

SCRATCH: uma linguagem de construção interativa de competências matemáticas

Tânia Knittel¹Marilene S. S. Garcia²Karen Iglesias³Diego Spitaletti Trujillo⁴Colégio Emilie de Villeneuve, Departamento de Informática Educacional, Brasil ¹UNINTER, Mestrado em Educação e Novas Tecnologias, Brasil ²Unimes, Brasil ³Universidade Aberta de Portugal, Portugal ⁴

Figura 1: Tela inicial do Scratch [1]

RESUMO

Este artigo tem como objetivo apresentar a ferramenta de programação Scratch, destacando o seu potencial para a aprendizagem. A atividade, ainda em desenvolvimento, foi proposta para o 5º ano (faixa etária entre nove e dez anos) de uma escola particular da zona sul de São Paulo, na qual os alunos deveriam criar jogos abrangendo as quatro operações matemáticas: adição, subtração, multiplicação e divisão. Para a execução dessa tarefa, foram trabalhados os principais comandos da linguagem do Scratch, como: movimento, aparência, som, controle, sensores, variáveis e operadores [1], o que resultou, inicialmente, em diferentes jogos a partir da criatividade dos alunos, contribuindo para o desenvolvimento de conceitos, competências e habilidades essenciais, além da criticidade tecnológica. Foi utilizado metodologia exploratória, com abordagem qualitativa dos dados. Os resultados preliminares deste estudo mostram que o Scratch pode ajudar a criar um espaço educativo, criativo e colaborativo, em que o aluno é agente ativo na construção de sua aprendizagem.

Palavras-chave: games digitais, matemática, criatividade, programação.

1 INTRODUÇÃO

O grande desafio da educação continua sendo a aprendizagem do aluno. Considerando os resultados das avaliações institucionais (SAEB – Sistema de Avaliação da Educação Básica, Prova

Brasil¹), que apresentam números preocupantes, constata-se que, apesar do tempo que os estudantes permanecem na escola, sua aprendizagem tem ficado abaixo das expectativas. Estudiosos consideram que as mudanças ocasionadas pelas tecnologias digitais e pela cultura do entretenimento interferem diretamente no interesse do aluno pela forma de aprender [2] [3] [4]. Nesse sentido, discussões sobre como tornar a aula mais significativa e atraente para o estudante, considerando a relação aluno-mundo-sociedade, ganham maior visibilidade e dão origem a metodologias ativas de ensino, de forma que cada vez mais a posição do aluno como protagonista da sua aprendizagem seja levada em conta.

Na esteira dessa discussão, Prensky [3] afirma que as gerações de alunos que estão na escola nasceram na era digital, o que, de forma geral, resulta em jovens com comportamentos mais dinâmicos e com alta fluência digital. Despertar o interesse desses estudantes e engajá-los em processos de aprendizagem demandam adequar o ensino à nova forma de pensar e de agir dessa geração.

Dando suporte a esse pensamento, a teoria da Aprendizagem Baseada em Games (ABG) considera que os jogadores aprendem enquanto jogam [3] [4] [5] [6], posto que os games apresentam situações-problema a serem solucionadas por meio de simulações. Além de sua natureza lúdica e do estímulo à participação do aluno, pode-se gerar experiência individual e interpessoal. Por meio da ABG, é possível provocar no jogador/aluno a reflexão acerca de juízos de valor; desenvolver o pensamento crítico e habilidades para trabalhar em grupo, além de solucionar problemas complexos de forma criativa.

*e-mail: tfknittel@gmail.com

¹ <http://portal.mec.gov.br/prova-brasil/matriz-es-da-prova-brasil-e-do-saeb>

Usando o Scratch em sala de aula, pode-se adquirir tanto experiências de jogabilidade quanto de programação. Para Blikstein [7], ensinar os estudantes a transformar teorias e hipóteses em programas computacionais, leva ao desenvolvimento do pensamento computacional, sendo este uma das habilidades essenciais aos cidadãos do século XXI. Infere-se, nesse sentido, que produção de game se enquadra nessa categoria de produção de programas para computadores. Nesse sentido, o Scratch, tratando-se de um programa que permite o desenvolvimento autônomo e criativo de jogos e animações, a partir de blocos de programação, é então capaz de viabilizar a aproximação do aluno ao pensamento computacional, ratificando Blikstein [7].

Assim, com a atividade, ora descrita, espera-se engajar o aluno para ir além das experiências de usuário, transformando-se em produtor de seus próprios games.

2 TEORIA

O lúdico faz parte da cultura da atividade humana; crianças, jovens e adultos têm necessidade de se envolver em atividades lúdicas. Desde os primórdios, há registros dos seres humanos jogando, brincando, dançando e se divertindo de diversas maneiras. Na educação, a potencialidade do lúdico foi identificada e reconhecida por muitos teóricos, inclusive na Antiguidade. Platão [8] enfatizava que o educador deveria “[...] esforçar-se por dirigir os prazeres e os gostos das crianças na direção que lhes permita alcançar a meta a que se destinarem”. Comênio [9], de igual modo, indicava que os alunos deveriam ser ensinados de maneira que se sentissem atraídos pelo estudo e “que as crianças experimentem nos estudos um prazer não menor que quando passam dias inteiros a brincar com pedrinhas, bolas, e corridas”.

Sobre o pensamento computacional, segundo Valente [10], é possível aplicá-lo na educação básica, podendo ser trabalhado a partir das disciplinas do currículo, a exemplo de escolas espalhadas pelo mundo, que exploram outras atividades como a robótica e as narrativas digitais.

Pensadores como Pestalozzi, Montessori, Fröbel, Rousseau, Decroly, Benjamin e Vigotsky dedicaram suas obras em defesa da aprendizagem por métodos que aproximam a criança ao seu contexto lúdico, por meio do reconhecimento da importância do uso de jogos, brinquedos, brincadeiras e narrativas na infância.

Conforme Vigotsky [10], para a criança que brinca, um brinquedo pode alavancar a criatividade, por meio do imaginário e ainda servir de pivô entre objetos reais e virtuais por perder sua força determinadora. Como exemplo, o autor cita um cabo de vassoura que na mão de uma criança pode se transformar em um cavalo. O brinquedo é então o objeto do jogo e da brincadeira.

Mesmo considerando a existência de uma grande quantidade de definições de jogos, selecionou-se a definição dada por Huizinga [11], entendendo-se ser a que mais se aproxima ao propósito desse estudo. Para o autor, jogo é “[...] uma atividade livre, conscientemente tomada como “não-séria” e exterior à vida habitual, mas ao mesmo tempo capaz de absorver o jogador de maneira intensa e total” [12]. Ainda, conforme o autor, os jogos são atividades “praticadas dentro de limites espaciais e temporais próprios, segundo uma certa ordem e certas regras”.

Para McGonigal [5], os jogos apresentam quatro elementos fundamentais: meta, que seria o objetivo final do jogo, de modo que os jogadores se esforcem para alcançá-lo; regras, que se definem como limitações aos jogadores no processo de busca da meta, impulsionando-os a explorar os espaços do jogo, estimulando a criatividade e o pensamento estratégico; sistema de feedback, funciona como uma espécie de bússola, orientando o jogador a alcançar mais claramente a meta e participação voluntária, a qual delibera que todos os participantes do jogo,

aceitem as regras, objetivos e feedbacks para ele elencados para o jogo.

2.1 Games digitais

Schuytema [13] define game digital ou jogo eletrônico como uma “atividade lúdica formada por ações e decisões que resultam numa condição final. Tais ações e decisões são limitadas por um conjunto de regras e por um universo que, no contexto dos jogos digitais, são regidos por um programa de computador. Para Karl Kapp [14] game é “um sistema em que os jogadores se envolvem em um desafio abstrato, definido por regras, interatividade e feedback, que resulta em uma saída quantificável e frequentemente provoca uma reação emocional”.

Os games digitais são compostos por elementos tais como: sistema; jogadores; desafios; abstração da realidade; regras; interatividade; *feedback*; saída quantificável e reação emocional.

É interessante destacar que, apesar desses elementos aparecerem em grande parte dos jogos digitais, nem sempre é possível encontrar todos eles em um mesmo jogo.

A produção de game por estudantes tem sido cada vez mais incentivada, pois acredita-se que durante sua produção, o estudante tenha de praticar e experimentar diferentes saberes e conceitos já estudados, a fim de testar, comprovar ou refutar suas hipóteses.

2.2 Programação

Kelly [15], ao enfatizar a importância da aquisição da linguagem de programação no empoderamento do indivíduo, traçou uma comparação entre a aquisição da linguagem oral e escrita e a linguagem de programação. Esse estudo sinalizou uma série de razões consistentes para que o ensino de linguagem de programação venha a acontecer na escola.

Programar é fornecer comandos ao computador a partir de uma linguagem sobre o que ele deve realizar, detalhando minimamente a ordem e a forma de cada uma das ações. Exige-se portanto, do aprendiz, as seguintes competências: o desenvolvimento do raciocínio lógico, o uso de conteúdos da matemática, como variáveis, condicionais, entre outros. Além dos fatores mencionados, para que se consiga desenvolver um jogo digital (game), deve-se ter ainda senso de estética, empatia com o jogador, domínio do assunto-tema do game, organização do pensamento, trabalho colaborativo e criatividade.

Políticas públicas de alguns países, dentre eles, os Estados Unidos da América, foram propostas para fornecer suporte ao ensino da programação. Nesse sentido, buscam dar oportunidade para desenvolver nos aprendizes as competências mencionadas acima, por meio do ensino da programação, introduzindo esses conteúdos no currículo da educação básica (K12).

Ainda neste contexto, é possível citar empresas e instituições como o Instituto de Tecnologia de Massachusetts (MIT, 2007), a Microsoft (2014) e a Code.org (2013) que criaram programas para facilitar a aprendizagem da programação, tais como: o SCRATCH, APPINVENTOR e o CODE. São programas que colaboram com a aquisição da lógica de programação, pois apresentam vários dos desafios enfrentados pelos profissionais da computação, com a vantagem da facilidade de seu uso, já que é intuitivo, funcionando a partir de ações como arrastar e soltar blocos. Outro ponto interessante é que possibilitam visualizar o código corrido, ou seja, a escrita da sintaxe do programa, que é a forma como o programador profissional escreve (fora dos blocos).

3 DISCUSSÃO

3.1 Descrição da atividade

A atividade proposta tinha como objetivo promover o primeiro contato dos alunos com a programação desenvolvendo habilidades e competências e aprimorar os conteúdos relacionados com as quatro operações matemáticas.

A partir da apresentação do programa, os alunos foram desafiados no primeiro momento a criar uma animação utilizando as ferramentas básicas oferecidas pelo programa como: movimento, aparência, som, controle, sensores, variáveis e operadores [1].

3.1.1 O Scratch [1]

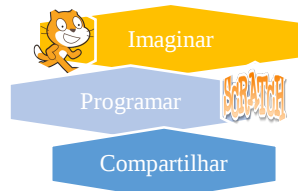


Figura 2: Princípios do Scratch [1]

O *Scratch* é um software de programação que utiliza blocos lógicos voltados para desenvolver projetos interativos como animação, games, histórias com recursos multimídia, de forma motivadora, que possibilita o desenvolvimento de várias habilidades essenciais para a formação do indivíduo no século XXI: a fluência tecnológica digital, raciocínio lógico, pensamento crítico, poder de tomar decisão, inovação, comunicação, análise, capacidade de resolver problemas, interpretação de códigos de programação, uso da razão e da argumentação, função produtiva, criatividade e também conceitos matemáticos específicos, como coordenada, variável e números aleatórios. Além disso, os alunos aprendem esses conceitos em contexto significativo e motivador.

O software pode ser acessado de duas maneiras: *online* (www.scratch.mit.edu) e *offline* por meio de um download (<https://scratch.mit.edu/scratch2download>).

Na figura 2, pode-se conhecer as ferramentas oferecidas pelo software:

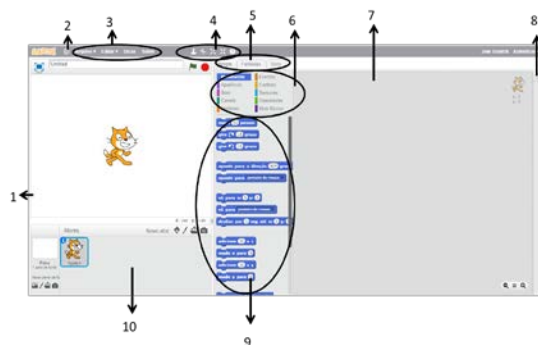


Figura 3: Tela de criação[1]

1. Palco: onde ficam os cenários e os *sprites* (personagens).
2. Idioma: pode-se escolher o idioma com que se deseja trabalhar.
3. Barra de Menus: serve para criar um novo projeto, abrir um já existente, salvar, compartilhar no próprio site e sair.
4. Ferramentas de cursor: serve para apagar, duplicar, diminuir ou aumentar personagens.

5. Paleta de blocos: está dividido em três partes: scripts (descrito no item 6), fantasias (contém o que é necessário para mudar a aparência do personagem), e sons (que podem ser adicionados aos personagens).
6. Paleta de blocos: a aba scripts está dividida por cores: movimento, aparência, sons, caneta, variável, eventos, controle, sensores, operadores.
7. Janela de scripts ou comandos: serve para programar o que os personagens farão a partir dos scripts, que serão construídos com os blocos de comandos.
8. Janela de dicas: serve para ajudar na criação de projetos.
9. Aba de blocos: são os blocos de comandos.
10. Janela de atores: mostra a miniatura de todos os atores do projeto.

3.2 Metodologia

Optou-se por trabalhar a metodologia do *design thinking*, a qual entende os alunos como protagonistas, que buscam soluções de problemas reais junto a professores e seus colegas. Na metodologia proposta buscou-se apresentar aos alunos o software Scratch. Fazendo sua exploração, definiu-se um problema para a sua aplicação, buscando a construção de um game com as noções básicas de matemática, criando uma animação em duplas, com personagens. Posteriormente, foi realizada a testagem e a conceituação da experiência, conforme mostra a figura 4, que apresenta todas essas etapas.

Apresentação do software aos alunos - exploração do software e definição de um problema a ser trabalhado pelos alunos.

Desenvolvimento da solução com elaboração de um game, com animação, simples explorando o palco, personagens e utilizando algumas ferramentas.

Testagem e aprimoramento dos conceitos aprendidos.

Figura 4: Fases do Design Thinking

3.2.1 Sujeitos da pesquisa

A proposta da atividade está sendo desenvolvida com alunos do 5º ano, faixa etária entre nove e dez anos, de uma escola particular da Zona Sul de São Paulo, como projeto prático executado durante as aulas de Matemática.

3.2.2 A atividade

A atividade está em processo e os alunos já criaram a animação. Segundo a fala de um aluno, “é muito legal, ele oferece vários personagens, comandos, eu pretendo fazer um jogo de futebol utilizando as operações. Para programar, você precisa escolher o personagem, o fundo do palco. Para começar a programação você vai em movimentos e escolhe se ele vai andar, girar, depois você tem que ir em controle para escolher, se ele irá fazer a atividade sempre ou repetir algumas vezes. Você pode ir em sensores para escolher, se ele vai bater na borda e voltar”.

Além de contribuir para o desenvolvimento da fluência e criticidade tecnológica, esse software pode contribuir para a aprendizagem de conceitos matemáticos de maneira

contextualizada. Dentre os diversos conceitos 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.

Na próxima etapa, os alunos irão usar a ferramenta operadores, além das que elas já trabalharam para criação dos games:

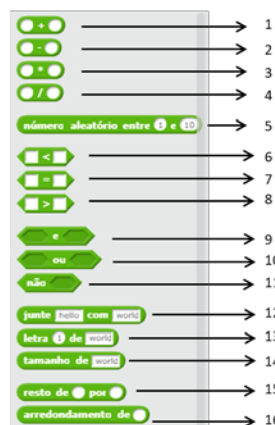


Figura 5: Tela de Operadores [1]

1. Soma dois números
2. Subtrai o segundo número do primeiro
3. Multiplica os dois números
4. Divide o primeiro número pelo Segundo
5. Escolhe aleatoriamente um número dentro do intervalo específico
6. "Verdadeiro" se o primeiro valor for inferior ao Segundo
7. "Verdadeiro" se os dois valores forem iguais
8. "Verdadeiro" se o primeiro valor for superior ao Segundo
9. "Verdadeiro" se ambas as condições forem simultaneamente verdadeiras
10. "Verdadeiro" se ambas ou apenas uma das condições forem verdadeiras
11. "Verdadeiro" se a condição for falsa. "Falso" se a condição for verdadeira
12. Combina cadeias de caracteres
13. Apresenta o caractere na posição especificada da cadeia de caracteres
14. Informa o número de caracteres em uma cadeia de caracteres
15. Tem o valor do número inteiro mais próximo de um número dado
16. Tem o valor do número inteiro mais próximo de um número dado

Na etapa seguinte os alunos desenharam os jogos e criaram os algoritmos para depois poderem programar utilizando o software.

4 CONCLUSÃO

Neste artigo, apresentamos um estudo em processo sobre o uso do Scratch durante as aulas de Matemática, no qual foi possível avaliar o interesse dos alunos em aprender a programar, despertando o primeiro contato com a lógica de algoritmos, fazendo com que não sejam apenas usuários da tecnologia, mas produtores, percebendo que os computadores dependem de instruções de pessoas para a realização das tarefas.

Nas próximas etapas, espera-se ainda mais autonomia por parte dos educandos, nas quais os processos de criação estarão relacionados com o desenvolvimento de habilidades como

pensamento crítico, poder de tomada de decisão, inovação, comunicação, análise, capacidade de resolver problemas, interpretação de códigos de programação, uso da razão e da argumentação, habilidade de escutar, função produtiva e criatividade.

Esperamos também com essa atividade que o erro não seja visto como um problema e sim como uma etapa da aprendizagem, buscando aperfeiçoamento de raciocínios e práticas, a partir da mediação do professor e criando-se um espaço em que o aluno seja o sujeito ativo na construção do seu conhecimento.

REFERÊNCIAS

- [1] <https://scratch.mit.edu/> Acesso em: 02 de ago. de 2017.
- [2] A. G. Sakata. *Desmotivação dos alunos: como a gamificação pode ajudar no engajamento dos estudantes*. Disponível em: <<http://www.playdea.com.br/beneficios>>. Acesso em: 04 de ago. de 2017.
- [3] M. Prensky. *Aprendizagem baseada em jogos digitais*. Tradução Eric Yamagute; revisão técnica de Romero Tori e Denio Di Lascio. São Paulo: Editora Senac São Paulo, 2012.
- [4] J. Mattar. *Games em educação: como os nativos digitais aprendem*. 1ª. Reimpressão. São Paulo: Pearson Prentice Hall, 2012.
- [5] J. McGonigal,. *Jogando por um mundo melhor*. TED: Ideas Worth Spreading, 2011. Disponível em: https://www.ted.com/talks/jane_mcgonigal_gaming_can_make_a_better_world?language=pt-br. Acesso em: 28 de maio de 2017.
- [6] J. P. Gee: *Vídeo Games, Aprendizado, e Alfabetização*. Educação Livre. Disponível em: https://www.youtube.com/watch?v=__LuGlGgkCE. Acesso em: 17 de jun. de 2017.
- [7] P. Blikstein. *O pensamento computacional e a reinvenção do computador na educação*. Disponível em <http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html> Acesso em: 04 de ago. de 2017. 2008.
- [8] M. J. M. Silveira. *O ensino e o lúdico*. Santa Maria Multiprees, 1988.
- [9] J. A. Comênio. *Didática Magna*. 3 ed. Lisboa: Fundação Calouste Gulbenkian, 1957.
- [10] J. A. Valente. Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. *Revista e-Curriculum*, v. 14, n. 3, 2016.
- [11] L. S. Vigotsky. *A formação social da mente*. 2 ed. São Paulo: Martins Fontes, 1988.
- [12] M. L. Fardo. *A gamificação como estratégia pedagógica: estudo de elementos dos games aplicados em processos de ensino e aprendizagem*. Universidade Caxias do Sul, 2013. Disponível em: <https://repositorio.uces.br/xmlui/bitstream/handle/11338/457/Dissertacao%20Marcelo%20Luis%20Fardo.pdf;jsessionid=9C7432DADB53972DA1AD5D3C7D20E53C?sequence=1>. Acesso em: 05 de ago. de 2017.
- [13] P. Schuytema. *Design de games: uma abordagem prática*. São Paulo: Cengage Learning, 2008. 447 p.
- [14] K. M. Kapp. *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons, 2012.
- [15] K. Kelly. *Programa ou seja programado*. SuperGeeks. 2013. Disponível em: https://www.youtube.com/watch?v=SF_B77jXrrk. Acesso em: 04 de ago. de 2017.