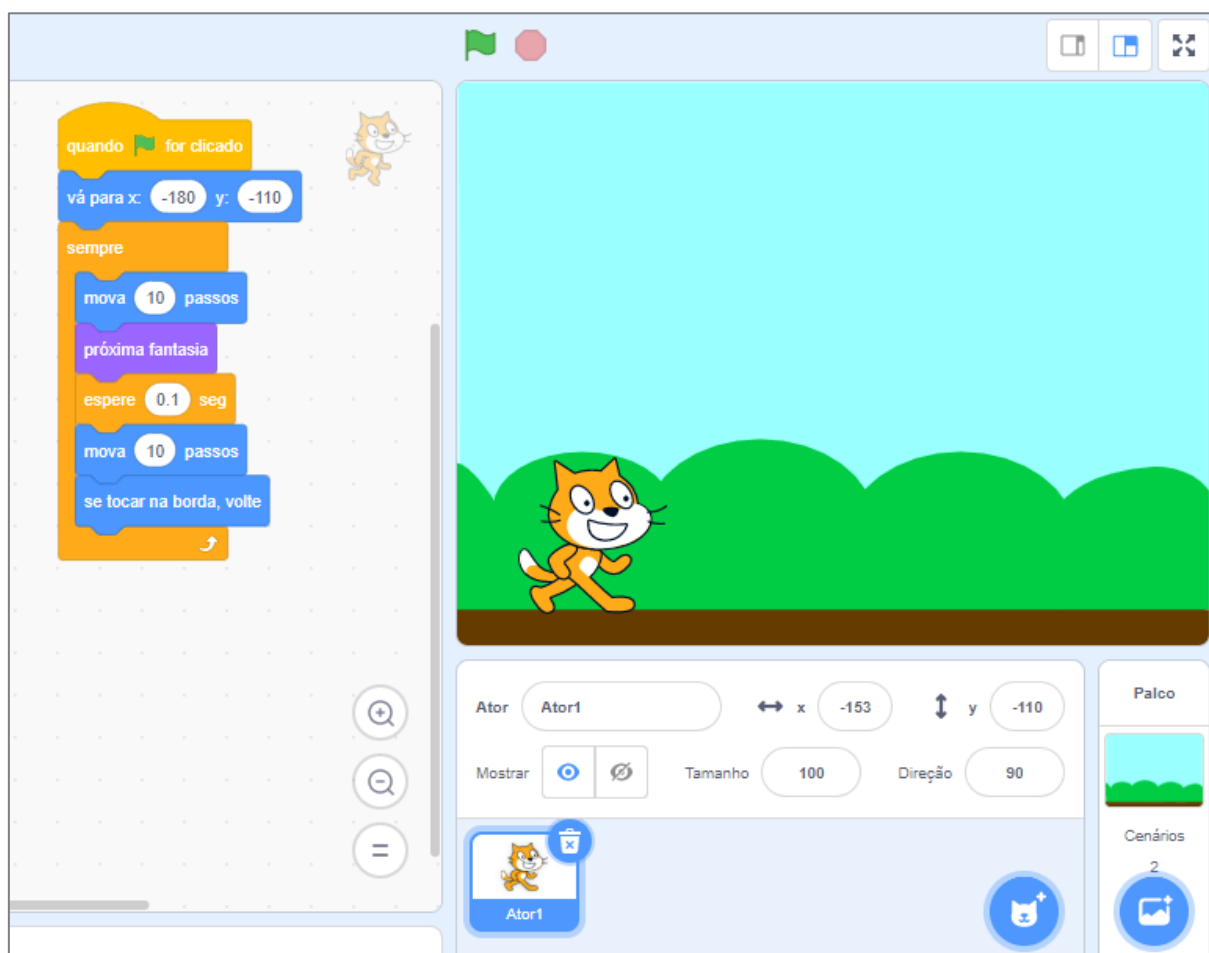
Para iniciar, vamos mostrar como movimentar os atores no Scratch. O movimento dos personagens no Scratch pode ser realizado de diferentes maneiras. Vamos mostrar alguns exemplos.

Em todos os exemplos, é possível que cada estudante selecione o ator e o cenário de sua preferência. Também é possível acrescentar detalhes ao projeto, além do que for solicitado nos enunciados.

- 1) **Movimentando o ator para a esquerda e para a direita:** Fazer com que um ator se movimente para a esquerda e para a direita, sem parar, quando a bandeira verde for clicada.

Sugestões: dividir o problema em partes menores e acompanhar o desenvolvimento da atividade incentivando as estudantes e os estudantes a experimentarem diferentes blocos, utilizando diferentes estratégias.

Figura 4.1 – Projeto no Scratch intitulado “Movimentando o ator para a esquerda e para a direita”.



Fonte: Das autoras (2022).

Para consultar este projeto na comunidade *on-line* do Scratch, acesse:

 [Projeto no Scratch: Movimentando o ator para a esquerda e para a direita](#)

No Quadro 4.1, apresentamos os comandos de programação utilizados no exemplo 1, denominado “Movimentando o ator para a esquerda e para a direita”, e descrevemos os conceitos matemáticos e computacionais presentes nesses algoritmos⁴.

Quadro 4.1 – Descrição dos conceitos matemáticos e computacionais utilizados na programação do exemplo intitulado “Movimentando o ator para a esquerda e para a direita”.

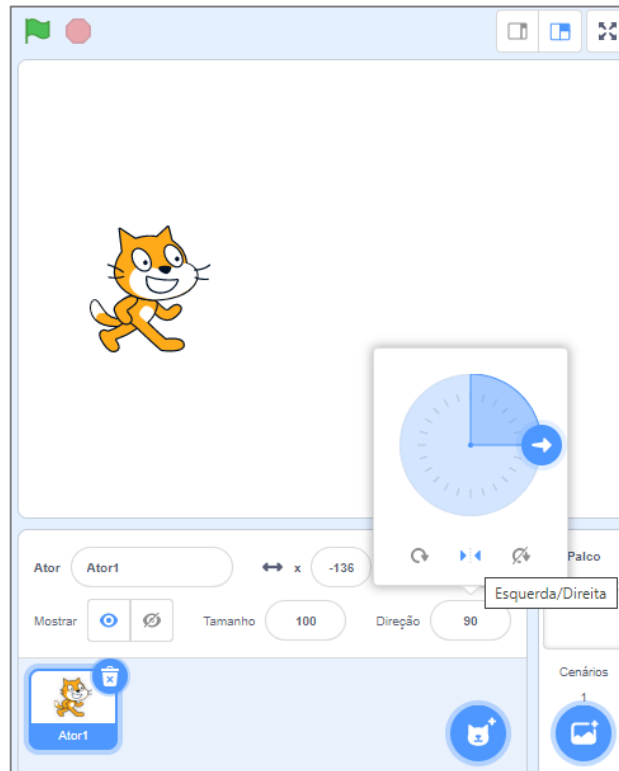
Comandos de programação em Scratch	Conceitos matemáticos e computacionais
	<u>Conceito de listas:</u> É importante considerar que a ordem em que a sequência de passos foi definida pode interferir no resultado da programação. <u>Controle de eventos:</u> O movimento do ator inicia-se ao clicar na bandeira verde. <u>Localização no referencial cartesiano:</u> Podemos escolher o local em que o ator deverá estar no início. Selecionamos o bloco “vá para x: y:”. Em seguida, definimos os valores de x e y correspondentes. Para a escolha desses valores, podemos usar o <i>mouse</i> para clicar sobre o ator e arrastá-lo para o local desejado do palco. Os valores de x e y serão mostrados logo abaixo do palco, na região em que ficam as informações do ator. Em seguida, basta escrever no bloco os valores de x e y. Nesse caso, nós escolhemos (x=-180, y=-110) para que o gato ficasse localizado próximo ao chão do cenário. <u>Repetição:</u> O comando “sempre” é a estrutura de repetição que fará com o ator se movimente sem parar. <u>Sincronização:</u> Para simular o movimento das patas do gato enquanto caminhava, utilizamos o bloco “próxima fantasia” após o bloco “mova 10 passos”. Foi necessário acrescentar o bloco de controle “espere 0.1 segundo” para sincronizar o movimento das patas do gato em uma velocidade adequada.

Fonte: Das autoras (2022).

⁴ Essa maneira de apresentar os comandos do Scratch e os conceitos matemáticos e computacionais foi inspirada em Azevedo (2017, p. 74).

Dica: Para movimentar o ator para a esquerda e para a direita neste exemplo, é necessário selecionar a opção “esquerda-direita”, localizada logo abaixo do palco, onde ficam os dados do ator, conforme mostra a Figura 4.2:

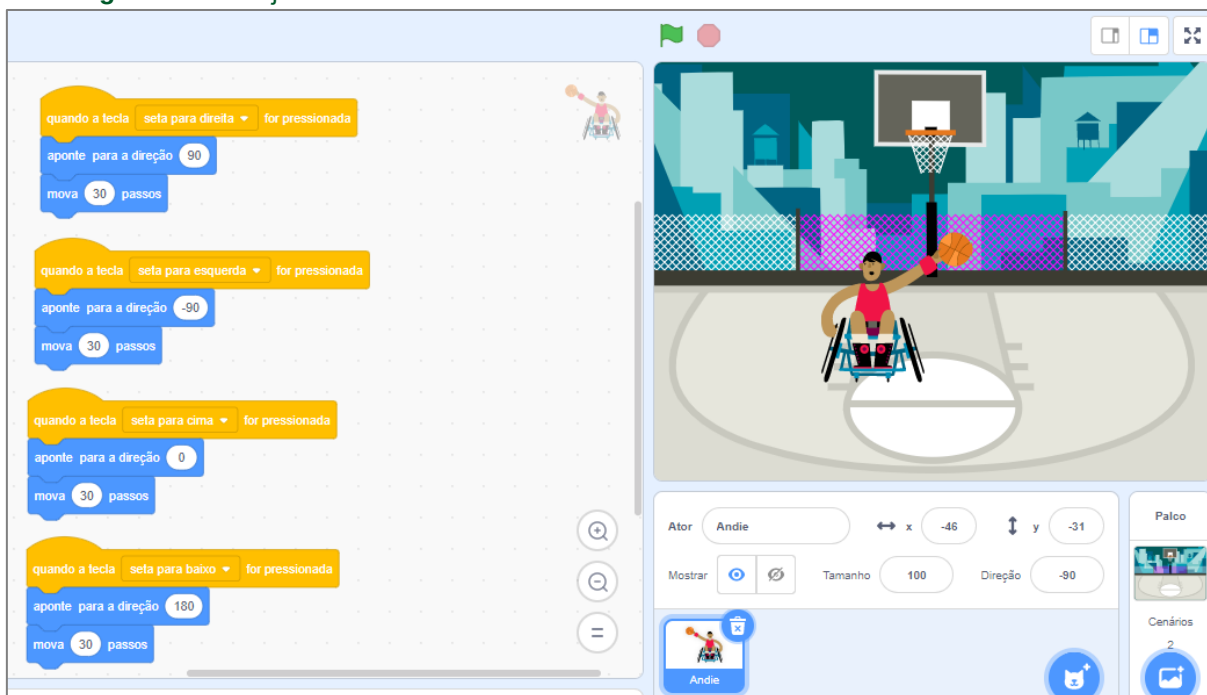
Figura 4.2 – Seleção da opção “esquerda-direita” para o ator no Scratch.



Fonte: Das autoras (2022).

- 2) Movimentando o ator com as setas do teclado:** Fazer um ator se movimentar quando forem pressionadas as teclas “seta para cima”, “seta para baixo”, “seta para a esquerda” e “seta para a direita”.

Figura 4.3 – Projeto no Scratch intitulado “Movimentando o ator com as setas do teclado”.



Fonte: Das autoras (2022).

Para consultar este projeto na comunidade *on-line* do Scratch, acesse:

 [Projeto no Scratch: Movimentando o ator com as setas do teclado](#)

Dica: No Scratch, você pode usar um bloco de evento inicial diferente para cada *script*. Relembrando: *script* é o conjunto de blocos de comando utilizados para que uma função seja executada.

Quadro 4.2 – Descrição dos conceitos matemáticos e computacionais utilizados na programação do exemplo intitulado “Movimentando o ator com as setas do teclado”.

Comandos de programação em Scratch	Conceitos matemáticos e computacionais
	Controle de eventos: Podemos usar um bloco de evento inicial diferente para cada <i>script</i>. Neste exemplo, utilizamos um bloco de evento inicial para programar cada movimento do ator de acordo com as setas do teclado. Para programar o movimento do ator para a direita, usamos o bloco “Quando a tecla ‘seta para direita’ for pressionada” e unimos os blocos “aponte para a direção 90” e “mova 30 passos”. Depois, utilizamos outros conjuntos de blocos para programar o movimento do ator para a esquerda, para cima e para baixo. <u>Noção de ângulo como mudança de direção:</u> Antes de mover os passos, é necessário que o ator aponte para a direção em que irá se movimentar. Ao utilizar o comando “aponte para a direção 0”, o Scratch relaciona ângulos e localização espacial. - Para a direita: aponte para a direção 90°. - Para a esquerda: aponte para a direção -90°. - Para cima: aponte para a direção 0°. - Para baixo: aponte para a direção 180°. <u>Velocidade:</u> A quantidade de passos selecionada no bloco “mova 10 passos” interfere na velocidade do ator no jogo.

Fonte: Das autoras (2022).

Sugestão de intervenção: Abrir o espaço para investigação para que as estudantes e os estudantes explorem a **ideia de velocidade** com essa atividade. Você pode sugerir que as estudantes e os estudantes alterem a quantidade de passos no bloco “mova 10 passos”, substituindo o valor “10” por valores menores e maiores, observando o movimento do ator cada vez que o valor for alterado.

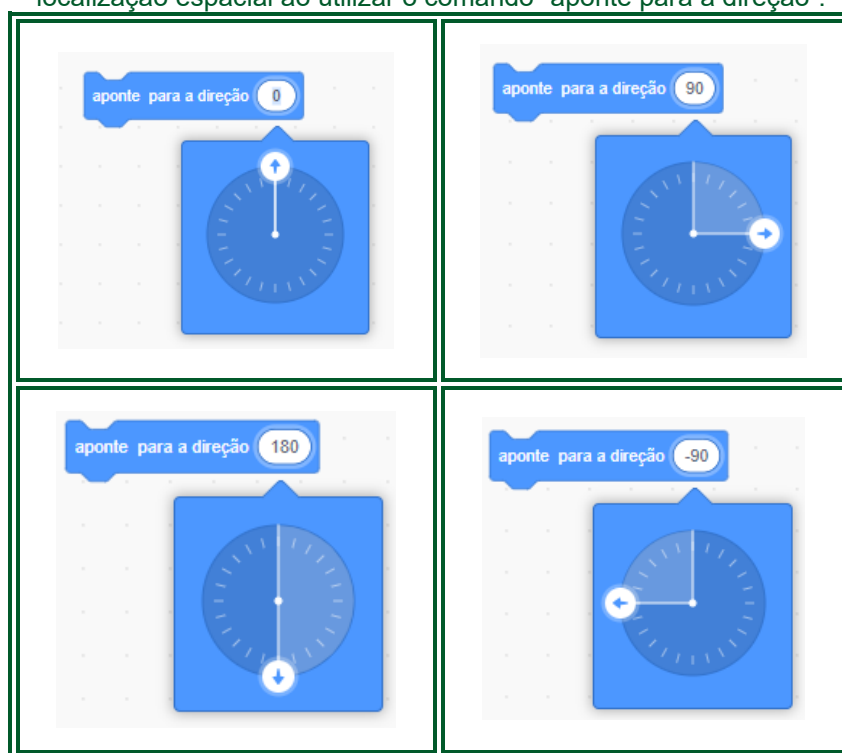
Assim, poderão perceber que:

- Aumentando o número de passos, aumentará a distância percorrida pelo ator no mesmo espaço de tempo, o que indica que ele se movimentará mais rápido;

- Diminuindo o número de passos, diminuirá a distância percorrida pelo ator no mesmo espaço de tempo, o que indica que ele se movimentará mais devagar.

Dica: Durante o desenvolvimento de jogos no Scratch em que seja necessário definir a direção para onde o ator selecionado irá apontar, pode surgir dúvidas a respeito do valor que deverá ser escolhido. Observamos que, no Scratch, há um **recurso visual** que pode auxiliar na compreensão dessa **relação entre ângulos e localização espacial**: ao utilizar o comando “aponte para a direção 0”, clicando sobre a parte branca editável do bloco, aparece o desenho de um círculo com uma seta e, à medida que movimentamos a seta, altera-se o valor dos graus nesse comando, como podemos observar na figura a seguir:

Figura 4.4 – Recurso visual que pode auxiliar na compreensão dessa relação entre ângulos e localização espacial ao utilizar o comando “aponte para a direção”.

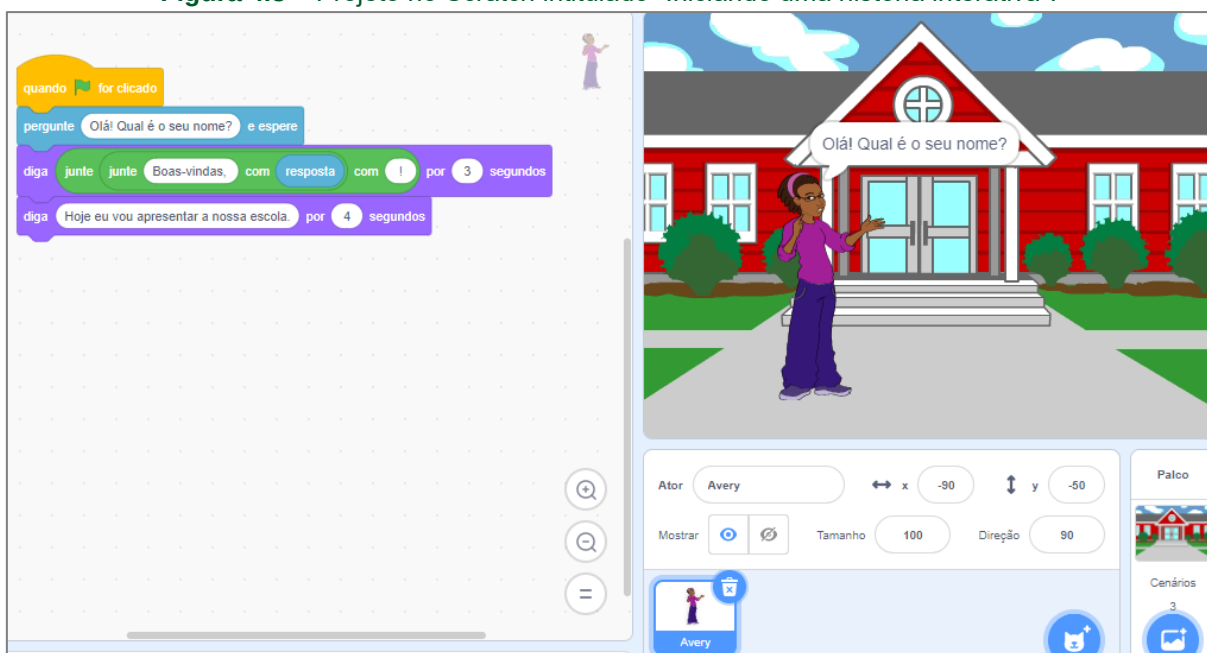


Fonte: Das autoras (2022).

3) Iniciando uma história interativa: Vamos construir uma história interativa? Escolha um ator e faça com que ele pergunte o nome da pessoa que está usando o computador. A pessoa deverá digitar o nome e, em seguida, irá aparecer na tela uma frase do ator usando o nome digitado. Em seguida, use a criatividade para continuar criando a sua história!

Sugestão: Você pode solicitar que as estudantes e os estudantes acrescentem mais personagens, novas frases, novos cenários e usem a criatividade para construir uma história. Poderão fazer diálogos entre personagens e poderão acrescentar comandos para que a pessoa que estiver usando o Scratch possa participar dessa história.

Figura 4.3 – Projeto no Scratch intitulado “Iniciando uma história interativa”.



Fonte: Das autoras (2022).

Para consultar este projeto na comunidade *on-line* do Scratch, acesse:

 [Projeto no Scratch: Iniciando uma história interativa](#)

Veja, nas imagens a seguir, exemplos de como seria a sequência de frases nesse projeto:

Figura 4.4 – Sequência de balões de fala no projeto intitulado “Iniciando uma história interativa”.



Fonte: Das autoras (2022).

No Scratch, podemos utilizar a extensão **“Texto para fala”**, para que as frases sejam ditas por áudio, possibilitando que **peessoas cegas e videntes** tenham acesso.

Veja a figura a seguir, em que apresentamos um exemplo de sequência de comandos utilizando blocos da extensão “Texto para fala”:

Figura 4.5 – Sequência de comandos utilizando blocos da extensão “Texto para fala”.



Fonte: Das autoras (2022).

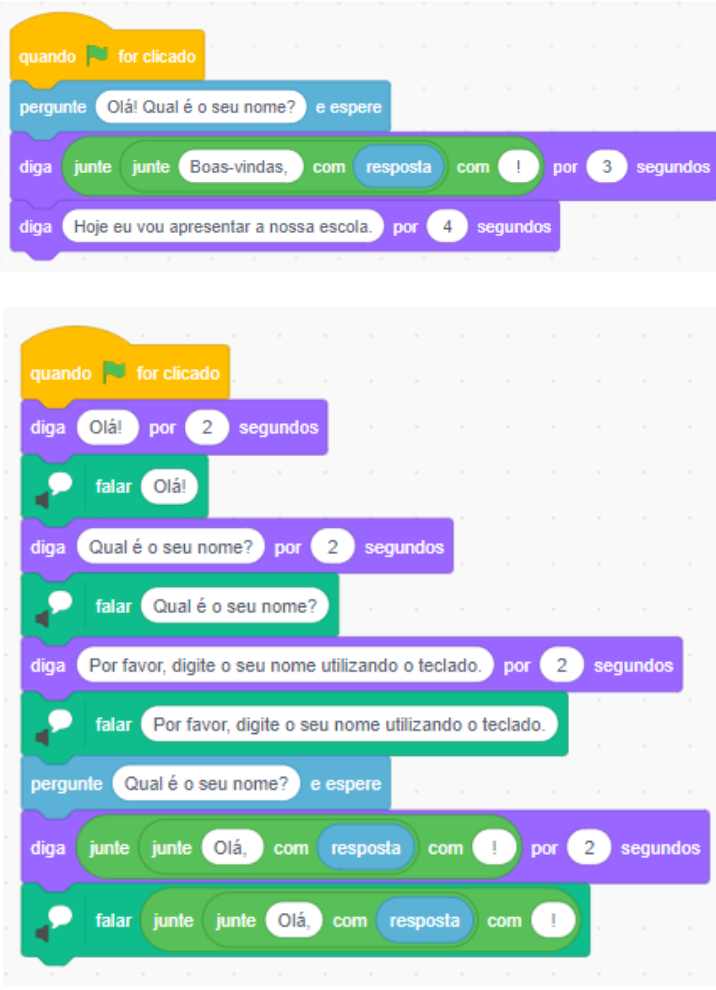
Para consultar este projeto na comunidade *on-line* do Scratch, acesse:

 [Projeto no Scratch: História interativa com texto e áudio](#)

Com base nesses exemplos, poderão ser construídos outros diálogos.

No quadro a seguir, apresentamos os comandos de programação utilizados nos dois projetos de histórias interativas apresentados e descrevemos os conceitos matemáticos e computacionais presentes nesses algoritmos.

Quadro 4.3 – Descrição dos conceitos matemáticos e computacionais utilizados na programação dos dois exemplos de histórias interativas apresentados.

Comandos de programação em Scratch	Conceitos matemáticos e computacionais
	<u>Controle de eventos</u>: A história inicia-se ao clicar na bandeira verde. <u>Conceito de variável</u>: O bloco “Pergunte e espere” é um bloco de <u>entrada de dados</u>. Esse comando possui a função de solicitar dados ao ser executado. O dado solicitado (neste exemplo, o nome da pessoa) será informado pela digitação do nome utilizando o teclado. O nome será armazenado em uma <u>variável</u>, para posterior processamento. O bloco “resposta” é um bloco de <u>saída de dados</u>. Esse comando possui a função de mostrar ao usuário o resultado desse processamento. Neste exemplo, o bloco “resposta” será utilizado para que o nome digitado apareça na frase do ator. <u>Sincronização</u>: O bloco “diga por segundos” é fundamental para sincronizar a fala dos atores. Poderão ser alterados os valores dos segundos de acordo com o tamanho das frases, por exemplo. <u>Operadores lógicos</u>: Os blocos “junte com” são operadores lógicos. Os blocos dessa categoria funcionam como operadores matemáticos lógicos e relacionais. Combinados com outros blocos de comando, podem ser feitos cálculos nos projetos desenvolvidos. Neste exemplo, o bloco “junte com” será utilizado para montar as frases dos diálogos, pois pode ser combinado com o bloco “resposta” para formar as frases.

Fonte: Das autoras (2022).

Sugestão: Além de criar histórias com diálogos entre personagens, podemos criar **jogos de perguntas e respostas**, por exemplo. Para isso, será necessário criar

outras variáveis para armazenar no Scratch as respostas corretas e incorretas. Também será necessário utilizar os blocos de estruturas condicionais como “se então” e “se então... senão”, para definir o que acontecerá cada vez em que for digitada uma resposta correta ou uma resposta incorreta. Jogos como esses podem ser criados pelas estudantes e pelos estudantes com a mediação das professoras e dos professores. Outra sugestão seria a construção desses jogos pelas professoras e pelos professores para retomar conceitos já trabalhados em sala de aula, por exemplo.

Para saber como construir um jogo de perguntas e respostas, acesse o seguinte exemplo de jogo compartilhado na comunidade *on-line* do Scratch:

 [Jogo de perguntas e respostas de matemática](#)

Dica: No Scratch, podemos utilizar a extensão “**Texto para fala**”, para que as perguntas do jogo sejam feitas por áudio, possibilitando que **pessoas cegas e videntes** participem do jogo. Veja, na imagem a seguir, os comandos que podem ser utilizados em um jogo de perguntas e respostas que utiliza texto e som:

Figura 4.6 – Sequência de comandos que podem ser utilizados em um jogo de perguntas e respostas com texto e som.



Fonte: Das autoras (2022).

Para consultar este projeto na comunidade *on-line* do Scratch, acesse:

 [Jogo de perguntas e respostas que utiliza texto e som](#)

4.2 Exemplos de construção de jogos no Scratch

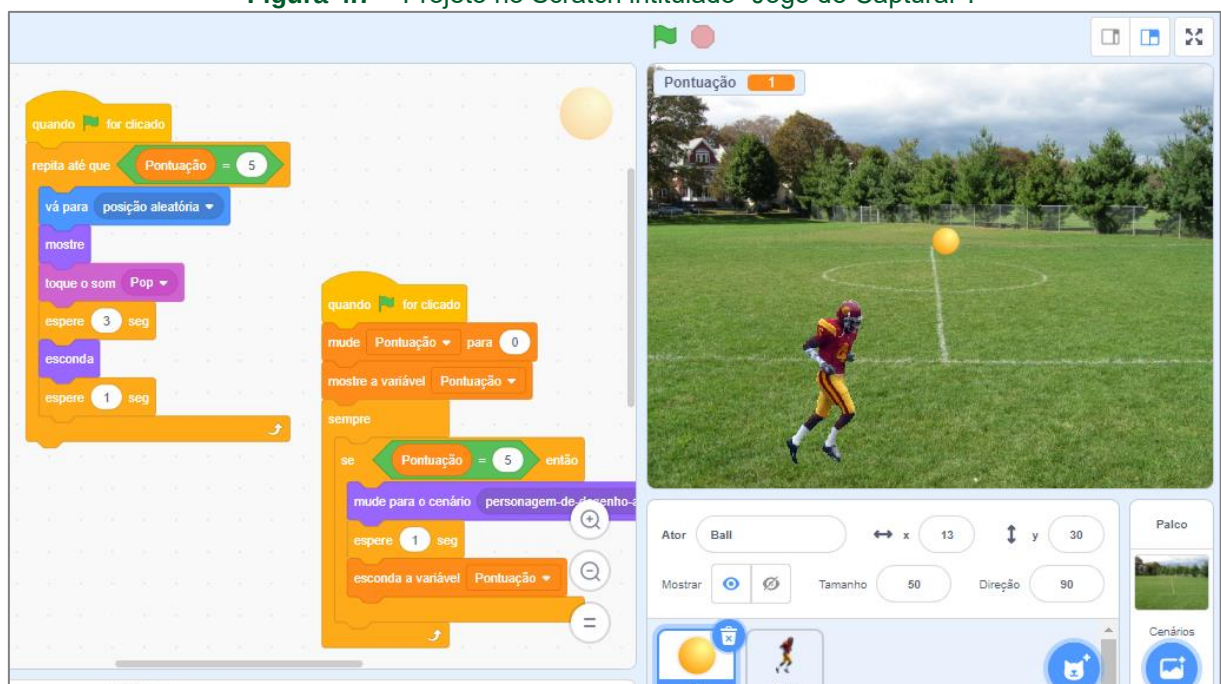
Após apresentar os exemplos para familiarização com o Scratch, compartilhamos alguns jogos que podem ser adaptados, modificados, incrementados por você e por estudantes em suas aulas.

Incentive as estudantes e os estudantes a utilizarem a criatividade para construírem outros jogos!

4.2.1 Jogo de Capturar

Vamos construir um jogo em que uma bola aparece e desaparece em lugares aleatórios do palco e um ator tenta alcançá-la sempre que ela aparecer. Cada vez que o ator alcançar a bola, ele ganhará 1 ponto. Para movimentar o jogador, pressione as teclas ‘seta para cima’, ‘seta para baixo’, ‘seta para a esquerda’ e ‘seta para a direita’.

Figura 4.7 – Projeto no Scratch intitulado “Jogo de Capturar”.



Fonte: Das autoras (2022).

Para consultar este projeto na comunidade *on-line* do Scratch, acesse:

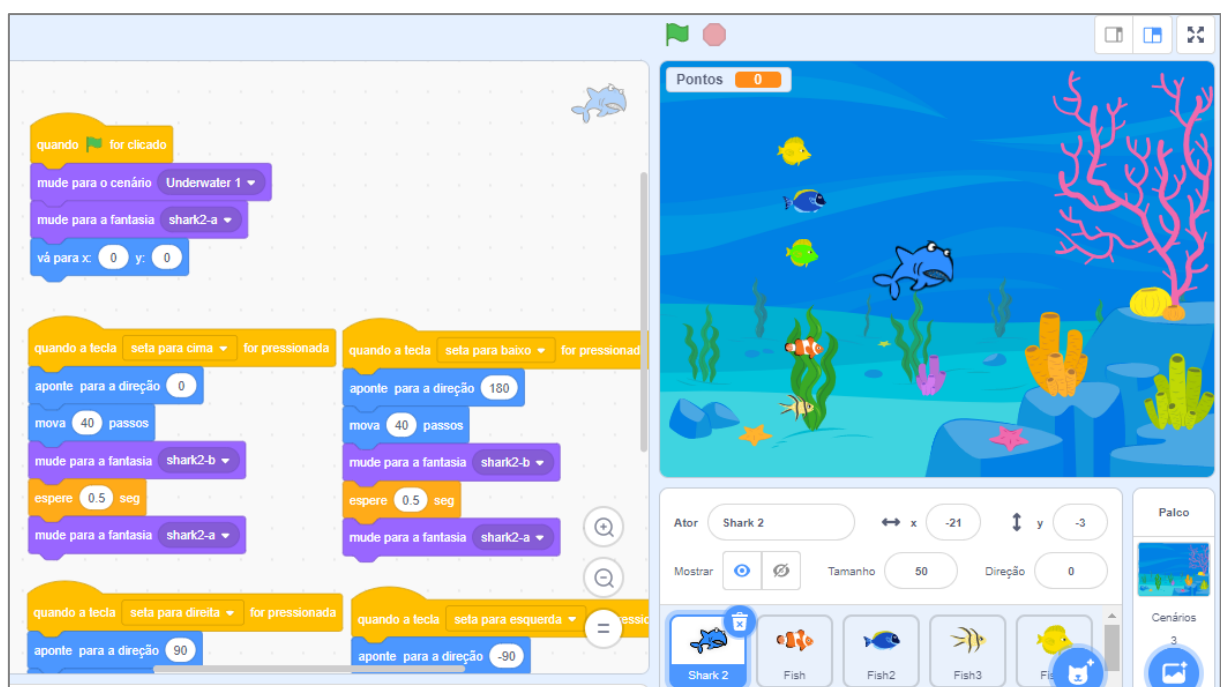
 [Projeto no Scratch: Jogo de Capturar](#)

4.2.2 Jogo do Tubarão⁵

Vamos criar um jogo em que o cenário seja o fundo do mar com tubarões e peixes. O objetivo do jogo é que o tubarão capture os peixes.

Sugestão: Ao programar o movimento dos atores, poderão ser utilizados blocos de movimento para mobilizar o conceito de localização e deslocamento no referencial cartesiano ou para mobilizar o conceito de ângulos.

Figura 4.8 – Projeto no Scratch intitulado “Jogo do Tubarão”.



Fonte: Das autoras (2022).

Para consultar este projeto na comunidade *on-line* do Scratch, acesse:

 [Projeto no Scratch: Jogo do Tubarão](#)

⁵ Este jogo foi construído a partir das ideias e sugestões das pessoas participantes do curso em que ocorreu a constituição dos dados da nossa pesquisa de Mestrado.

Quadro 4.4 – Descrição dos conceitos matemáticos e computacionais utilizados na programação do Jogo do Tubarão para movimentar o tubarão de acordo com as setas do teclado.

Comandos de programação em Scratch	Conceitos matemáticos e computacionais
	<u>Controle de eventos:</u> Neste exemplo, utilizamos quatro <i>scripts</i> para movimentar o ator de acordo com as setas do teclado. Para iniciar cada um desses <i>scripts</i>, utilizamos o bloco “Quando a tecla for pressionada”. Para programar o movimento do ator para a direita, usamos o bloco “Quando a tecla ‘seta para direita’ for pressionada” e unimos os blocos “aponte para a direção 90” e “mova 40 passos”. Depois, utilizamos outros três <i>scripts</i> para programar o movimento do ator para a esquerda, para cima e para baixo.
	<u>Noção de ângulo como mudança de direção:</u> Antes de mover os passos, é necessário que o ator aponte para a direção em que irá se movimentar. Ao utilizar o comando “aponte para a direção 0”, o Scratch relaciona ângulos e localização espacial. - Para a direita: aponte para a direção 90°. - Para a esquerda: aponte para a direção -90°. - Para cima: aponte para a direção 0°. - Para baixo: aponte para a direção 180°.
	<u>Velocidade:</u> A quantidade de passos selecionada no bloco “mova 10 passos” interfere na velocidade do ator no jogo. <u>Sincronização:</u> Para simular o movimento de abrir e fechar a boca do tubarão, utilizamos o bloco “próxima fantasia” após o bloco “mova 40 passos”. Foi necessário acrescentar o bloco de controle “espere 0.5 segundo” para sincronizar o movimento de abrir e fechar a boca do tubarão.



Fonte: Das autoras (2022).

Sugestão: Na versão do jogo apresentada na figura anterior, foi utilizado o comando “vá para x: 0 e y: 0”, para que, ao iniciar o jogo clicando na bandeira verde, o tubarão esteja localizado no centro do palco. Podem ser escolhidos outros valores para x e y utilizando esse mesmo bloco de comando. Também é possível utilizar o comando “vá para posição aleatória”, de maneira que, cada vez que o jogo for iniciado, o tubarão apareça em posições diferentes. Em todas essas situações, é possível discutir sobre a **localização no referencial cartesiano** durante o desenvolvimento do jogo. Assim, a mobilização desses conceitos pode ocorrer a partir da necessidade de definir a localização inicial do tubarão e dos peixes.

No jogo apresentado na figura anterior, o movimento do tubarão ocorre devido ao comando “mova 40 passos” e, embora não haja diretamente a representação das coordenadas cartesianas nesse comando, ao utilizá-lo, podemos observar que serão adicionadas 40 unidades no valor do x ou do y da posição do tubarão, de acordo com os demais comandos selecionados. Por exemplo, na figura a seguir, apresentamos o trecho da programação em que selecionamos o comando de evento “quando a tecla seta para cima for pressionada” e os comandos de movimento “aponte para a direção 0” e “mova 40 passos”.

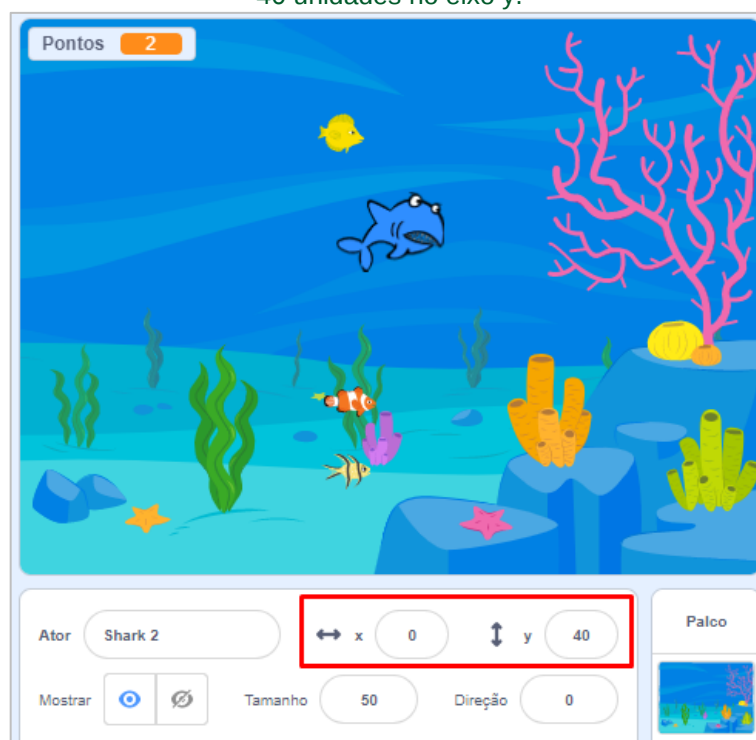
Figura 4.9 – Sequência de comandos para movimentar o ator ao clicar com a seta do teclado.



Fonte: Das autoras (2022).

Nesse caso, considerando que a localização inicial do tubarão era (0,0) e foi selecionado o comando “aponte para a direção 0”, o tubarão se movimentou 40 unidades para cima, ou seja, foram adicionadas 40 unidades no eixo y, passando a localizar-se no ponto (0, 40), como podemos observar na figura a seguir:

Figura 4.10 – Localização do ator denominado “tubarão” no plano cartesiano após deslocamento de 40 unidades no eixo y.



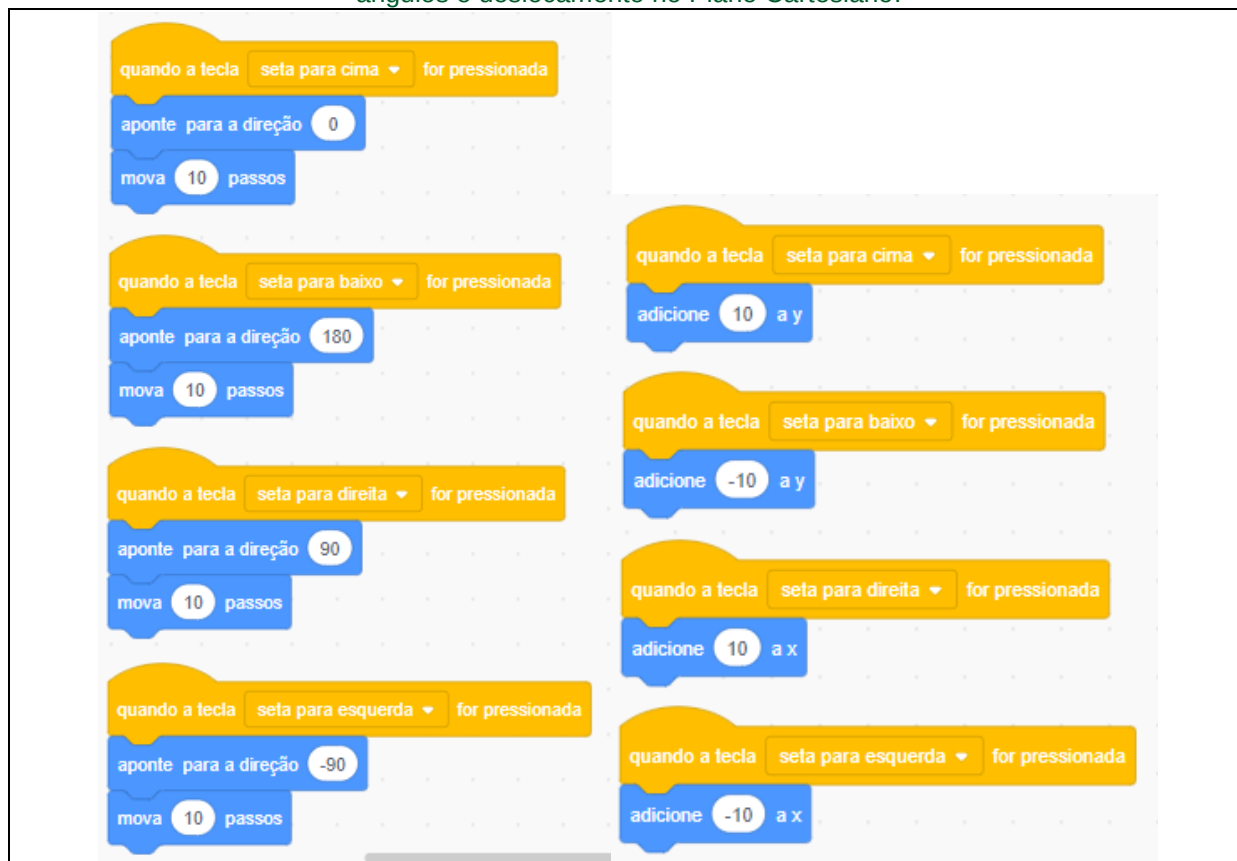
Fonte: Das autoras (2022).

Ressaltamos que o Scratch possui detalhes que podem auxiliar na compreensão dos conceitos que estão sendo trabalhados, favorecendo o aprendizado de estudantes que possuam ou não dificuldades de aprendizagem e que estejam na fase inicial da assimilação do conteúdo de referencial cartesiano. Entre esses detalhes, destacamos o seguinte **recurso visual** na área em que são exibidas as coordenadas da posição do ator selecionado: ao lado do x, há uma seta na horizontal e, ao lado do y, há uma seta na vertical, para representar a **direção dos eixos cartesianos**.

Ao utilizar o comando “aponte para a direção 0”, observamos, ainda, que o Scratch relaciona ângulos e localização espacial, considerando os pontos cardeais (Norte, Sul, Leste e Oeste) e colaterais (Nordeste, Noroeste, Sudoeste e Sudeste). Nesse caso, o ponto cardinal representado pela localização norte representa o ângulo de 0° e, ao mudarmos a direção em qualquer sentido, o movimento ocorre envolvendo a unidade de medida de ângulos denominada grau.

Além dessa maneira de realizar a programação utilizando a relação entre ângulos e localização espacial para movimentar o ator com as setas do teclado, também é possível utilizar os comandos “adicione 10 a x” e “adicione 10 a y”. Nesses blocos de comando, o valor poderá ser editado de acordo com o objetivo que se pretende alcançar. Nesse caso, poderá ser mobilizado o conceito de deslocamento no referencial cartesiano novamente, incluindo o conceito de números positivos e negativos. Ao utilizar o comando “adicione 10 a x”, o movimento ocorre na direção horizontal, já que se trata do movimento no eixo x. Nesse comando, o valor positivo do x indica que o movimento ocorre para a direita e o valor negativo do x indica que o movimento ocorre para a esquerda. Ao utilizar o comando “adicione 10 a y”, o movimento ocorre na direção vertical, já que se trata do movimento no eixo y. Nesse comando, o valor positivo do y indica que o movimento ocorre para cima e o valor negativo do y indica que o movimento ocorre para baixo. Na figura a seguir, apresentamos essas duas maneiras de programar o movimento do ator de acordo com as setas do teclado:

Figura 4.11 – Programação do movimento do ator de acordo com as setas do teclado utilizando ângulos e deslocamento no Plano Cartesiano.



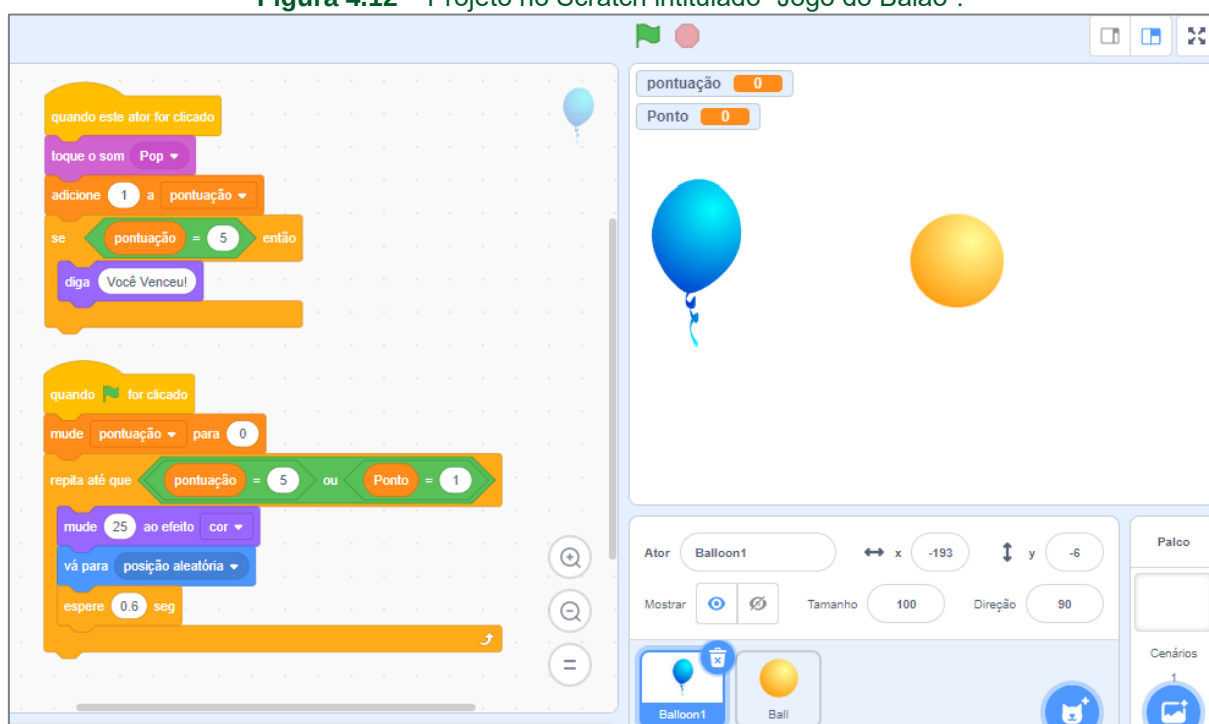
Fonte: Das autoras (2022).

4.2.3 Jogo do Balão⁶

Utilize o *mouse* para clicar nos balões. Cada vez que clicar nos balões, você ganhará um ponto. Ao atingir cinco pontos, você ganhará o jogo. Se clicar na bola, perderá o jogo.

⁶ Jogo desenvolvido pelas pessoas participantes durante o curso. Inspirado em um dos tutoriais disponíveis no Scratch 3.0 e no "Jogo de Capturar" que apresentamos no primeiro exemplo da seção 4.2.

Figura 4.12 – Projeto no Scratch intitulado “Jogo do Balão”.

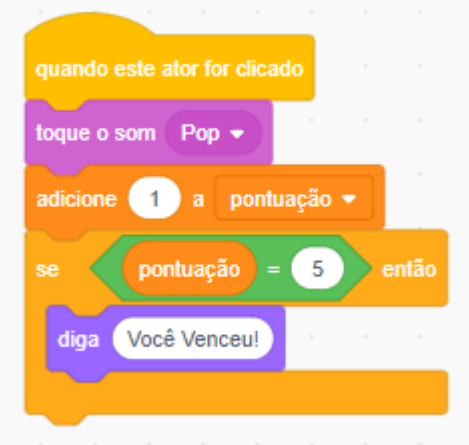


Fonte: Das autoras (2022).

Para consultar este projeto na comunidade *on-line* do Scratch, acesse

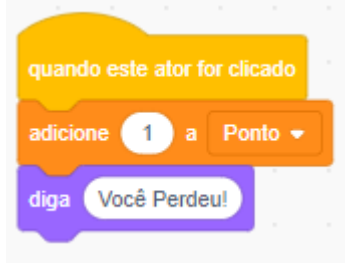
 [Projeto no Scratch: Jogo do Balão](#)

Quadro 4.5 – Descrição dos conceitos matemáticos e computacionais utilizados para calcular a pontuação ao clicar no botão no “Jogo do Balão”.

Comandos de programação em Scratch	Conceitos matemáticos e lógica de programação
	<u>Controle de eventos</u>: Na programação do balão, foi utilizado o bloco de evento: “Quando este ator for clicado”, para iniciar o <i>script</i> que terá a função de calcular a pontuação do balão. <u>Variável</u>: Foi criada a variável denominada “pontuação”, para armazenar os valores quando o balão for clicado no jogo. Foi usado o bloco “adicione 1 a pontuação”, para que ao clicar no balão seja adicionado 1 ponto à variável “pontuação”. <u>Argumentos condicionais</u>: A condição para vencer o jogo é acumular cinco pontos. O bloco “se então” foi utilizado em conjunto com o <u>operador lógico</u> que compara se dois números são iguais. Nesse exemplo, se a variável “pontuação” for igual a 5, então aparecerá a frase “você venceu”. Neste mesmo exemplo, pode ser mobilizado o conceito de <u>comparação entre números</u>.

Fonte: Das autoras (2022).

Quadro 4.6 – Descrição dos conceitos matemáticos e computacionais utilizados para calcular a pontuação ao clicar na bola no “Jogo do Balão”.

Comandos de programação em Scratch	Conceitos matemáticos e lógica de programação
	<u>Controle de eventos:</u> Na programação da bola, foi utilizado o bloco de evento: “Quando este ator for clicado”, para iniciar o <i>script</i> que terá a função de calcular a pontuação da bola. <u>Variável:</u> Foi criada a variável denominada “ponto” para armazenar o valor dos pontos obtidos ao clicar na bola. Quando clicar na bola, aparecerá a mensagem: “Você perdeu”.

Fonte: Das autoras (2022).

Quadro 4.7 – Descrição dos conceitos matemáticos e computacionais utilizados para iniciar o movimento da bola no “Jogo do Balão”.

Comandos de programação em Scratch	Conceitos matemáticos e lógica de programação
	<u>Controle de eventos:</u> O movimento da bola inicia-se ao clicar na bandeira verde. <u>Variável:</u> O jogo inicia com a variável “ponto” igual a 0. <u>Estrutura de repetição:</u> Foi utilizado o bloco “repita até que”, para que o valor da variável “ponto” continue sendo igual a 0 até que o jogo termine, ou seja, até a variável “pontuação” atingir o valor igual a 5 ou a variável “ponto” atingir o valor igual a 1. <u>Operadores lógicos:</u> - Operadores lógicos de comparação entre números (pontuação = 5; ponto = 1) - Operador lógico de disjunção [(pontuação = 5) ou (ponto = 1)]: Resulta em um valor verdadeiro se ao menos um dos dois valores de entrada da operação for verdadeiro, caso contrário, o resultado é falso.

Fonte: Das autoras (2022).

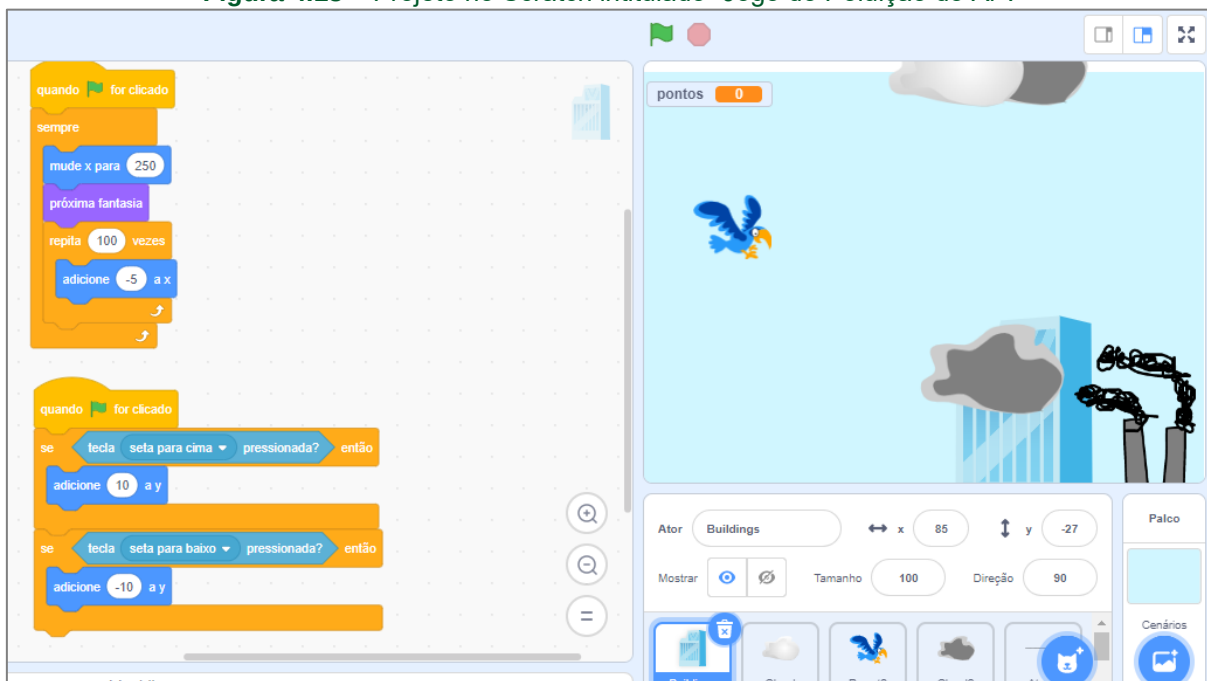
4.2.4 Jogo Poluição do Ar

Vamos construir um jogo em que um pássaro perde pontos ao se encostar nas nuvens poluídas e ganha pontos ao se encostar nas nuvens não poluídas. O pássaro deve se movimentar ao apertarmos as teclas “seta para cima” e “seta para baixo” do teclado do computador.

Referência: Esse jogo foi criado por estudantes sob a orientação do autor do texto *“Produção de Games nas Aulas de Matemática: Por que não?”* (Azevedo, 2017). Conforme mencionado pelo autor, a escolha do tema do jogo foi feita por um grupo de estudantes que relataram que, ao brincar de soltar pipas no bairro, viam o céu poluído, com fumaças escuras provenientes da emissão de gases dos carros e das fábricas locais.

Sugestão: Ao propor a construção de jogos no Scratch em sala de aula, os temas dos jogos podem ser escolhidos pelas estudantes e pelos estudantes, a partir de situações que consideram relevantes em seu cotidiano, por exemplo.

Figura 4.13 – Projeto no Scratch intitulado “Jogo do Poluição do Ar”.



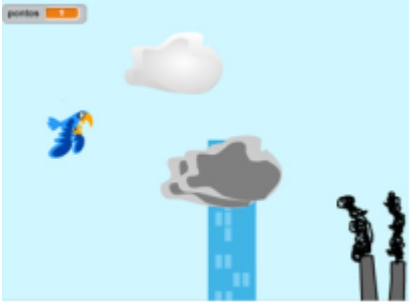
Fonte: Das autoras (2022).

Link para acessar o projeto original do texto:

 [Projeto no Scratch: Jogo do Poluição do Ar](#)

No Quadro 4.8, apresentamos o recorte da programação do movimento do pássaro no jogo “Poluição do Ar” e a descrição realizada por Azevedo (2017), rerepresentada em Azevedo et. al (2018, p. 957).

Quadro 4.8 – Ideias e conteúdos matemáticos mobilizados no jogo “Poluição do Ar”.

Programação em Scratch (Comandos de programação)	Matemática (Conceitos e algumas ideias)
	O cenário do jogo é estruturado em plano cartesiano (2 dimensões). As suas dimensões são: $-240 \leq x \leq 240$ (eixo horizontal – comprimento) $-180 \leq y \leq 180$ (eixo vertical – altura ou largura). Para localizar qualquer personagem do jogo no cenário, utilizam-se as coordenadas cartesianas ou pares ordenados (x, y). Por exemplo, podemos localizar o pássaro (personagem) na seguinte posição (240, -180), onde $x = 240$ e $y = -180$.
	MOVIMENTO DO PÁSSARO Para mover o pássaro (para direita, para esquerda, para cima e para baixo), é preciso utilizar a ideia de coordenadas cartesianas (x, y). Utilizam-se, também, números positivos, negativos, além do zero (números inteiros) para indicar se o pássaro vai à direita ou à esquerda. À esquerda -10 e à direita 10, utilizando a variável x. Enquanto para cima ou para baixo, utiliza-se a variável y.

Fonte: Azevedo et. al (2018, p. 957).

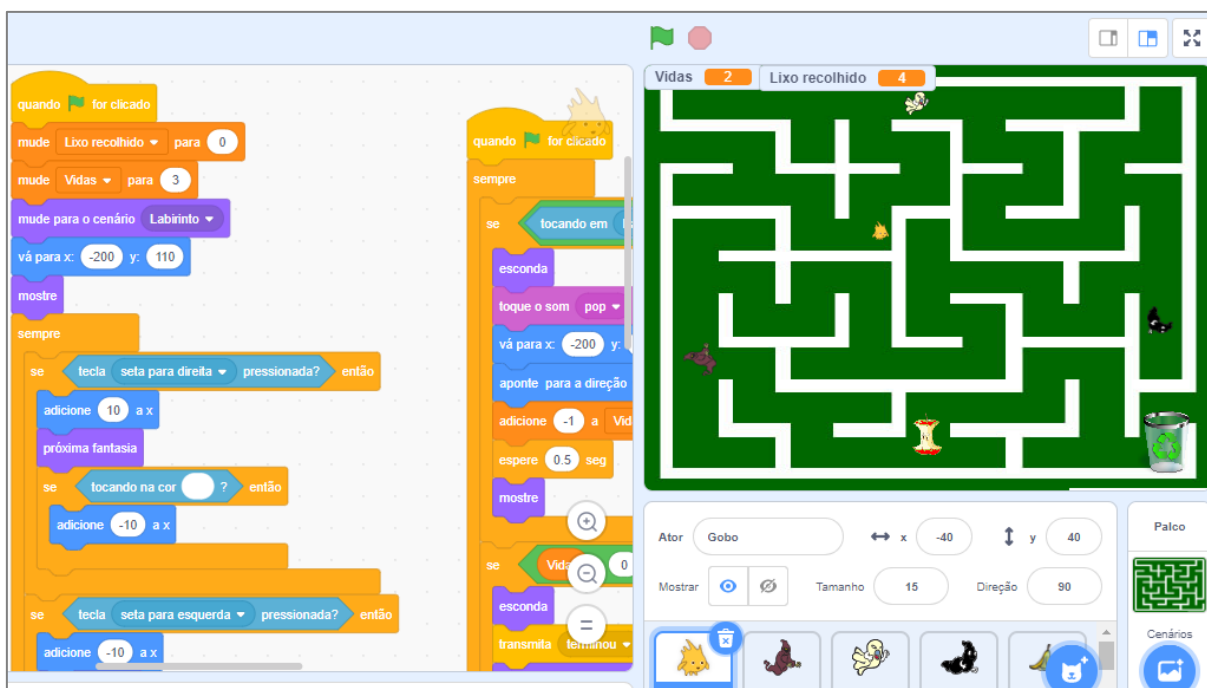
Link para acessar o projeto na comunidade *on-line* do Scratch:



4.3.2 Jogo “Labirinto: Recolhendo o lixo”

O principal objetivo do jogo é recolher todos os lixos espalhados pelo chão do labirinto, sem tocar nos fantasmas. Se o jogador conseguir reuni-los e caminhar até o final do labirinto, onde se encontra a lixeira, vencerá o jogo, caso contrário, não. Além disso, cada vez que o jogador tocar em algum fantasma voltará para a posição inicial e se tocar mais de três vezes em um deles perderá o jogo.

Figura 4.15 – Projeto no Scratch intitulado “Jogo ‘Labirinto: Recolhendo o lixo’”.



Fonte: Das autoras (2022).

Para consultar este projeto na comunidade *on-line* do Scratch, acesse:





COMO UTILIZAR O SCRATCH NA PERSPECTIVA INCLUSIVA

Até o momento, apresentamos elementos que podem ser considerados ao planejar aulas utilizando o Scratch, buscando garantir o acesso de todas as estudantes e todos os estudantes às atividades propostas.

Ressaltamos a necessidade de considerar as características das estudantes e dos estudantes envolvidos no processo educativo, para que o planejamento seja coerente e possam ser feitas as adaptações necessárias.

Conforme mencionamos no Capítulo 2 deste produto educacional, os cenários para investigação inclusivos compartilham a ideia principal do design universal. Portanto, o ambiente precisa ser acessível a todas as pessoas e oferecer uma variedade de desafios que levem em conta as diferenças entre as pessoas.

Veja, a seguir, as sugestões de leituras que podem auxiliar a pensar em como utilizar o Scratch na perspectiva da Educação Matemática Inclusiva.

5.1 Estudantes de diferentes níveis de ensino

No artigo intitulado **Investigando as possibilidades do Scratch para o ensino e a aprendizagem de conceitos matemáticos em cenários inclusivos**, Moraes, Silva e Fernandes (2019) apontaram que o Scratch pode ser utilizado em diversos segmentos de ensino e permite atender às demandas de estudantes *eficientemente diferentes*⁷ nas mais diversas esferas da educação, além de explorar conceitos matemáticos de uma forma não tradicional. As autoras destacam, ainda, que o uso do Scratch contribuiu para o desenvolvimento de habilidades essenciais a formação integral das estudantes e dos estudantes “como criatividade, iniciativa, curiosidade, motivação, autonomia, observação de regularidades, estímulo ao

⁷ Termos utilizado pelas autoras do artigo mencionado.

trabalho em grupo, colaborativo, valorização da diversidade, dentre outros” (MORAIS; SILVA; FERNANDES, 2019, p. 13).

5.2 Estudantes Surdas e Surdos

Quanto às percepções relacionadas ao uso do Scratch para o processo de ensino e aprendizagem de pessoas Surdas, podemos destacar a potencialidade desse *software* devido aos recursos visuais que são fundamentais considerando essa deficiência específica. Campello (2008, p. 128) aponta que:

os aspectos da visualidade na educação de Surdos, ou pedagogia surda é assim denominada considerando-se que a mesma pode ser compreendida como aquela que se ergue sobre os pilares da visualidade, ou seja, que tem no signo visual seu maior aliado no processo de ensinar e aprender.

Dessa forma, o uso do Scratch tem potencialidades ao ensino de estudantes Surdas e Surdos justamente em virtude de ser um *software* que tem sua funcionalidade construída por meio de recursos visuais.

5.3 Estudantes com Deficiência intelectual

Na Dissertação de Mestrado intitulada **A aprendizagem do conceito científico de fração por alunos com deficiência intelectual: os resultados de uma intervenção**⁸, Rodrigues (2017) relata que realizou uma intervenção pedagógica com duas estudantes que possuíam deficiência intelectual e apresentavam dificuldades em Matemática, principalmente em frações. A autora criou um jogo intitulado “Frações no Scratch”, utilizando duas ferramentas mediadoras: a linguagem e a imagem. Como resultado, a autora apontou que uma mudança nos procedimentos de ensino, incluindo a mediação docente, pautada no trabalho coletivo e na crença do potencial das estudantes e dos estudantes contribuiu para o desenvolvimento dos conceitos científicos das estudantes com deficiência intelectual relacionados às frações, além de provocar, no espaço escolar, reflexões sobre a proposta inclusiva.

⁸ Disponível em: <https://dspace.unipampa.edu.br/handle/riu/2557>.

No trabalho intitulado **Contribuições do software Scratch para aprendizagem de crianças com Deficiência Intelectual**⁹, Zednik et al. (2019) relataram uma experiência que envolveu o desenvolvimento de jogos utilizando o Scratch, como estratégia didática para potencializar o aprendizado de crianças com Deficiência Intelectual (DI) na sala de Atendimento Educacional Especializado (AEE). Com esses jogos, “buscou-se desenvolver habilidades de atenção, concentração, percepção, coordenação motora fina e ampla por meio de jogos personalizados às necessidades específicas dos alunos” (ZEDNIK et al., 2019, p. 394). Como resultados, Zednik et al. (2019) apontaram que o uso de jogos auxilia na aprendizagem de crianças com DI e que podem ter um impacto emocional positivo para as estudantes e os estudantes.

As autoras apontaram, ainda, que o *software* Scratch pode auxiliar na aprendizagem das estudantes e dos estudantes com deficiência intelectual ao permitir a criação de diversos recursos que podem despertar-lhes o interesse. Além disso, a construção dos jogos “pode ser direcionada a atender demandas específicas e criar situações de intervenções de acordo com as potencialidades e necessidades de cada aluno” (ZEDNIK et al., 2019, p. 397).

Também é possível desenvolver os jogos considerando o ritmo de cada estudante, fazendo adaptações conforme a necessidade de mais tempo ou menos tempo para alcançar os objetivos do jogo.

5.4 Estudantes Cegas e Cegos

Quanto à questão da utilização do Scratch por estudantes Cegas e Cegos, o fato de ser uma programação visual, num primeiro momento, impossibilitaria que pessoas Cegas utilizassem esse *software* sozinhas, pois, para realizar a programação, era necessário arrastar para a área de edição os blocos de código utilizando o *mouse*, o que não favorecia a utilização desse *software* por pessoas Cegas com autonomia. Uma das possibilidades para superar esse obstáculo seria a realização das atividades em duplas ou pequenos grupos, para que estudantes

⁹ Disponível em: <https://br-ie.org/pub/index.php/wie/article/view/8526>.

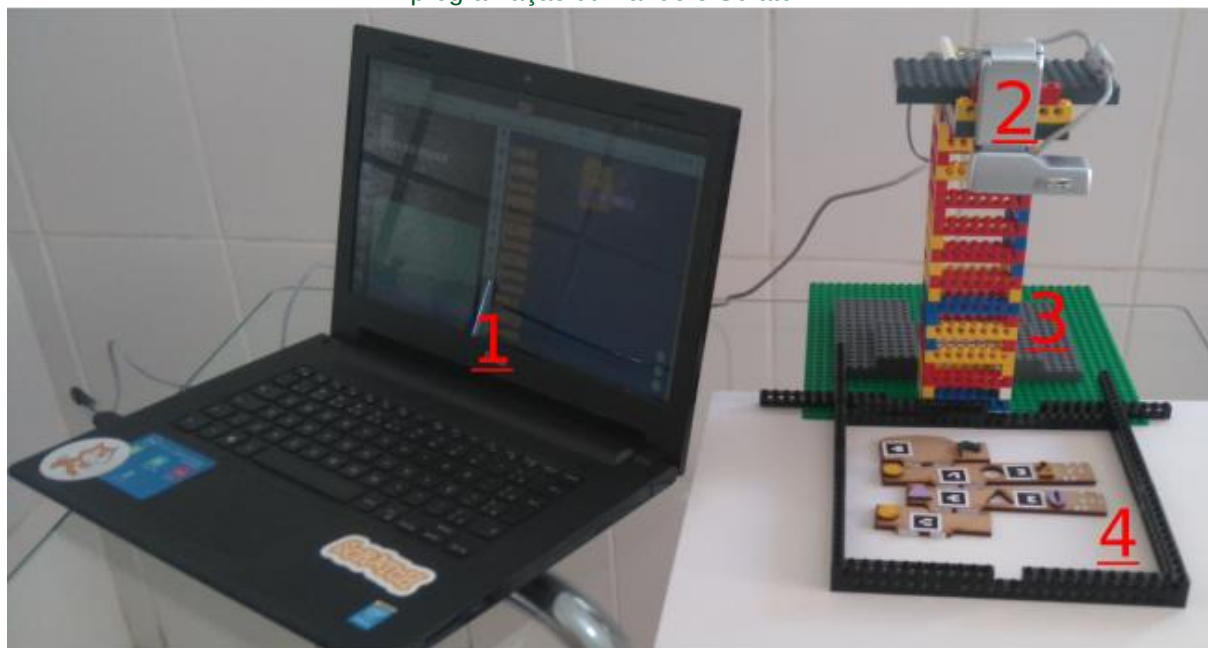
videntes auxiliassem estudantes Cegas e Cegos durante as atividades de programação, em um cenário investigativo inclusivo e colaborativo. Outra possibilidade seria que pessoas cegas utilizassem projetos desenvolvidos pelas professoras e pelos professores no Scratch, como um jogo de perguntas e respostas, por exemplo, que poderia ser utilizado em sala de aula por pessoas cegas e videntes em conjunto.

Também é possível pensar em outras formas de utilizar o Scratch de modo que as pessoas Cegas poderiam participar das atividades com o aplicativo com mais autonomia. Na Dissertação de Mestrado intitulada **Recomendações para o desenvolvimento de ambientes de programação inclusivos para crianças cegas**¹⁰, Angelo (2018) defende a necessidade de criar possibilidades para a utilização do Scratch por crianças Cegas. A autora desenvolveu e testou um sistema como exemplo para incluir crianças Cegas em atividades de introdução à programação utilizando o Scratch. Entre as recomendações apontadas pela autora, sugere-se que esse sistema seja um ambiente de linguagem de programação tangível e que possua Desenho Universal.

Apresentamos, a seguir, a imagem do protótipo do sistema desenvolvido por Angelo (2018), que mostra o computador (1) e o suporte montado de lego (3) com a *webcam* (2) que registra as imagens da área de código em uma superfície plana delimitada (4):

¹⁰ Disponível em: <https://teses.usp.br/teses/disponiveis/3/3142/tde-27082018-090956/en.php>. Acesso em: 14 abr. 2021.

Figura 5.1 – Protótipo do sistema para incluir crianças cegas em atividades de introdução à programação utilizando o Scratch.



Fonte: Angelo (2018, p. 58).

Sugerimos que esse trabalho seja consultado na íntegra para melhor entendimento do assunto. Essa ideia está de acordo com a busca por possibilidades para a utilização do Scratch na perspectiva da Educação Matemática Inclusiva.



CONSIDERAÇÕES FINAIS

Neste Produto Educacional, apresentamos as funcionalidades do *software* Scratch, dicas, sugestões e elementos que podem ser considerados ao planejar aulas utilizando o Scratch com o objetivo de possibilitar o acesso de todas as estudantes e todos os estudantes às atividades propostas.

Destacamos que, ao propor atividades com o Scratch em sala de aula, a cada ideia apresentada pelas estudantes e pelos estudantes, as professoras e os professores poderão usar diferentes estratégias e diferentes comandos, pois não há uma única maneira de resolver as situações-problema que surgirem.

Desse modo, será necessário parar para refletir como os jogos poderiam ser feitos utilizando essas ideias diferentes, considerando as estratégias das estudantes e dos estudantes e buscando caminhos para auxiliá-los a encontrar a solução sem fornecer as respostas prontas.

Nesse sentido, entendemos que as professoras e os professores precisam conhecer o *software* antes de utilizá-lo em sala de aula, mas, ainda assim, sempre poderão deparar-se com desafios que podem ser superados ao assumirem uma postura de aprendizes ao invés de imporem seu modo de resolver os problemas.

Este trabalho não para por aqui, pois estamos em constante aprendizado. Esperamos que as sugestões aqui apresentadas sejam um estímulo para o uso do Scratch em sala de aula na perspectiva inclusiva, bem como para o desenvolvimento de novas práticas pedagógicas e pesquisas sobre essa temática.

Sugerimos, ainda, que as percepções das estudantes e dos estudantes que participarem das atividades sejam consideradas, para que possamos ampliar as possibilidades para que todas as pessoas com e sem deficiência se desenvolvam, aprendam e conquistem autonomia.

REFERÊNCIAS

ANGELO, Isabela Martins. **Recomendações para o desenvolvimento de ambientes de programação inclusivos para crianças cegas**. Dissertação (Mestrado). 125 p. Universidade de São Paulo. São Paulo. 2018. Disponível em: <https://teses.usp.br/teses/disponiveis/3/3142/tde-27082018-090956/en.php>. Acesso em: 14 abr. 2021.

AZEVEDO, Greiton Toledo de. 2017. **Construção de conhecimento matemático a partir da produção de jogos digitais em um ambiente construcionista de aprendizagem**: possibilidades e desafios. 2017. 235 p. Dissertação (Mestrado em Educação em Ciências e Matemática) – Programa de Pós-Graduação em Educação em Ciências e Matemática, Universidade Federal de Goiás, Goiânia. Disponível em: <https://repositorio.bc.ufg.br/tede/handle/tede/7415>. Acesso em: 10 mar. 2020.

AZEVEDO, Greiton Toledo de, et. al. Produção de Games nas Aulas de Matemática: Por que não?. **Revista Acta Scientiae**, v. 20, p. 950-966, 2018. Disponível em: <http://www.periodicos.ulbra.br/index.php/acta/article/view/4152>. Acesso em: 14 ago. 2020.

BRASIL. Ministério da Educação. Secretaria de Educação Especial. **Política Nacional de Educação Especial na Perspectiva da Educação Inclusiva**. Inclusão: revista da educação especial, v. 4, n 1, janeiro/junho 2008. Brasília: MEC/SEESP, 2008. Disponível em: <http://portal.mec.gov.br/seesp/arquivos/pdf/revinclusao5.pdf>. Acesso em: 14 ago. 2020.

CAMPELLO, Ana Regina e Souza. **Aspectos da visualidade na educação de surdos**. 2008. 245 f. Tese (Doutorado em Educação) – Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <https://repositorio.ufsc.br/handle/123456789/91182>. Acesso em: 14 abr. 2021.

FERNANDES, Solange Hassan Ahmad Ali. Educação Matemática Inclusiva: adaptação x construção. **Revista Educação Inclusiva - REIN**, Campina Grande, PB, v. 1. 01, n. 01, p.78-95, julho/dezembro-2017. Disponível em: <http://novo.revista.uepb.edu.br/index.php/REIN/article/download/68/52/215>. Acesso em: 14 ago. 2020.

FIGUEIRAS, Lourdes. HEALY, Lulu. SKOVSMOSE, Ole. Difference, inclusion, and mathematics education: launching a research agenda. **International Journal of Studies in Mathematics Education**. 9(3), 15–35, 2016. Disponível em:

<https://seer.pgsskroton.com/index.php/jieem/article/view/4588>. Acesso em: 10 set. 2021.

MORAIS, Tula Maria Rocha; SILVA, Elizabete Leopoldina da; FERNANDES, Solange Hassan Ahmad Ali. Investigando as possibilidades do Scrath para o ensino e a aprendizagem de conceitos matemáticos em cenários inclusivos. In: **Encontro Nacional de Educação Matemática**, XIII, Mato Grosso, 2019. Disponível em: <https://www.sbemmatogrosso.com.br/eventos/index.php/enem/2019/paper/view/579/1078>. Acesso em: 30 abr. 2021.

RESNICK, Mitchel. Learn to Code Code to Learn. **EdSurge**, mai. 2013. Disponível em: <http://web.media.mit.edu/~mres/papers/L2CC2L-handout.pdf>. Acesso em: 10 mar. 2020.

RODRIGUES, Sastria de Paula. **A aprendizagem do conceito científico de fração por alunos com deficiência intelectual**: os resultados de uma intervenção. Relatório Crítico-Reflexivo (Mestrado em Educação). 168 p. Universidade Federal do Pampa. Jaguarão. 2017. Disponível em: <https://dspace.unipampa.edu.br/handle/riu/2557>. Acesso em: 10 mar. 2020.

SCRATCH. **Comunidade on-line**. Crie histórias, jogos e animações. Partilhe com outros em todo o mundo. Disponível em: <https://scratch.mit.edu/>. Acesso em: 20 jul. 2020.

SILVA, Admilson Iaresk da. **Trabalhando matemática com o Scratch**. 2020. 55 p. Produto Educacional (Mestrado em Formação Científica, Educacional e Tecnológica) - Universidade Tecnológica Federal do Paraná, Curitiba, 2020. Disponível em: http://repositorio.utfpr.edu.br/jspui/bitstream/1/4829/2/CT_PPGFCET_M_Silva%2C_Admilson_iaresk_da_2020_1.pdf. Acesso em: 14 ago. 2020.

SKOVSMOSE, Ole. Inclusões, encontros e cenários. **Educação Matemática em Revista**, Brasília, v. 24, n. 64, p. 16-32, set./dez. 2019. Disponível em: http://funes.uniandes.edu.co/24127/1/Skovsmose2019_Inclus%C3%B5es.pdf. Acesso em: 20 dez. 2021.

SOCIEDADE BRASILEIRA DE EDUCAÇÃO MATEMÁTICA. **GT13 - Diferença, Inclusão e Educação Matemática**. Brasil, 2012. Disponível em: <http://www.sbembrasil.org.br/sbembrasil/index.php/grupo-de-trabalho/gt/gt-13>. Acesso em: 22 set. 2021.

ZEDNIK, Herik et. al. Contribuições do Software Scratch para Aprendizagem de Crianças com Deficiência Intelectual. In: WORKSHOP DE INFORMÁTICA NA ESCOLA, 25., 2019, Brasília. **Anais do XXV Workshop de Informática na Escola**. Porto Alegre: Sociedade Brasileira de Computação, 2019. p. 394-403. DOI: <https://doi.org/10.5753/cbie.wie.2019.394>.

A RESPEITO DAS AUTORAS

Ana Clara Dias Bernardino

Possui Graduação em Matemática pela Universidade Federal de Lavras (2015); Especialização em Gestão de Tecnologia da Informação pela Faculdade Única de Ipatinga (2019); Mestrado em Ensino de Ciências e Educação Matemática pela Universidade Federal de Lavras (2022). É pesquisadora com interesse em pesquisas na área de Educação Matemática, atuando principalmente nos seguintes temas: Educação Matemática Inclusiva, educação mediada por Tecnologias Digitais de Informação e Comunicação, aprendizagem criativa, pensamento computacional, construção de jogos digitais, potencialidades do ensino de programação para o processo de ensino e aprendizagem da Matemática.

Rosana Maria Mendes

Possui graduação em Matemática pelo Centro Universitário Assunção (2000); mestrado em Educação pelo Programa de Estudos Pós-Graduados Em Educação pela Universidade São Francisco - Itatiba (2006) e doutorado em Educação Matemática pelo Programa de Pós-Graduação em Educação Matemática, área de Concentração em ensino e aprendizagem da matemática e seus fundamentos filosófico-científicos pela Universidade Estadual Paulista "Julio de Mesquita Filho", UNESP/Rio Claro (2013). Atualmente está trabalhando na Universidade Federal de Lavras (UFLA), no Departamento de Educação em Ciências Físicas e Matemática (DFM). Tem experiência na área de Educação Matemática, atuando principalmente nos seguintes temas: Jogos, Tecnologia da Informação e Comunicação, Formação de Professores, Programa Institucional de Bolsa de Iniciação à Docência (Pibid), Educação Especial, Educação Inclusiva.

