

## LINGUAGEM VISUAL DE PROGRAMAÇÃO: OPÇÃO DE MÍDIA E TECNOLOGIA INTERESSANTE À EDUCAÇÃO MATEMÁTICA

*William Vieira Gonçalves*

*Universidade do Estado do Mato Grosso – Câmpus de Barra do Bugres  
williamvieira@unemat.br*

*Wilson Massashiro Yonezawa*

*Universidade Estadual Paulista “Júlio de Mesquita Filho” – Câmpus de Bauru  
yonezawa@fc.unesp.br*

*Helio Fernando Gomes Maziviero*

*Universidade Estadual Paulista “Júlio de Mesquita Filho” – Câmpus de Bauru  
heliomaziviero@gmail.com*

*Armando Paulo da Silva*

*Universidade Tecnológica Federal do Paraná – Câmpus Cornélio Procópio  
armando@utfpr.edu.br*

*Rudolph dos Santos Gomes Pereira*

*Universidade Estadual do Norte do Paraná – Câmpus de Cornélio Procópio  
rudolph.matematica@gmail.com*

### **Resumo:**

Nesse material tem o intuito de destacar alguns softwares que possibilitam o desenvolvimento de mídias digitais como atividades didáticas que entrelaçam aprendizagem de conceitos matemáticos e as perspectivas construcionistas inicialmente desenvolvidas por Seymour Papert. Uma linguagem visual de programação se constitui de blocos de comandos e com recursos para clicar e para arrastar, concatenando-os para formar algoritmos. Deste modo, o usuário pode construir diferentes aplicações que usam texto, imagem, som e movimento e, sem precisar lidar com sintaxes ou códigos de linguagens extremamente complexas. O objetivo é apoiar a divulgação desses recursos em espaços de educação matemática e reconhecer sua viabilidade pedagógica com uma breve exposição das suas ferramentas e das comunidades colaborativas voltadas à sua discussão para a educação. Concomitantemente buscamos corroborar a relevância dessa proposição ressaltando seus potenciais pedagógicos.

**Palavras-chave:** Linguagem visual de programação; Construcionismo; Mídias e Tecnologias na Educação Matemática.

### **1. Introdução**

Para fazer os programas de computador há a necessidade de domínio de conhecimentos e linguagens específicos de informática e esse é um fator que dificulta sua exploração por parte de professores e estudantes para compor representações, animações,

jogos ou outros objetos digitais. O grupo de pesquisa *Ensino de Ciências e Tecnologia Educacional*, constituído por professores e estudantes da Universidade Estadual Paulista "Júlio de Mesquita Filho" (UNESP) no Câmpus de Bauru, dentre outras questões, estuda e se interessa por esta discussão. Objetivamente, a disciplina *Jogos Digitais no Ensino de Ciências*, ofertada junto ao Programa de Pós Graduação *Educação para a Ciência* da UNESP em Bauru, a qual reúne estudantes oriundos de licenciaturas em Biologia, Física, Matemática, Pedagogia e Química; procura promover o desenvolvimento de jogos digitais através do uso de linguagem visual de programação e tem mobilizado discussões acerca das dificuldades em lidar com programação por parte dos professores e dos estudantes.

Nesse momento de resgate e de reflexão das perspectivas da Educação Matemática brasileira acredita-se ser oportuno resgatar a ideia da aprendizagem *Construcionista* e as intenções da proposta do *Logo*. Nossa intenção é apontar a perspectiva de aprendizagem que o recurso aqui exposto se associa e justificar seu uso a partir de algumas dificuldades e limites encontrados em estudos nesse ambiente de programação. A linguagem de programação *Logo* foi criada por Wally Feurzeid e Seymour Papert no *Massachusetts Institute of Technology* (MIT) voltada para a Educação, mas foi Papert que defendeu e utilizou computadores e essa linguagem como um instrumento de auxílio no processo de aprendizagem. No Brasil, este aplicativo inicialmente difundido pelo Professor *José Armando Valente* e está presente em experiências no ensino de matemática, em especial, no ensino de geometria com apoio de informática educacional. Para uma revisão geral no assunto abordado recomenda-se a leitura de Miskulin (1999). Vale ressaltar que:

é possível construir computadores de modo que aprender a comunicar-se com eles seja um processo natural, mais parecido com o processo de aprender francês morando na França do que com o método antinatural do ensino americano de línguas estrangeiras. [...] aprender a comunicar-se com um computador pode mudar a maneira como outras aprendizagens acontecem. O computador pode ser um interlocutor-de-matemática, ou um interlocutor-de-línguas (PAPET, 1985, p. 18 apud MISKULIN, 1999, p. 230)

Tendo os ambientes de programação como um interessante modo de implementar a comunicação, consideramos o *ciclo descrição - execução - reflexão – depuração*, proposto e intensamente discutido em Valente (2005); este conceito parece articulável com os processos de aquisição de conhecimento matemático e as *habilidades de aprendizagem* percebidas no uso de linguagem visual de programação. Associando os recursos de interação e informação que a internet proporciona, apoia-se o seguinte posicionamento de educadores matemáticos:

Hoje, há os que defendem a posição de que os conceitos e generalizações de uma programação não devem ser ensinados a todos. No entanto, a experiência em programação é útil ao desenvolvimento do raciocínio lógico, pois o ato de programar exige sucessivas antecipações e projeções sobre os resultados das várias partes de um objeto simbólico (MOTTA; SILVEIRA, 2010)

Depois de apontar as perspectivas pedagógicas que estimulam esta proposta, elenca-se as dificuldades e as limitações que motivam a busca por outras formas de se utilizar a linguagem de programação para as atividades escolares de matemática.

Apesar de todas as potencialidades apresentadas pelo SuperLogo, algumas dificuldades técnicas e pedagógicas do *software* são destacadas. Sobre as dificuldades técnicas, destacam-se: o programa não tem *plugin* que permita a publicação de projetos em páginas da *internet*. Dessa forma, atividades realizadas não podem ser apresentadas em formato *html*, o que dificulta a divulgação dos projetos desenvolvidos pelas escolas. Os projetos realizados somente são armazenados na forma de procedimentos, ou seja, os comandos necessários, não a figura obtida, e toda vez que esses arquivos são utilizados faz-se necessário utilizar procedimentos. Só é possível rodar o SuperLogo em *Linux* se estiver instalado um emulador que faz com o programa não fique lento. O SuperLogo não possui uma interface e interatividade gráfica que agrada aos alunos. Em relação às dificuldades pedagógicas, destacam-se: atualmente, alguns professores, principalmente das séries iniciais da Educação Básica, sentem dificuldade em trabalhar com o SuperLogo, pois a linguagem possui implícito um conhecimento matemático e um raciocínio lógico prévio por parte do educador. Alguns professores, ao trabalharem com o SuperLogo, afirmam que os alunos sentem-se cansados em interagir com o *software* e solicitam outro tipo de atividade. Esse cansaço deve-se provavelmente ao fato de se proporem atividades em que os alunos somente aplicam os comandos do programa, não interagindo entre si e nem desenvolvendo suas habilidades intelectuais (MOTTA, SILVEIRA, 2010)

Cabe ressaltar que o ambiente *Logo* e as versões derivadas, possuem uma interface escrita para entrada dos comandos, sendo esta, uma das reclamações de seus usuários. Muitos afirmam ser difícil memorizar e interpretar a sintaxe da linguagem, exigindo um tempo relativamente longo para criar segurança e intimidade com o software.

Nas seções subsequentes, indica-se algumas referências sobre as linguagens visuais de programação e suas potencialidades pedagógicas. E ainda, sintetiza-se informações para reconhecer e acessar esses ambientes.

## **2. Potencialidades pedagógicas percebidas em ambientes de programação visual**

No relatório *The Partnership for 21st Century Skills* (2002), são apontadas as *21st Century Learning Skills* (Habilidades em Aprendizagem para o século 21, nossa tradução), e em resumo, é importante estimular *habilidades de aprendizagem em Informação e Comunicação, Resolução de Problemas junto a Avaliação e Auto Avaliação*. Os trabalhos de Harvey, Mönig (2010), Maloney, et al. (2010) e Brennan, Resnick (2012) trazem

relevantes e atuais fundamentações e estudos de caso com resultados e discussões sobre os usos desta tecnologia em contextos escolares. Em geral as atividades vêm confirmando que as *habilidades de aprendizagem* se consolidam ao usar os ambientes propostos e demonstram engajamento e interação por parte dos professores e dos estudantes. Outros dois aspectos têm-se mostrado producentes nestes ambientes ao construir diferentes objetos de mídia, os usuários estariam explorando a *aprendizagem por design*, o que equivale a *aprendizagem por projetos*. Essa opção permite formular propostas específicas de objetos para os mais intrincados interesses, o que se aproxima das orientações do Programa Nacional de Tecnologia Educacional de valorizar e encontrar *ferramentas de autoria* que permitem que professores e alunos se tornem autores do tema que está sendo estudado.

### 3. Sugestões de aplicativos com linguagem visual de programação

O *Scratch* foi desenvolvido pelo Lifelong Kindergarten Group at the MIT Media Lab, originalmente voltado para crianças de 6 a 8 anos de idade, porém os trabalhos citados apontam que pode ser usado em todos os níveis de ensino. Todos os recursos de tutoriais, vídeos, exemplos, comunidades virtuais e versões do aplicativo podem ser acessados a partir de <http://scratch.mit.edu/>. No Brasil este ambiente já possui comunidades interessadas e podem ser obtidas informações em <http://www.scratchbrasil.com.br/>.

O *Byob* deriva do *Scratch* e seu desenvolvimento acrescenta possibilidades de programação mais avançadas, além de manter as funcionalidades do aplicativo que o inspirou. Criado por Jens Mönig e Brian Harvey pode-se obter suas versões e materiais em <http://byob.eecs.berkeley.edu/>. Os dois ambientes anteriores demandam instalação nos computadores, no entanto, o ambiente *Snap!* é executado diretamente nos *browsers* (navegadores da *internet*), mantendo todas as funcionalidades das propostas anteriores. Oferecido pela Universidade da Califórnia em Berkeley, foi desenvolvido por Jens Mönig/*MioSoft Corporation*, com apoio de Brian Harvey e estudantes da Universidade de Berkeley. Acesse <http://byob.eecs.berkeley.edu/> para obter materiais e em <http://snap.berkeley.edu/snapsource/snap.html> pode-se acessar diretamente o ambiente.

### 4. Considerações Finais

As sugestões de ambientes apresentadas funcionam em qualquer sistema operacional e possuem suporte em português e disponibilizam espaços virtuais para

publicação dos objetos de mídia criados. Apesar de se prestarem a vários objetivos pedagógicos, focam-se em estimular a aprendizagem de ideias matemáticas e computacionais, contextualização e motivação intrínsecas a criação de diferentes interpretações de problemas, histórias, jogos e arte; sendo essencialmente interessantes os espaços de divulgação da produção e avaliações colaborativas. Nosso grupo de pesquisa está gerando projetos de extensão em parcerias com escolas da Educação Básica a fim de aproveitar sua eficácia e usabilidade no contexto brasileiro.

## 5. Referências

BRENNAN, K.; RESNICK, M. New frameworks for studying and assessing the development of computational thinking. **Proceedings of the 2012 annual meeting of Educational Research Association**, 2012. Disponível em:

<[http://web.media.mit.edu/~kbrennan/files/Brennan\\_Resnick\\_AERA2012\\_CT.pdf](http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf)>.

Acesso em: 01 jan. 2013.

HARVEY, B.; MÖNIG, J. Bringing “No ceiling” to scratch: can one language serve kids and computer scientists. **Proc. Constructionism**, 2010. Disponível em:

<<http://byob.eecs.berkeley.edu/BYOB.pdf>>. Acesso em: 20 marc. 2012.

MALONEY, J.; RESNICK, M.; RUSK, N. The scratch programming language and environment. **ACM Transactions on Computer Education**, v. 10, n. 4, p. 15, 2010.

MISKULIN, R. G. S. **Concepções teórico-metodológicas sobre a introdução e a utilização de computadores no processo ensino/aprendizagem da geometria**, 1999.

Universidade Estadual de Campinas. Disponível em:

<<http://www.cempem.fae.unicamp.br/lapemmec/coordenacao/tese.pdf>>. Acesso em: 01 jan. 2013.

MOTTA, M. S.; SILVEIRA, I. F. Contribuições do Superlogo ao ensino de geometria. **INFORMÁTICA NA EDUCAÇÃO: teoria & prática**, v. 13, n. 1, p. 115-127, 2010.

PAPERT, S. **Logo: Computadores e Educação**. 1985. Trad. J. A. Valente, B. Bitelman, A. V. Ripper. São Paulo: Brasiliense.

THE PARTNERSHIP FOR 21ST CENTURY SKILLS. **Learning for the 21st century**. 2002. Disponível em: <[http://www.p21.org/storage/documents/P21\\_Report.pdf](http://www.p21.org/storage/documents/P21_Report.pdf)>. Acesso em: 30 marc. 2012.

VALENTE, J. A. **A espiral da espiral de aprendizagem: o processo de compreensão do papel das tecnologias de informação e comunicação na educação**, 2005. Universidade Estadual de Campinas. Disponível em: <<http://pan.nied.unicamp.br/~lia>>. Acesso em: 12 fev. 2011.