

LÓGICA DE PROGRAMAÇÃO E SUA INFLUÊNCIA NO ENSINO

Gabriel Silva Delgado¹

Leonam Mendonça Pereira²

Paula Zitko Alves Ramos³

Roseane Salvio⁴

Resumo

Este trabalho visa abordar o uso da lógica de programação em sala de aula, baseado nas experiências de dois bolsistas de um projeto de extensão de uma faculdade federal do estado de São Paulo, aplicado a uma escola municipal de Campos do Jordão. Para isso, foi desenvolvida uma metodologia de ensino para o ensino fundamental II, com alunos do 8º e 9º ano. A ação se desenvolveu no laboratório computacional da própria escola municipal, onde foram ensinados conceitos lógicos de introdução a lógica de computadores, utilizando o software livre *Scratch*.

Palavras-chaves: Educação. Lógica de Programação. *Scratch*.

Abstract

This paper aims to address the use of *Scratch* in the classroom based on the experiences of two scholarship from an extension project of a federal college of São Paulo applied to a municipal school of Campos do Jordão. It was developed a teaching methodology for elementary education, with students of the 8th and 9th year, the action was developed in the computational laboratory of the municipal school, where was taught logical concepts of introduction to computer logic using the free software *Scratch*.

Keywords: Education. Programming logic. *Scratch*.

¹ Instituto Federal de São Paulo – Campus Campos do Jordão, Graduando em Licenciatura em Matemática, E-mail: gsd2005online@hotmail.com

² Instituto Federal de São Paulo – Campus Campos do Jordão, Graduando em Tecnologia de Análise e Desenvolvimento de Sistemas, E-mail: leonammp.15@gmail.com

³ Instituto Federal de São Paulo – Campus Campos do Jordão, Mestrado em Engenharia Elétrica, Professora, E-mail: paula.zitko@ifsp.edu.br

⁴ Instituto Federal de São Paulo – Campus Campos do Jordão, Especialização em Gestão em TI e Banco de Dados, Professora, E-mail: roseane@ifsp.edu.br

Introdução

O projeto para ensino de lógica de programação na educação básica visa demonstrar a influência positiva de tal aplicação, de forma que os alunos desenvolvam melhor o poder cognitivo para as demais disciplinas exigidas no ensino básico. Segundo Pereira (2013), a lógica de programação deveria andar junto com outras disciplinas do ensino básico, tais como biologia, química e física. Neste contexto, conforme Kafai e Burke (2013), o ensino de programação para crianças poderia desenvolver o pensamento computacional e os passos lógicos para a resolução automatizada de problemas.

O projeto de extensão que possibilitou a execução desta ação, inicialmente, tinha como objetivo analisar duas turmas. Em uma turma, aplicaríamos uma metodologia de ensino de lógica de programação com computadores, e em outra, não. Porém, com a falta de recursos disponíveis, foi possível executar o projeto em apenas uma turma, sendo decidido ensinar lógica de programação com o software livre *Scratch*.

1. Preliminares

Nesta seção, serão elencadas as principais características e tarefas acerca do projeto, por meio da contextualização da programação de computadores como parte do ensino de matemática e da discussão a respeito do *software* livre *Scratch* e sua introdução em sala de aula.

1.1. Contextualização

Segundo Fini e Miceli (2012), a matemática, no Currículo do Estado de São Paulo (CESP), é vista como algo inerentemente necessário para todo ser humano, de modo que, frequentemente, mesmo que de forma tácita, adultos precisam da matemática em quase todas suas ações rotineiras; e, por conta disso, é papel da escola preparar os alunos para enfrentar a vida fora dela. O CESP trabalha, em seu desenvolvimento, com a contextualização do mercado de trabalho como estratégia para a exploração de cada competência matemática, além de identificar e explorar a ideia fundamental de cada tema, tendo em foco sempre a transformação da informação em conhecimento.

É na área da matemática que a programação de computadores aparece no CESP, em que é vista como uma ferramenta capaz de desenvolver a capacidade de arguição e tomada de decisão, por meio do trabalho com tratamento da informação, em busca de uma visão crítica do tema disciplinar em questão.

Na Base Nacional Curricular Comum Curricular (BNCC), documento que busca regularizar o currículo escolar, o uso de tecnologias é bastante mencionado, sendo citado em uma de suas 10 competências gerais:

Utilizar diferentes linguagens – verbal (oral ou visual-motora, como Libras, e escrita), corporal, visual, sonora e digital –, bem como conhecimentos das linguagens artística, matemática e científica, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao entendimento mútuo. (BRASIL, 2016).

É interessante observar que, para a BNCC, o uso do recurso da tecnologia digital estende-se para praticamente todos os conteúdos de todas disciplinas. Portanto, é importante colocar isso em prática, visto que o uso de novas tecnologias revolucionou o cotidiano do ser humano.

De acordo com Simões e Gouveia (2008),

Nos últimos 20 anos, as tecnologias de informação foram responsáveis por uma profunda reorganização do modo como as pessoas vivem, comunicam-se e aprendem. Estas alterações têm permitido o aparecimento de novas práticas e hábitos associados com as novas gerações [...] (SIMÕES; GOUVEIA, 2008, p. 21).

Deste modo, é imprescindível que o professor tenha domínio e prática de usos tecnológicos para o ensino, no entanto, ainda não é obsoleto o uso de tarefas não computacionais. Assim, não devemos descartar os conhecimentos provectos, mas complementá-los com novos.

As tecnologias que surgem se mostram como uma diferente abordagem das mais variadas disciplinas, que pode ser empregada com alunos de qualquer idade. Tais tecnologias são enriquecedoras para a sala de aula, visto que podem introduzir assuntos como a história da matemática, jogos educativos, projetos interdisciplinares, além de propiciar uma ótima alternativa de fonte de pesquisa para uma aula invertida, dentre outros. Porém, é necessário que o professor saiba optar pela tecnologia que mais se adeque à aula e tenha cuidado com seus conteúdos.

Em nosso projeto, por um período de 3 meses, foi ofertado, em uma escola municipal de Campos do Jordão, um curso optativo de lógica de programação. O público alvo foram alunos dos anos finais do ensino fundamental II regular e os professores a ministrar o curso

eram graduandos bolsistas de um projeto de extensão de uma faculdade local. Os objetivos destas aulas ministradas consistiram não só em apresentar a disciplina de lógica de programação para os alunos, mas, também, em potencializar a formação do estudante e o seu desenvolvimento na escola, oferecendo-lhe a oportunidade de exercitar seu raciocínio lógico desde cedo.

Tentar contextualizar problemas acadêmicos em sala de aula pode ser difícil, então, usar a imaginação pode ser uma boa solução. Para chegar a esse pressuposto, tivemos como base didática a teoria de Vygotsky (1986), que aborda a importância do desenvolvimento da imaginação para a progressão do indivíduo. É nesse âmbito que o software livre *Scratch* surge como uma ferramenta para alcançar nossos objetivos.

1.2. Software livre *Scratch*

O *Scratch* é um ambiente de programação que permite escrever e compilar códigos de programação em linguagem *Scratch*. Mitch Resnick (2012), o criador do *Scratch*, destaca a necessidade de aprendermos programação, considerando as novas gerações como nativos tecnológicos, pelo fato de estarem constantemente utilizando novas tecnologias digitais. O autor faz uma analogia, comparando tecnologias digitais com um idioma. Para ele, é como se os jovens soubessem ler um idioma diferente, mas não conseguissem escrever nesse idioma, isto é, ter contato com celulares, computadores, videogames, entre outros, não implica necessariamente que a pessoa saiba como criar conteúdos para eles.

O *Scratch* foi criado em 2007 (MIT, 2007) e, conforme Schnur (2015), permite que qualquer pessoa comece a programar, sem conhecimentos prévios. Em pouco tempo de contato, o *software* permite que o usuário seja capaz de produzir jogos, animações, pequenas histórias, entre outras coisas. Segundo o autor, sua praticidade e simplicidade são fatores cativantes para novos usuários. Resnick et al (2009) mencionam que o *Scratch* é uma linguagem mais maleável, pois foi modificado de acordo com o *feedback* e sugestões gerados pelos usuários. É, também, uma linguagem mais significativa, pois o usuário aprende a programar e aprende com o programa, ou seja, o *Scratch* é capaz de gerar um contexto de aprendizado muito mais significativo do que a sala de aula tradicional.

Além disso, segundo os autores, há um terceiro elemento no *Scratch* que o diferencia da maioria das outras linguagens de programação: a facilidade de interagir com outros projetos, de outras pessoas, em uma comunidade com membros em mais de 150 diferentes países. Nas palavras dos autores, “*three core design principles for Scratch: Make it more tinkerable, more meaningful, and more social than other programming environments*”

(RESNICK et al., 2009), ou seja, o *Scratch* tem o objetivo de ser mais significativo e social que outros programas. Com essa facilidade, em sala de aula, os alunos conseguiriam facilmente compartilhar seus projetos entre si, e, inclusive, aprender ao observar outras pessoas. Essa foi a metodologia aplicada pelos professores em sala.

A comunidade *Scratch* é livre, qualquer pessoa pode ter acesso. Isso possibilita uma gama de possibilidades, entre elas, o uso pelo professor do conteúdo disponível gratuitamente, que pode usar, em suas aulas, jogos educativos ou animações, previamente feitos por outros usuários. Ressaltamos que o *Scratch* pode ser utilizado em sala de aula de diferentes formas, tudo dependerá da metodologia do professor em questão.

1.3. Introduzindo o *Scratch* em sala de aula

Quando falamos em linguagem de programação, dificilmente pensamos nela como, de fato, uma linguagem. Segundo Steven Pinker (2003) a linguagem é definida como uma habilidade complexa e especializada, que se desenvolve espontaneamente na criança sem qualquer esforço consciente ou instrução formal. Ou seja, a linguagem é uma capacidade natural humana.

Por ser uma atividade complexa e especializada, alguns cognitivistas definiram linguagem como instinto. A tese mais famosa desse conceito foi desenvolvida por Noam Chomsky (2002), que acredita que cada pessoa, ao enunciar uma frase, cria uma nova combinação de palavras, ou seja, toda frase que enunciamos não é apenas uma repetição do que já ouvimos, mas sim um conjunto de novas palavras que compreendemos.

A BNCC estabelece como uma de suas competências gerais o uso de linguagens digitais em sala de aula, o que envolve tanto o aluno quanto o professor. No entanto, segundo Aranha, Silva e Araújo (2014),

No Brasil, o aprendizado dos fundamentos da Ciência da Computação é reservado apenas àqueles que optam por cursos de graduação na área. [...] em outros países como os Estados Unidos, Canadá e Israel, o ensino de programação já é aplicado também no ensino básico. [...] Para que esse movimento se desenvolva, contudo, primeiro é necessária a preparação dos professores de outras disciplinas para aplicarem esses conceitos em sala de aula. (SILVA; ARAUJO; ARANHA, 2014, p. 381).

Aranha, Silva e Araújo (2014) utilizaram o *Scratch* em seus estudos para poder ensinar programação aos professores da rede estadual e municipal de ensino do Rio Grande do Norte,

com o objetivo de usar o *Scratch* como ferramenta auxiliar na prática docente e, também, como fonte de atividades lúdicas para ensinar computação através da Computação Desplugada (conceito cunhado por Bell; Witten; Fellows, 2011 apud Silva; Araujo; Aranha, 2014), que se baseia na não utilização do computador para o ensino de fundamentos da Ciência da Computação.

Atualmente, na formação de professores, algumas licenciaturas oferecem a disciplina de lógica de programação, permitindo, assim, que futuros professores tenham acesso a ela. Corroborando com a afirmação de Resnick (2012), consideramos que este é um caminho para que as pessoas comecem a "escrever" tecnologia, e não apenas ler.

Para o desenvolvimento do nosso projeto, a formação dos professores foi de grande ajuda, sendo um deles graduando em licenciatura em matemática e o outro em tecnologia de análise e desenvolvimento de sistemas. Ambos possuíam contato com programação de computadores e já haviam tido um contato prévio com a plataforma *Scratch*. Consideramos, assim, que o *Scratch* ajuda jovens a aprenderem de maneira criativa, refletir de maneira sistemática e trabalhar de forma colaborativa – habilidades essenciais para a vida no século 21.

2. Resultados principais

Para estruturar um código computacional, vários temas precisam ser trabalhados, como lógica de programação, o planejamento da solução de um problema, a organização do pensamento e, ainda, um conhecimento sobre expressões matemáticas e variáveis. Com isso, alunos com diferentes habilidades podem explorar diversos conhecimentos com a Programação de Computadores.

O curso de Lógica de Programação com a ferramenta *Scratch* foi planejado para alunos do 8º e 9º ano do Ensino Fundamental. Inicialmente, planejou-se dividir em duas turmas, porém, a quantidade de alunos do 8º ano permitiu a união entre as duas séries em um único grupo. Em todas as aulas, houve faltas de alguns estudantes, contudo, a motivação para desenvolver um jogo, aliada com a facilidade de manuseio da plataforma *Scratch*, formaram pontos fortes para que os alunos pudessem superar suas dificuldades.

No início, os bolsistas aplicaram uma avaliação simples com 4 questões, contendo testes de raciocínio lógico e matemático. Obteve-se uma média de 50% de acertos por parte dos alunos, que tiveram mais dificuldades com o tema de raciocínio lógico. Com isso, os professores conseguiram ter uma base do nível de conhecimento da turma sobre o assunto, e assim, obter um planejamento mais assertivo e eficaz das aulas que seriam ministradas.

No encerramento do curso, todos saíram com os conhecimentos básicos exigidos para o desenvolvimento da programação computacional. Os alunos levaram para casa um jogo que possuía, em sua estrutura, variáveis, expressões lógicas e matemáticas, personagens que eles criaram e importaram para o jogo e, além disso, compreenderam o conceito de dividir um problema muito complexo em pequenas partes, para facilitar o desenvolvimento do código e chegar mais rápido ao objetivo. Ao fim do projeto, não foi aplicado um novo questionário de raciocínio lógico, mas, ao longo das últimas aulas, os bolsistas puderam observar um amadurecimento no raciocínio dos alunos, que davam sugestões mais assertivas sobre o tema sendo desenvolvido em sala.

Para Resnick, essas habilidades serão úteis não apenas para cientistas da computação, mas para qualquer pessoa, independentemente da idade, da experiência, do interesse ou da profissão que optar por seguir.

2.1. Aulas ministradas

Durante o semestre, foram ministradas aulas semanais de 100 minutos, no período vespertino, para uma média de 9 alunos. Em cada aula, foram trabalhados assuntos diferentes e foram feitas gincanas, para uma melhor absorção do conteúdo por parte dos estudantes.

Para as aulas, consideramos a necessidade de que as atividades propostas promovam a interação, não apenas com o computador, mas também com os colegas da turma, visando criar um ambiente colaborativo, onde a troca de informações e a busca de novas hipóteses se façam presentes, conforme apontam Correia e Silva (2005).

Os bolsistas iniciaram o curso trabalhando o conceito de Algoritmo. Posteriormente, laços, estrutura condicional, variáveis e programação com jogos foram temas explorados, todos com a plataforma do *Scratch*. Uma das grandes vantagens desta plataforma é a sua versão online, para a qual o aluno não precisa possuir um bom computador e nem ter conhecimento para a instalação do programa.

Além de observarmos os alunos em sala, para avaliar a absorção dos conteúdos passados, utilizamos a ferramenta Kahoot!⁵, para analisar e estimar o nível de compreensão dos estudantes. Este recurso possibilitou uma visão geral de aprendizado do grupo de maneira muito divertida, descontraída e tecnológica, pois os alunos respondiam algumas questões por meio do computador, assemelhando-se a um jogo de perguntas e respostas.

⁵ Kahoot! É uma plataforma de aprendizado baseada em jogos, usada como uma tecnologia educacional em escolas e outras instituições.

2.2. Dificuldades encontradas

O *Scratch* oferece uma facilidade em seu uso, que é a não-necessidade de instalação. Apenas acessando seu site, qualquer pessoa pode, mesmo sem registro, começar a programar, ou seja, com apenas um link, qualquer aluno poderia programar e exercitar em suas casas os conhecimentos adquiridos em sala de aula. Entretanto, isso não ocorreu em nenhum momento. Mesmo com avisos constantes em todas as aulas sobre essa praticidade, nenhum aluno relatou ter utilizado o *Scratch* em casa, portanto, eles só viam o ambiente de programação em aula, uma vez por semana.

O número decrescente de alunos a cada aula foi um fator desmotivante para os bolsistas, mas que não abalou seus desempenhos como professores. O projeto foi feito no contraturno dos alunos, ou seja, era aplicado à tarde para alunos que tinham aula de manhã. A escola oferecia almoço para os alunos, os quais moravam perto da escola o suficiente para poderem ir embora de ônibus ou a pé.

Ao ponto de vista dos bolsistas, o modo como a escola opera deve mudar, de forma que, ao invés de apenas um professor por sala, houvessem dois profissionais, com um profissional lecionando e o outro tirando rápidas dúvidas e mantendo a ordem da sala de aula, que foi o que ocorreu, de forma exemplar, durante as aulas ministradas no projeto. Em contrapartida, a escola pode não ter condições para isso e, então, a solução seria diminuir o número de alunos por sala e, conseqüentemente, o número de alunos totais na escola. Tal solução, mesmo que drástica, reduziria o mal comportamento dos alunos, segundo Smith e Glass (1979).

É importante ressaltar, também, as dificuldades que surgiram para cumprir o cronograma do curso no tempo planejado, por conta de uma greve que impossibilitou a aplicação de uma aula. Logo, para cumprir o planejamento e atingir o objetivo, os bolsistas ajustaram as aulas para conseguirem passar o conteúdo do curso por completo aos estudantes, sem aumentar a quantidade de aulas do cronograma.

3. Conclusão

Ensinar programação para as pessoas cada vez mais cedo é algo que vem sendo discutido nos últimos anos. Para Steve Jobs, por exemplo, todos deveriam aprender a programar um computador, pois, segundo ele, isso ensina você a como pensar. Não há como discordar dessa afirmação, visto que, realmente, ter argumentos lógicos para uma tomada de decisões parece ser adequado para qualquer momento da vida.

Como afirma José Armando Valente (2009), as características dos computadores, que incluem a expressão do que o aprendiz está pensando em termos de uma linguagem formal e precisa e a execução do que ele está pensando em termos de resultados fiéis e imediatos, estão presentes nas atividades de programação e auxiliam o aprendiz a alcançar a fase de compreensão de conceitos. O aprendiz pode, então, refletir sobre os resultados de suas ações e ideias, sendo essa reflexão o mecanismo pelo qual o aprendiz se torna consciente de seu conhecimento e, assim, pode transformar seus esquemas mentais em operações e noções mais complexas.

Essas afirmações reforçam a relevância do tema de programação de computadores, não só para atuantes na área e cientistas, mas para qualquer pessoa de qualquer idade, evidenciando os benefícios em raciocínio e estratégias para resolução de problemas que crianças, que ainda estão em formação mental, podem adquirir explorando essas áreas.

Foi interessante observar, ao longo das aulas, o desenvolvimento lógico dos alunos. Nas últimas aulas do projeto, os estudantes já tinham concepções de que em um jogo com inimigos, por exemplo, quando o inimigo morre, não significa que ele deixou de existir; ele apenas está invisível ou em uma posição que seja impossível de o usuário ver. Perceberam, ainda, que em jogos com o cenário infinito, em que a personagem principal apenas precisa pular, é preciso fazer a personagem principal andar, e que o cenário se movimento já é capaz de criar toda essa ilusão. Concluimos, assim, que o uso da ferramenta *Scratch* foi eficaz para o cumprimento dos nossos objetivos.

REFERÊNCIAS

BELL, T.; WITTEN, I. H.; FELLOWS, M. *Computer science unplugged*. Ensinando ciência da computação sem o uso do computador. Tradução coordenada por Luciano Porto Barreto, 2011.

BRASIL. Ministério da Educação. Secretaria da Educação Básica. *Base nacional comum curricular*. 2016. Disponível em: <<http://basenacionalcomum.mec.gov.br/>>. Acesso em: 08/06/2018.

CORREIA, L. H. A.; SILVA, A. J. de C. *Computador Tutelado*. Lavras: UFLA/FAEPE, 2005.

CHOMSKY, N. *Syntactic structures*. [S.l.]: Walter de Gruyter, 2002.

FINI, M. I.; MICELI, P. *Currículo do Estado de São Paulo: Matemática e suas tecnologias*. 2012.

KAFAI, Y. B.; BURKE, Q. *Computer programming goes back to school*. PhiDelta Kappan, SAGE Publications Sage CA: Los Angeles, CA, v. 95, n. 1, p.61–65, 2013.

MIT, M. L. *About Scratch*. 2007. Disponível em: <<https://Scratch.mit.edu/about/>>. Acesso em: 08/06/2018.

PEREIRA, L. *Escolas Defendem Ensino de Programação a Crianças e Adolescentes*. 2013. Disponível em: <<http://olhardigital.uol.com.br/noticia/-escolas-defendem-ensino-de-programacao-acriancas-e-adolescentes/35075>>. Acesso em: 08/06/2018.

PINKER, S. *The language instinct: How the mind creates language*. [S.l.]:Penguin UK, 2003.

RESNICK, M. *Let's teach kids to code*. 2012. Disponível em: <<https://www.ted.com/talks/mitchresnickletsteachkidstocode>>. Acesso em: 08/06/2018.

RESNICK, M. et al. *Scratch: programming for all*. Communications of the ACM, ACM, v. 52, n. 11, p. 60–67, 2009.

SCRATCH. *About Scratch (Scratch Documentation Site)*. Disponível em: <<https://Scratch.mit.edu/about/>>. Acesso em: 03/07/2018.

SCHNUR, C. A. *A linguagem Scratch como ferramenta para o desenvolvimento da inteligência lógico-matemática: Um estudo de caso*. 49 p.— Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, 2015.

SERRANO, F. Geração Geek. *Revista Exame Informática*. São Paulo: Editora Abril, 2014.

SILVA, T. R. da; ARAUJO, G. G. de; ARANHA, E. H. da S. Oficinas itinerantes de *Scratch* e computação desplugada para professores como apoio ao ensino de computação—um relato de experiência. In: *Anais do Workshop de Informática na Escola*. [S.l.: s.n.], 2014. v. 20, n. 1, p. 380.

SIMÕES, L.; GOUVEIA, L. Geração net, web 2.0 e ensino superior. *Cadernos de Estudos Mediáticos*, Universidade Fernando Pessoa Porto, v. 6, p. 21–32, 2008.

SMITH, Mary Lee; GLASS, Gene V. *Relationship of Class-Size to Classroom Processes, Teacher Satisfaction and Pupil Affect: A Meta-Analysis*. 1979.

VALENTE, J. A. (Org). *O computador na sociedade do conhecimento*. Campinas, SP: UNICAMP/NIED, 1999.

VYGOTSKY, L. S. *Thought and language-revised edition*. The MIT Press, 1986.