

APRENDENDO MATEMÁTICA AO SE PROGRAMAR NO SCRATCH

LEARNING MATH AS YOU PROGRAM ON SCRATCH

Airan Priscila de Farias

Mestranda do Programa de Pós-Graduação em Ensino de Matemática da Universidade
Tecnológica Federal do Paraná
airan.p.farias@gmail.com

Marcelo Souza Motta

Doutor em Ensino de Ciências e Matemática
Professor Adjunto da Universidade Tecnológica Federal do Paraná
marcelomotta@utfpr.edu.br

Resumo

Este artigo tem como objetivo apresentar a ferramenta de programação Scratch, destacando o seu potencial na aprendizagem de conceitos matemáticos. A pesquisa caracterizou as principais ferramentas do software e suas principais contribuições ao desenvolvimento das competências e habilidades essenciais ao desenvolvimento da fluência e criticidade tecnológica. O quadro teórico buscou aporte de pesquisadores das tecnologias educacionais e da aprendizagem mediada pelo uso do computador. Os principais apontamentos deste estudo mostram que o Scratch pode criar um espaço educativo, em que o aluno é agente ativo na construção de sua aprendizagem, e o ambiente criado torna a transição entre os conceitos matemáticos e a programação um espaço lúdico e colaborativo.

Palavras-Chave: Ensino de Matemática; Software de Programação; Scratch.

Abstract

This article aims to present the Scratch programming tool, highlighting its potential in learning mathematical concepts. The research characterized the main tools of the software and its main contributions to the development of the skills and abilities essential to the development of technological fluency and criticality. The theoretical framework sought the contribution of researchers of the educational technologies and of the learning mediated by the use of the computer. The main notes of this study show that Scratch can create an educational space, in which the student is an active agent in the construction of his learning, and the created environment makes the transition between mathematical concepts and programming a playful and collaborative space.

Keywords: Mathematics Teaching; Programming Software; Scratch.

1 Introdução

A ideia do uso de computadores como recurso didático não é atual. Essa concepção vem sendo idealizada por estudiosos em educação como José Armando Valente e o precursor desta ideia Seymour Papert antes mesmo dos computadores se tornarem ferramentas comuns no dia a dia das pessoas.

O computador não é mais um privilégio de poucos e são variados os recursos que ele disponibiliza. É comum vermos em todas as partes as pessoas com seus computadores pessoais, *smartphones* ou *tablets*, tornando-os uma extensão de seu próprio corpo. É quase que estranho quando nos deparamos com alguém que não compartilha desses equipamentos.

Com isso, percebemos a mudança cultural pela qual passa a sociedade, em que o mundo, cada vez mais globalizado, tem suas barreiras de espaço e tempo ampliadas pelos ambientes virtuais.

Contudo, embora estejamos vivenciando uma imersão tecnológica em nosso cotidiano, quase não vemos uma presença significativa da tecnologia digital dentro das salas de aula. Isso pode ser confirmado observando os cursos do Ensino Superior, falamos aqui, especificamente, dos cursos de Licenciatura em Matemática, em que alguns adotam aulas de “mídias tecnológicas” ou levando outros nomes, como um curso que possui uma disciplina inovadora. No entanto, não é! Pois, a presença dessas disciplinas nos mostra que mesmo em tempos em que as tecnologias digitais são partes integrantes das nossas vidas elas não são tratadas como algo natural no ambiente educacional.

Saindo dos centros de formação de professores e adentrando nos muros das escolas da educação básica encontramos os laboratórios de informática, em que muitas vezes são usados de forma tradicional, isto é, apenas informatizando formas tradicionais de ensino. Isto pode ser justificado, pelo tipo de formação tecnológica que o professor vem recebendo, na formação inicial ou continuada.

Logo, percebemos que a tecnologia na educação ainda é usada de uma forma “antinatural” na escola, isto é, diferente da forma como a usamos em nosso cotidiano, que vai do entretenimento ao trabalho ou da comunicação interpessoal à fonte de informações. Os Parâmetros Curriculares Nacionais (PCN) (BRASIL, 2000) afirmam que o impacto das tecnologias no cotidiano das pessoas vai exigir competências e habilidades que vão além do simples manuseio de equipamentos digitais.

Para os PCN (BRASIL, 2000), esse impacto exige que a Matemática seja reorganizada, de forma que o uso do computador em sala de aula proporcione ao indivíduo o aprimoramento de habilidades que o conduza de forma significativa em uma sociedade em constante modificação social e tecnológica.

Nesse contexto, destacamos o objetivo central deste artigo, que é descrever as possibilidades de utilização do software de programação Scratch e suas contribuições ao desenvolvimento da aprendizagem matemática.

O Scratch é um software de programação voltado para a criação de projetos interativos com recursos multimídia, de forma contextualizada e motivadora, que possibilita o desenvolvimento da fluência tecnológica, habilidade essencial a ser desenvolvida na formação do cidadão atual.

Acreditamos que a alteração de um paradigma educacional é fortemente produzida pela mudança de paradigma do professor, por isso, trazemos um pouco do software de programação Scratch, como mais uma ferramenta a ser adicionada no variado conjunto de recursos digitais disponíveis para uso didático do professor de Matemática.

2 Conhecendo o Scratch

Com a introdução dos computadores nas escolas na década de 70, houve grande movimentação e entusiasmo em relação à programação. Nessa época, Papert apresentou a linguagem de programação LOGO como um pilar para se repensar abordagens pedagógicas e a forma de aprendizagem. O computador viria como uma ferramenta em que as crianças poderiam fazer construções que representassem seus pensamentos matemáticos.

Com o tempo o computador passou a ser cada vez mais comum na vida das pessoas, principalmente, na vida das crianças desde a mais tenra idade, sendo uma forma de comunicação forte, entretenimento, fonte de informação e aprendizado informal¹.

Contudo, o uso do computador na educação não representou uma mudança de paradigma, pois, continuou a ser usado como “máquina de ensinar”, ou seja, sua aplicação

¹ Referimo-nos ao aprendizado informal como sendo àquele que aprendemos fora do ambiente formal de ensino como escolas e universidades.

ficou restrita a uma metodologia tradicional de ensino, em que o aluno recebe o conhecimento de forma direta e instrucional.

Neste contexto, em resposta à distância entre os avanços tecnológicos e a fluência tecnológica dos cidadãos, o *Lifelong Kindergarten Group / MIT Media Lab do Massachusetts Institute of Technology* (MIT) criou em 2007, o Scratch como um recurso para minimizar essa distância.

O nome “Scratch” vem da técnica utilizada pelo *hip-hop* e *disc jockeys* em mixar músicas com discos de vinil movimentando-os com as mãos para frente e para trás, e o software Scratch possui essa ideia, ao misturar diferentes estilos de mídia, como imagens, áudios, animações, fotos e músicas. Trata-se de um software de programação visual voltado para a criação de projetos interativos com recursos multimídia, tais como: jogos, histórias animadas, simulações, dentre outros, podendo ser usado por pessoas de todas as idades, promovendo um rico ambiente de aprendizagem e desenvolvendo o pensamento criativo e sistemático em diversas áreas do conhecimento.

Neste ponto, Resnick et al. (2009) aponta o Scratch como uma possibilidade de aprendizagem de conceitos matemáticos de forma contextualizada e motivadora contribuindo com a fluência tecnológica dos usuários. Desta forma, acreditamos que o professor, ao propor atividades com o uso de softwares de programação, pode proporcionar ao aluno um ambiente de aprendizagem que enfatiza os significados dos conceitos, promovendo a criticidade e a ludicidade. Isso se justifica pela forma contextualizada e motivadora de aprendizagem que ocorre durante a utilização do software. Dizemos aqui motivadora no sentido da teoria construcionista defendida por Papert, em que a motivação está no engajamento do aluno, ao produzir um artefato de seu interesse.

De acordo com Marji (2014), o Scratch desenvolve habilidades relacionadas à resolução de problemas, as quais tem sua relevância destacada não só na programação, mas em todos os âmbitos da vida. Por conseguinte, visto que a resolução de problemas está intrinsicamente ligada ao ensino da Matemática, e que a linguagem de programação na construção de jogos digitais proporciona a criação de um ambiente de aprendizagem construcionista promovendo o ciclo descrição-execução-reflexão-depuração apresentado por Valente (1998).

Partilhamos do pensamento de que o uso do Scratch pode potencializar o ensino e a aprendizagem de conceitos matemáticos motivando a busca pelo conhecimento e incentivando o aprendizado prático e autônomo por meio da exploração e da descoberta.

Apresentaremos a seguir uma breve descrição das principais ferramentas e comentaremos uma possível aplicação do Scratch no ensino de conceitos matemáticos.

2.1 Tela Inicial do Scratch

Há duas formas de acesso ao software. Uma é *online*, acessando o site do Scratch em www.scratch.mit.edu, na aba “Criar”, conforme descrito na Figura 01. O interessante ao se criar projetos no Scratch *online* são os recursos que o *site* oferece. O usuário cria uma conta e pode compartilhar seus projetos e usar projetos de outros usuários, além de que, ao estar *logado* na página, todos os projetos são salvos, automaticamente, em sua conta.



Figura 01 - Página inicial do site do Scratch
Fonte: www.scratch.mit.edu

Caso não haja acesso à internet, é possível utilizar o Scratch no modo off-line. Para isso, basta fazer o download do aplicativo em <https://scratch.mit.edu/scratch2download/>. Vale ressaltar que os projetos criados off-line podem ser carregados na plataforma online e compartilhados no site.

A Figura 02 ilustra a tela inicial do Scratch em que estão disponíveis todas as ferramentas de interação com o software. A programação se dá pelo encaixe dos blocos de comandos, o que elimina erros de sintaxe presentes em outras linguagens de programação.

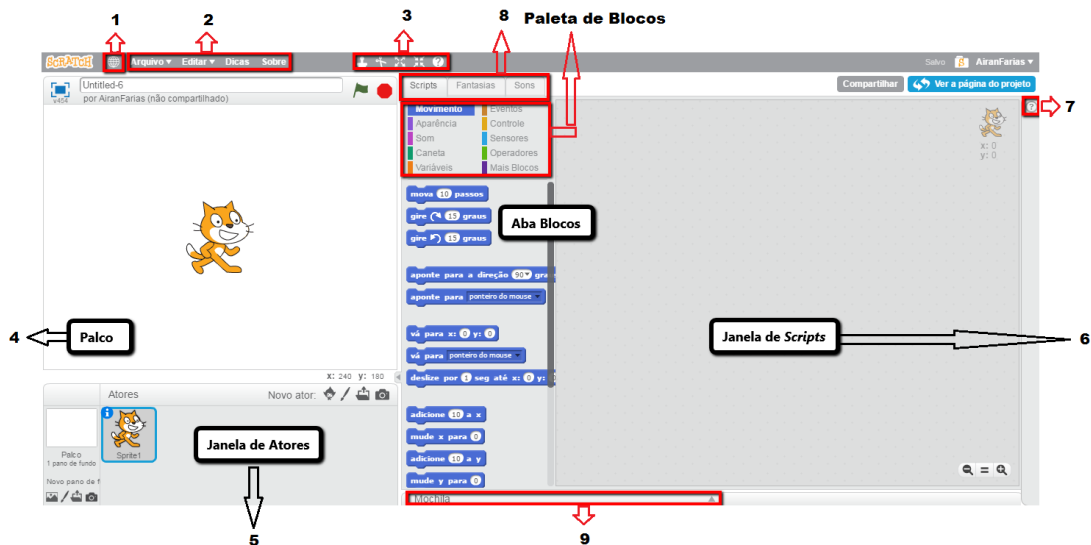


Figura 02 - Interface inicial do Scratch
 Fonte: www.scratch.mit.edu

1. **Idioma:** no ícone em forma de globo pode ser alterado o idioma do software. Vários idiomas estão disponíveis, sendo o inglês o idioma original. Isto justifica algumas ações estarem somente em inglês, como por exemplo, a janela de dicas;
2. **Barra de Menus:** Abre uma nova janela para criar um novo projeto, importar um projeto salvo no computador, salvar, compartilhar no site do Scratch, verificar as atualizações e sair do software;
3. **Ferramentas de Cursor:** duplicar algum comando ou imagem, apagar, diminuir ou aumentar atores e imagens;
4. **Palco:** é o local onde os atores são desenhados, se movem e interagem. Ele sempre é iniciado em branco e com um único ator, o Gato;
5. **Lista de Atores:** exibe a miniatura de todos os atores do projeto e seus respectivos nomes e informações ao clicar no ícone “i” no canto superior esquerdo da miniatura do ator. A escolha do ator pode ser feita no acervo do Scratch, criado no *Paint Editor* do Scratch, carregado de um arquivo do computador ou uma fotografia. (Ver Figura 03).



Figura 03 – Botões da Lista de Atores
 Fonte: recorte do software Scratch elaborado pelos autores.

6. Janela de Scripts: é o espaço em que programamos o que os atores farão. Os scripts são construídos arrastando os blocos de comandos para essa área e encaixando-os como blocos de montar;

7. Janela de Dicas: disponibiliza algumas sugestões do que é possível fazer com o software;

8. Abas: este espaço está dividido em três partes: Scripts, Fantásias e Sons. A aba Scripts contém a Paleta de Blocos, que está subdividida em dez paletas, diferenciadas por cores para melhor identificação, sendo elas: Movimento, Aparência, Som, Caneta, Variável, Eventos, Controle, Sensores, Operadores e Mais Blocos, como mostra a Figura 04.



Figura 04 – Blocos de comandos na aba Scripts
 Fonte: recorte do software Scratch elaborado pelos autores.

Já a aba Fantásias contém tudo o que é necessário para mudar a aparência do ator. A aparência do ator, chamada de fantasia pode ser modificada ou criada no *Paint Editor* do Scratch, ou ainda, selecionar um novo ator, como podemos observar na Figura 05 nos botões à esquerda da tela.

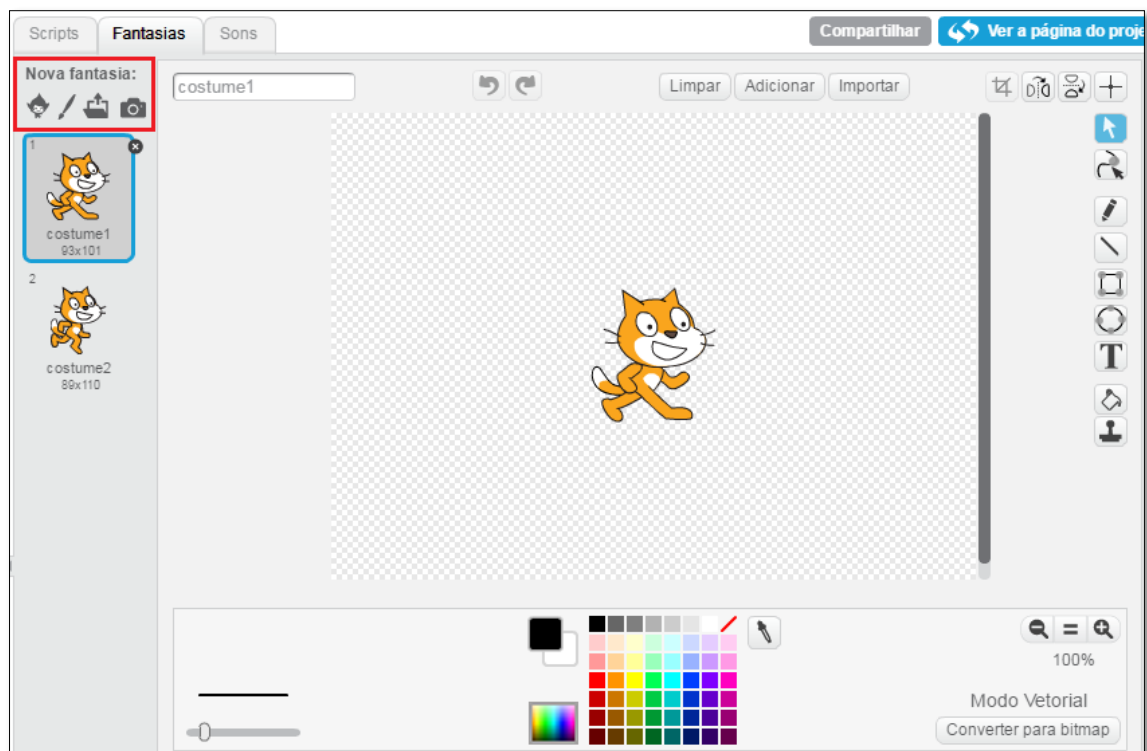


Figura 05 – Aba Fantasias. Tela do *Paint Editor* do Scratch
Fonte: recorte do software Scratch elaborado pelos autores.

E por fim, na aba Sons, contamos com os sons que podem ser adicionados aos atores, como ilustra a Figura 06.

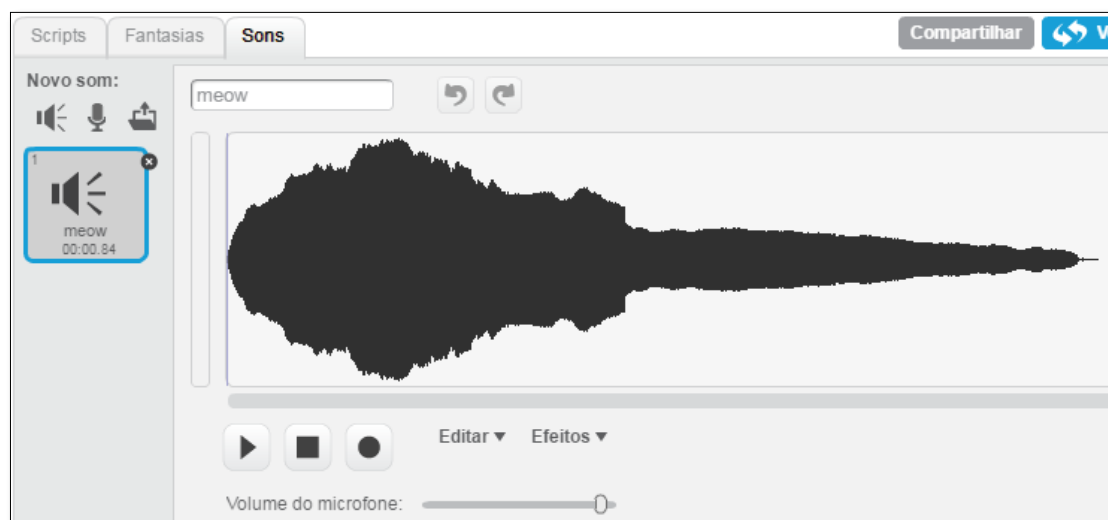


Figura 06 – Aba Sons
Fonte: recorte do software Scratch elaborado pelos autores.

9. Mochila: localizado na parte inferior direita da tela inicial. Neste espaço podem ser adicionados comandos ou atores para serem utilizados posteriormente, assim como projetos existentes no site do Scratch. Contudo, essa opção só é possível quando se está

logado na página do Scratch, ou seja, quando está criando o seu projeto online e não pelo editor off-line. (Ver Figura 07).

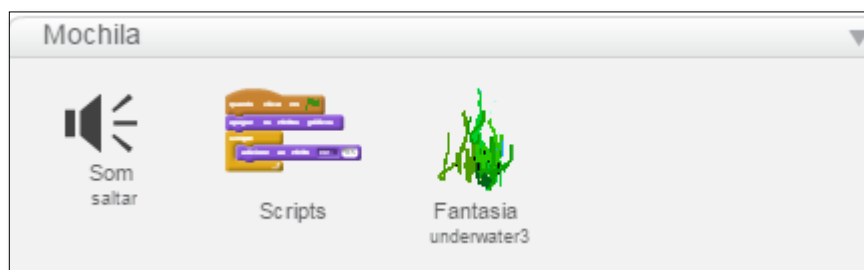


Figura 07 – Mochila

Fonte: recorte do software Scratch elaborado pelos autores.

O Scratch possui uma ampla possibilidade de trabalho e criações, desde histórias animadas à construção de gráficos. Embora seja um software de entretenimento também está voltado para fins educacionais, contando com uma comunidade de educadores pela página do ScratchEd em www.scratched.gse.harvard.edu, desenvolvido e comandado pela *Harvard Graduate School of Education*. Além de guias, tutoriais e dicas na própria página do Scratch para suporte ao educador.

O software já é usado por educadores em diferentes níveis de ensino, desde ao ensino Fundamental e Médio, assim como em algumas universidades, sendo utilizado em disciplinas de Letras, Ciências, Estudos Sociais, Matemática, Línguas Estrangeiras e Artes, como o próprio *site* do Scratch aponta, pois durante o processo de criação os alunos aprendem a pensar de maneira criativa, a raciocinar sistematicamente e a trabalhar colaborativamente, competências que são essenciais na sociedade contemporânea.

Considerando o ensino de Matemática, na aba **Scripts**, por exemplo, conforme a Figura 08, contamos com os blocos de comandos **Variáveis** e **Operadores**. As Variáveis são tratadas como um espaço em que armazena informações, e em Operadores podem ser encontrados mais de dez operações matemáticas. Também, há os blocos condicionais (se, então; senão) da paleta de blocos **Controle**.



Figura 08: Aba Blocos

Fonte: recorte do software Scratch elaborado pelos autores.

Explorando ferramentas

De acordo com Lima e Santos (2014, p. 313) o Scratch “é muito mais acessível do que outras linguagens de programação, isso acontece pela utilização de uma interface gráfica que permite que cada projeto seja criado como blocos de montar, lembrando o Lego²”. O software realmente foi desenvolvido pela observação dos pesquisadores em como as crianças brincavam com o LEGO. Resnick et. al (2009) aponta que no laboratório do *Lifelong Kindergarten Group*, por meio da robótica com o LEGO, os pesquisadores observaram, por anos, o comportamento das crianças. O autor descreve que ao dar uma caixa cheia das peças de LEGO para elas, imediatamente começam a mexer e a construir. Observaram que as ideias vão surgindo umas atrás das outras, enquanto planos e metas evoluem conjuntamente com as estruturas.

Os desenvolvedores do software almejavam que ao se programar em Scratch, a pessoa sentisse essa mesma sensação de liberdade de criação proporcionada pelo LEGO, de forma que a programação fosse acessível a pessoas leigas e crianças, como já acontecia com a linguagem LOGO, mas que não possuíssem os erros de sintaxe como acontecia com este. Assim, a estrutura de sequência de comandos no Scratch se dá pelo encaixe de blocos de forma gráfica e lúdica. A Figura 09 exemplifica uma construção feita no Scratch no estudo da classificação dos sólidos geométricos.

² LEGO é um brinquedo composto por peças de plástico que se encaixam, de forma que a montagem dessas peças permite a construção de muitas combinações.

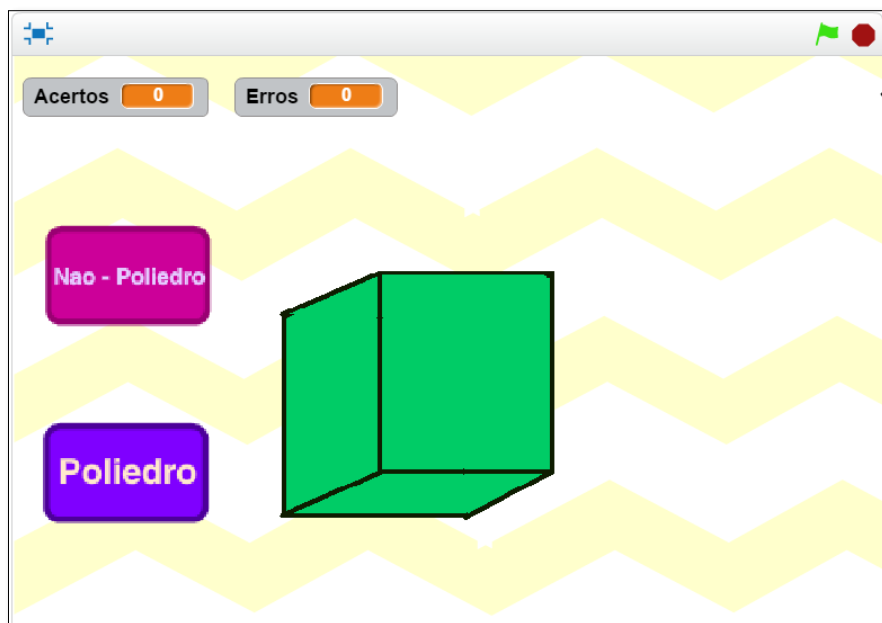


Figura 09: Apresentação do jogo Classificação de Sólidos Geométricos
Fonte: elaborado pelos autores.

Este é um jogo de classificação de sólidos geométricos e possui três atores, o botão Não-Poliedro, o botão Poliedro e o sólido geométrico. Este último possui cinco fantasias como mostra a Figura 10.

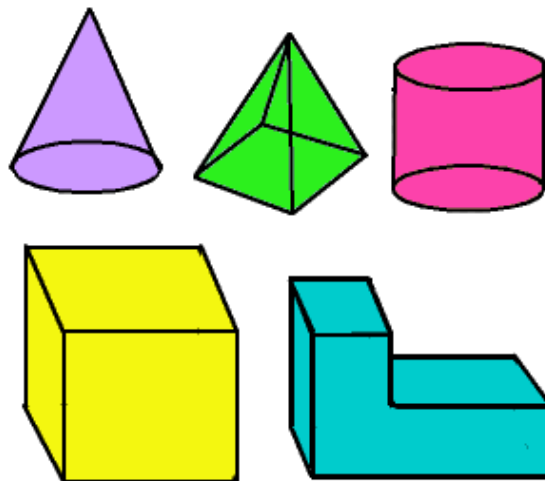


Figura 10: Fantasias do ator Sólido Geométrico
Fonte: elaborado pelos autores.

Os atores e suas fantasias podem ser construídos no *Paint Editor* do Scratch, como apresentamos na Figura 05. O Scratch, também possui um acervo de atores, que vai desde fotografias de pessoas e animais, até desenhos animados, letras do alfabeto, coisas da natureza e objetos em geral. Na Figura 11 podemos ver uma pequena parte desse acervo.



Figura 11: Acervo de atores e/ ou fantasias do Scratch
Fonte: recorte do software Scratch elaborado pelos autores.

O mesmo se configura para a definição do pano de fundo do Palco. Neste caso, a miniatura do Palco precisa estar selecionada, conforme a Figura 12.

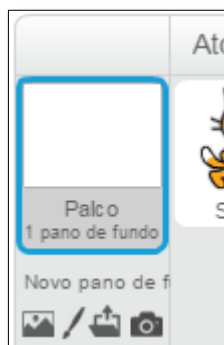


Figura 12: Miniatura do Palco selecionado
Fonte: recorte do software Scratch elaborado pelos autores.

Assim que a miniatura do Palco é selecionada a aba Fantasias muda para o nome **Panos de fundo**, como mostra a Figura 13, em que se pode trabalhar a fantasia do tema de fundo do Palco.

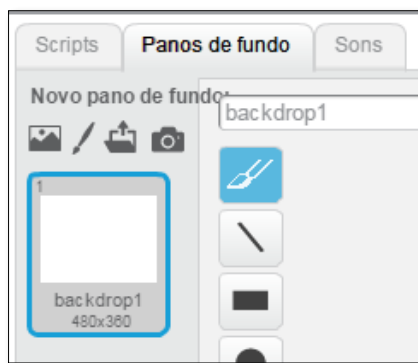


Figura 13: Aba Panos de fundo

Fonte: recorte do software Scratch elaborado pelos autores.

Os aspectos básicos de criação de um projeto estão em preparar o Palco escolhendo um pano de fundo e na definição dos atores que farão parte do projeto, pois são a eles que damos os comandos por meio dos *scripts*. Vale ressaltar, que os *scripts* não são exclusivos dos Atores, os panos de fundo também podem receber comandos, como a mudança de cenário no início e término de um jogo, por exemplo, ou ainda na transição da mudança de níveis em um jogo.

2.2 Construindo *scripts*

Os *scripts* são os comandos dados aos atores ou ao pano de fundo, ou seja, a programação. Esses comandos são construídos por meio do encaixe dos blocos, os quais devem ser arrastados para a Janela de *Scripts*, conforme destacamos na Figura 02. O processo de depuração do comando dado é realizado durante a construção do *script* ou da programação, o que significa que as correções das falhas ou a verificação positiva dos comandos podem ser feitas sem a necessidade de parar o *script*, promovendo um retorno imediato para o usuário, promovendo uma reflexão sobre os erros e acertos.

A Figura 14 ilustra a construção de um *script* para o ator Lápis. Note que toda a borda do *script* possui um leve brilho de cor amarelo. Isso significa que o comando está em execução. Contudo, quando o destaque amarelo não existir, significa que o *script* não está sendo executado ou que ele já terminou.

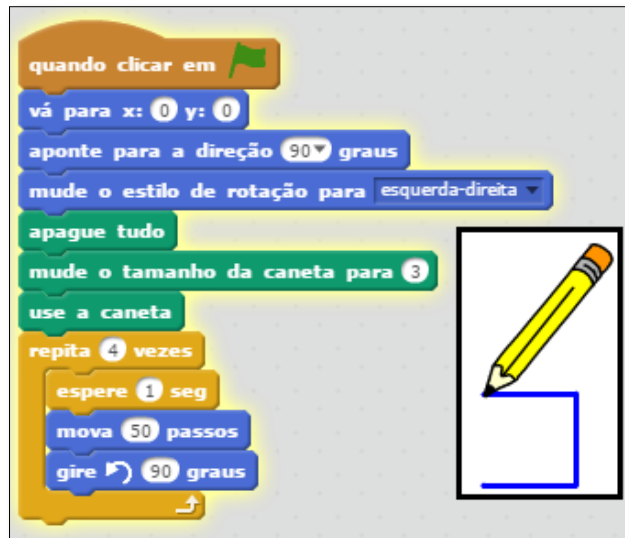


Figura 14: Script do desenho de um quadrado em execução
 Fonte: elaborado pelos autores.

Para que o *script* seja executado, basta clicar sobre ele, na Janela de *Scripts* ou clicar na bandeira verde na parte superior direita do Palco, conforme destacamos na Figura 15.



Figura 15: Execução do script pela
 Fonte: recorte do software Scratch

Bandeira verde
 elaborado pelos autores.

Observe na Figura 16, o procedimento realizado pelo Lápiz, e a maneira como ele é executado:



Figura 16: Execução do script que desenha um quadrado
 Fonte: elaborado pelos autores.

1. Quando o ícone da bandeira verde for clicado, o script iniciará;

2. O ator Lápis irá para a posição no placó de acordo com a coordenada $x = 0$ e $y = 0$;
3. A direção do ator Lápis será de 90° ;
4. Para que o ator Lápis não gire durante o desenho do quadrado, esse bloco determina que a sua rotação seja somente para a esquerda e para a direita;
5. Esse bloco garante que o programa sempre inicie em branco, ou seja, sem nenhum desenho anterior;
6. Esse bloco altera a espessura do traço da caneta;
7. O bloco **use a caneta** é o que permite que o ator Lápis deixe o rastro de seu movimento;
8. O bloco **repita** determina quantas vezes serão executados os comandos dentro dele. Esse bloco simplifica o comando de construção do quadrado;
9. São esses três blocos que constroem o quadrado.

Além da organização por categorias e cores da linguagem de programação, o Scratch também classifica os blocos em quatro tipos, conforme mostrado na Figura 17.

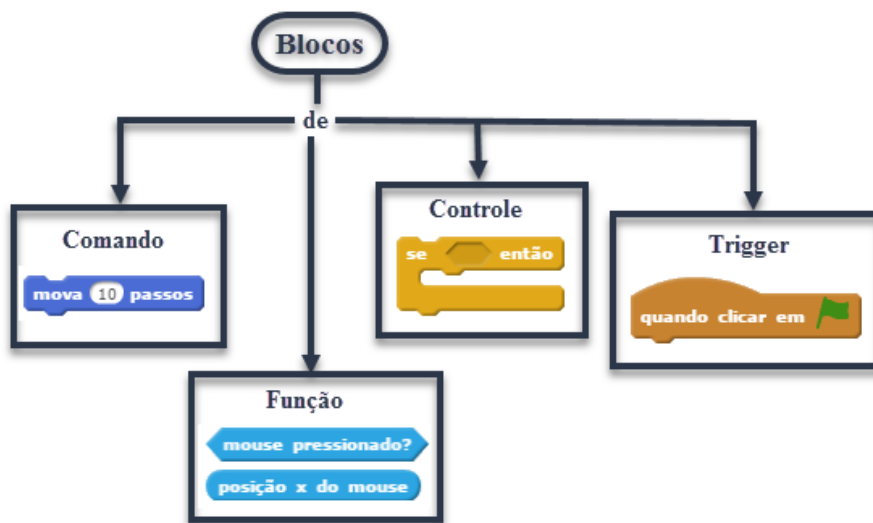


Figura 17: Tipos de blocos do Scratch
 Fonte: elaborado pelos autores.

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

Os **blocos trigger** também conhecidos como “chapéus” só podem ser usados no início de um script.

Já os **blocos de função** não possuem saliências e retornam um valor. Eles são usados como entradas para outros blocos, o que não permite serem usados sozinhos para compor um script.

Os blocos de função possuem duas características:

- os blocos com extremidades arredondadas retornam números ou *strings*³;
- os blocos com extremidades pontiagudas informam se algo é verdadeiro ou falso, retornam valores *Booleanos*.

A construção de um *script* demanda do construtor, planejamento e reflexão. A busca pelo objetivo final de uma criação em Scratch proporciona aprendizagens no decorrer da programação que podem ser amplamente exploradas pelo professor de acordo com os objetivos propostos pela atividade. E a proposta do aluno como programador e o professor como mediador e orientador da aprendizagem pode tornar essa experiência enriquecedora, do ponto de vista em que o aluno é construtor de seu próprio conhecimento e o educador um facilitador desse processo.

Concordamos com Valente (1998) ao apontar que o professor quando se utiliza de softwares como meio de ensino deve possuir os conhecimentos computacional, pedagógico e psicológico, pois o uso do computador por si não apresenta ganhos educacionais, mas sim pelo uso consciente deste pelo educador. E isso causa consequentemente uma formação consciente do educando.

Considerações finais

Falamos inicialmente um pouco a respeito do uso do computador na educação. Apontamos que embora o uso da tecnologia digital seja defendido na educação brasileira por mais de 30 anos, não vemos essa presença de forma marcante, principalmente, na busca por uma formação que prepare o cidadão para o século 21.

Dentro das tecnologias disponíveis para apoio e enriquecimento da prática docente foi apresentado o software de programação Scratch, que mesmo não sendo um software voltado para o ensino de Matemática, acreditamos que pode contribuir com o aprimoramento ou a aprendizagem de conceitos matemáticos.

³ *String* é um tipo de dado que serve para armazenar nomes, endereços, títulos de livros. Isto é, trata-se de uma sequência de caracteres incluindo letras, tanto maiúsculas quanto minúsculas, números e símbolos que podem ser digitados a partir do teclado do computador.

Uma proposta é o aluno como programador de jogos digitais e o professor como orientador desse processo. O processo de criação proporcionará ao aluno aprendizagens que estão, intrinsicamente, ligadas ao Scratch, como o desenvolvimento das competências de informação e comunicação, as competências de raciocínio e resolução de problemas as competências interpessoais e de autodirecionamento, competências essas essenciais para todos os indivíduos da sociedade contemporânea. Além de contribuir com o desenvolvimento da fluência e criticidade tecnológica.

Tomando o ensino de Matemática como foco de uso do Scratch, este pode contribuir com a aprendizagem de conceitos matemáticos de maneira contextualizada. Dentre os diversos conceitos matemáticos que podem ser desenvolvidos com o auxílio do Scratch, destacamos: Figuras planas e espaciais, conceitos algébricos, conceitos aritméticos, ângulos, poliedros, polígonos, equações, área, volume, polinômios, plano cartesiano, dentre outros.

Para que os objetivos educacionais sejam alcançados a mediação do professor é essencial. Nessa perspectiva, o docente precisa criar um espaço educativo em que o aluno se sinta sujeito ativo na construção de sua própria aprendizagem, tornando o ambiente criado na transição entre conceitos matemáticos e a programação do Scratch, amigável, colaborativo e lúdico, em que o erro não é mais visto como um “problema” e sim como uma fase essencial ao desenvolvimento dos aspectos cognitivos.

Referências

LIMA, R. R. da SILVA de; SANTOS, M. B. dos. **Angry Birds no mundo das Funções Afim e Quadrática – Aprendendo Matemática com Scratch**. Encontro Regional de Estudantes de Matemática da Região Sul. Universidade Federal do Pampa. Rio Grande do Sul: Bagé, 2014. Disponível em:

<http://eventos.unipampa.edu.br/eremat/files/2014/12/MC_LIMA_79167578004.pdf>. Acesso em: 12 set. 2015.

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

PRENSKY, M. Digital Natives, Digital Immigrants. **On the Horizon**, v. 9, n. 5, out. 2001.

RESNICK, M., MALONEY, J., MONROY-HERNÁNDEZ, A., RUSK, N., EASTMOND, E., BRENNAN, K., MILLNER, A., ROSENBAUM, E., SILVER, J., SILVERMAN, B., KAFI, Y. Scratch: Programming for all. In: **Communications of the ACM**, nov. 2009, vol. 52, nº11, 2009. Disponível em: <

<http://www.web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf>>. Acesso em: 15 nov. 2016.

VALENTE, J. A. **Computadores e Conhecimento**: repensando a educação. 2. ed. Campinas, SP: UNICAMP/NIED, 1998. 501 p.