

Escolha do Ambiente de Programação de Computadores *Scratch* para Investigação de Aprendizagens Matemáticas

**Valéria Espíndola Lessa, Gabriel Paludo Licks, Guilherme Zanatta Tocchetto,
Franciele Meinerz Forigo**

Instituto de Ciências Exatas e Geociências – Universidade de Passo Fundo
Passo Fundo – RS – Brasil

lessavaleria@gmail.com, {138119, 141979}@upf.br,
francieleforigo@gmail.com

***Resumo.** Este artigo apresenta brevemente um estudo sobre ambientes de programação para crianças, que faz parte das etapas de uma tese de doutorado realizada no GEPID da UPF, a fim de escolher um ambiente para a realização de investigação sobre aprendizagens matemáticas. Primeiramente realizamos uma busca na internet dos principais ambientes de programação utilizados para fins educacionais, seguido de uma seleção dos quatro mais adequados à nossos estudos. Posteriormente, realizamos testes nestes quatro softwares por meio da realização de programações. Como resultados parciais, obtivemos o ambiente de programação Scratch como o mais indicado para dar continuidade à investigação da tese.*

1. Introdução

Este artigo resumido apresenta um estudo sobre ambientes de programação para a educação, na qual faz parte de uma pesquisa de doutorado realizada no Grupo de Estudos e Pesquisas em Inclusão Digital (GEPID) da Universidade de Passo Fundo (UPF). A referida pesquisa objetiva investigar as potencialidades da programação de computadores na aprendizagem de conceitos matemáticos e, para tanto, necessita estabelecer o software mais adequado para este fim.

Dessa forma, apresentaremos uma das primeiras etapas da pesquisa de doutorado, na qual alguns ambientes de programação foram encontrados e analisados quanto a certos critérios que consideramos importante para um software de programação voltado a aprendizagens de crianças em escolarização. A seguir mostraremos brevemente o referencial teórico da pesquisa de doutorado que embasa também este levantamento, a metodologia utilizada neste estudo preliminar e os resultados parciais.

2. Breve referencial teórico

Muitas ideias matemáticas como espaço, movimento e padrões repetitivos podem chegar às crianças de maneira mais natural quando construídas numa simulação e a programação de computadores em softwares específicos permite a realização de inúmeras simulações. Com isso, a programação poderia concretizar conhecimentos abstratos e formais uma vez que permite dar “[...] formas concretas a áreas do conhecimento que pareciam ser anteriormente inatingíveis [...]” (Papert, 1985, p.39).

Para Papert (1985), o ensino por meio da transferência de informações ou de “mostrar” como se faz, conforme acontece na escola tradicional, nem sempre surte o efeito desejado de aprendizagem, ele sugere que o professor proporcione para que “

aquilo a ser aprendido faça sentido” (p.87) ao estudante. A experiência de vivenciar a matemática é importante e a programação de computadores permite isso.

A atividade em ambientes de programação também possibilita que a criança, ao testar suas hipóteses sobre a realização de um procedimento na forma de linguagem de programação, confronte imediatamente os resultados a partir do *feedback*. Este retorno imediato, permite a reflexão sobre a ação e, se necessário, a reformulação das hipóteses iniciais para testá-las novamente. O erro não é entendido com algo negativo, mas como processo necessário na busca da solução para o problema. Este é aproximadamente o processo em espiral da Teoria Construcionista de Papert, na qual podemos perceber que cada descrição do pensamento precisa, necessariamente, ser explicitada na forma de linguagem de programação ou de regras lógicas específicas dependendo do software utilizado. E esta descrição está "recheada" de conhecimentos, principalmente de matemática.

Dessa forma, buscamos investigar quais os ambientes de programação voltados à educação estão disponíveis na internet para escolher um que possa servir como ferramenta de investigação na tese de doutorado em desenvolvimento.

3. Metodologia e resultados parciais

O levantamento dos diferentes ambientes de programação para a educação foi realizado a partir de informações que o grupo já possuía sobre os softwares e por meio de pesquisas na internet que retornaram 16 ambientes. Organizamos estes achados numa planilha¹ segundo alguns critérios que julgamos importantes para a escolha de um ambiente. Posteriormente, com o intuito de melhorar o processo de seleção e de valorizar o que é realmente relevante num software, escolhemos quatro destes critérios e pontuamos, conforme pode ser observado na Tabela 1.

Tabela 1. Critérios e pontuação

É gratuito?	Possui idioma Português?	Roda em Linux e Windows?	Faixa etária a partir dos 11?
0 - não 1 - sim	0 - não 1 - sim	0 - nenhum 1 - em um deles 2 - nos dois	0 - não 1 - sim

Julgamos que ser gratuito e poder ser instalado em sistema operacional Linux ou Windows, contribui para que os estudantes que participarem da pesquisa possam baixar os ambientes em seus computadores pessoais, considerando que os sistemas mencionados são os mais comuns. Tendo a opção pelo idioma português, permite que o estudante entenda com mais clareza os comandos do ambiente e não divida esforços entre a tradução e a programação. E, a escolha pela faixa etária a partir dos 11 anos de idade vai ao encontro da opção metodológica da pesquisa de doutorado, que pretende investigar a aprendizagem de estudantes que, a princípio, já tivessem contato com um maior número de conceitos formais de matemática.

Após realizada a pontuação conforme a Tabela 1, quatro ambientes obtiveram a pontuação máxima 5: Alice; Light Bot; Scratch; e Super Logo. A partir disso, tentamos realizar programações simples como a construção de figuras geométrica nos ambientes,

¹ Link para planilha on-line com os ambientes estudados: <<https://goo.gl/65eKhl>>.

a fim de analisarmos o nível de detalhamento da representação do código, o que este nível de detalhamento apresenta de informações sobre os conhecimentos de matemática - visto que alguns ambientes utilizam abstrações que podem ser importantes nessa avaliação, os recursos que este ambiente oferece e as dificuldades que pode apresentar aos seus usuários. De início, pretendíamos construir figuras geométricas planas regulares nos quatro ambientes para poder comparar os códigos fontes. No entanto, devido às especificidades de cada ambiente, não foi possível.

O ambiente Lightbot possui uma versão gratuita, porém é apenas de demonstração. A versão de download para Windows é paga. Além disso, ele é um jogo, ou seja, para programar temos que seguir fases e determinadas regras que o ambiente ensina ao usuário. Por exemplo, em cada fase há um caminho que o personagem deve trilhar. O ambiente oferece passos, pulos, giros de 90°, mas o número de utilizações de cada comando é limitado. Os códigos fonte são representados por desenhos, ou seja, um passo para frente é representado por uma flecha e um pulo é representado por uma mola. Consideramos um ambiente de programação inicial que não permite avançar na programação nem usar da criatividade para elaborar as próprias regras e o próprio cenário, ou seja, não conseguimos construir figuras geométricas. A matemática por trás do código fonte fica restrita aos conceitos relacionados à movimento como: medidas, ângulos e direções.

O Alice² é um software de animação 3D cuja versão 3.2 que analisamos não traduz os blocos de códigos de programação para o português, sendo um grande limitador para nossa investigação. Apesar disso, o ambiente oferece muitos recursos interessantes e os comandos estão em blocos, ou seja, não precisam ser digitados pelo usuário, basta serem escolhidos dentre uma série de outros blocos existentes e encaixados segundo uma sequência lógica. Este ambiente permite a criação de mundos digitais, o que, conseqüentemente, permite a construção de muitas figuras geométricas e exige conhecimentos de matemática nos mais diversos níveis.

O SuperLogo 3.0³, é um dos softwares criados para ser trabalhado com a linguagem LOGO e sua programação se dá por comandos digitados, na qual uma tartaruga gráfica se movimenta na tela desenhando linhas retas, necessitando de conhecimento prévio dos comandos principais do ambiente. É um ambiente de programação que gera elementos da geometria. Por exemplo, para desenhar um quadrado é necessário escrever: “parafrente 50 paradireita 90 parafrente 50 paradireita 90 parafrente 50 paradireita 90 parafrente 50”; ou “repita 4 [parafrente 50 paradireita 90]. Também é possível criar comandos que possuam uma sequência de outros comandos (subrotinas), dessa forma é possível que o ambiente desenhe uma figura completa, por exemplo, a partir de apenas um comando simples. Este ambiente foi bastante investigado por outras pesquisas há alguns anos e, se os estudantes aprenderem seus comandos e suas possibilidades, muita coisa interessante pode ser construída com o SuperLogo.

² Criado pela Carnegie Mellon University. < <http://www.alice.org> >

³ Software de programação desenvolvido no Massachusetts Institute of Technology (MIT), na qual um de seus criadores foi Seymour Papert. Mais informações em: <<http://projetologo.webs.com/slogo.html>> e http://www.nied.unicamp.br/oea/mat/LOGO_IMPLICACOES_bette_nied.pdf

O ambiente Scratch⁴ pode ser apresentado de forma que sua linguagem é vista como uma evolução da linguagem LOGO, posto que nele os comandos não são digitados, mas selecionados a partir de uma série de scripts existentes e encaixados um no outro, como blocos, se aproximando da ideia do jogo LEGO. Dentro de cada tipo de script há diferentes comandos. Os scripts são: movimento; aparência; som; caneta; variáveis; eventos; controle; sensores; operadores; e permite a criação de outros blocos. Aqui não há tartaruga, mas uma raposa ou qualquer outro personagem que se queira inserir na tela. Trabalhando no software, podemos observar que a lógica de construção é muito parecida com a do SuperLogo, porém sem a escrita dos comandos. Ele permite ir além do desenho de figuras. É possível inserir elementos na tela e criar jogos e simulações.

A partir destas análises, consideramos que o Lightbot e o Alice, não atendem nossas necessidades para a investigação tendo em vista que o primeiro possui limitação na criação e na elaboração de estratégias de programação e o segundo, apesar das grandes possibilidades de criação de jogos e simulações em terceira dimensão, os comandos somente na língua inglesa são um limitador para os estudantes e não atende aos critérios iniciais que estabelecemos. O SuperLogo, apesar de seu histórico em atividades de sala de aula, de envolver muita matemática em suas construções e de podermos enxergar matemática em seu código fonte, é limitado em seus recursos.

Optamos, dessa forma, em utilizar o ambiente de programação Scratch para nossa investigação de doutorado por possibilitar tudo o que os demais não possibilitaram: (i) totalmente traduzido para o português; (ii) versão de download grátis; (iii) possibilidade de criação de personagens e cenários; (iv) possibilidade de criação de elementos simples como um quadrado à simulações e jogos; (v) comandos já “prontos” bastando serem encaixados. Além disso, acreditamos que, por trás da representação de código fonte gerada pelas diferentes programações em Scratch, podemos encontrar muitas representações matemáticas. E essa busca será o próximo passo da investigação de doutorado.

Referências

- PAPERT, S. (1985) "Logo: computadores e educação".Tradução de José A. Valente, Beatriz Bitelman e Afira V. Ripper. São Paulo: Editora Brasiliense.
- PAPERT, S. (2008) "A máquina das crianças: repensando a escola na era da informática". Ed.rev. Porto Alegre: Artmed.

⁴ Também foi elaborado no MIT e é um projeto do grupo Lifelong Kindergarten no Media Lab. <<https://scratch.mit.edu/>>