Ensino de programação para crianças e adolescentes: um estudo exploratório

Vitor H. Gomes[1], Renata F. Pontes[1], Carlos A. S. Camelo[1], Givonaldo A. S. Cavalcanti[1], Mirko B. Perkusich[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia da Paraíba Campus Monteiro (IFPB) – Monteiro, PB - Brazil

vitor_gomes_18@live.com, renata.pontes@ifpb.edu.br,
{carlos.avelino2.0, givonaldoc, mperkusich}@gmail.com

Abstract. *Several researches have been made showing the importance of teaching and learning programming in different areas, not limited to computer courses. By developing skills that are important as Logical thinking and logic programming, many of these teaching and learning programming studies geared toward the elementary and secondary education. With that, several visual programming languages have been created such as Scratch, Alice, Kodu, Greenfoot and App Inventor for Android. The aim of this study is to conduct a survey and characterization of existing tools for teaching programming to children and teenagers.*

Resumo. *Várias pesquisas já foram feitas demonstrando a importância do ensino e aprendizagem da programação em diferentes áreas, não se limitando apenas aos cursos de computação. Por desenvolver habilidades que são essenciais como o raciocínio lógico e a lógica de programação, muitos desses estudos sobre ensino e aprendizagem de programação estão voltados para o ensino fundamental e médio. Com isso, diversas linguagens de programação visuais foram criadas, tais como Scratch, Alice, Kodu, Greenfoot e App Inventor for Android. O objetivo deste trabalho é realizar um levantamento e caracterização das ferramentas existentes para o ensino de programação para crianças e adolescentes.*

1. Introdução

Com o desenvolvimento da tecnologia da informação, o Ensino de Ciência da Computação (ECC) se torna essencial. O computador tornou-se uma ferramenta tão necessária no dia, a dia que é importante desenvolver recursos humanos não apenas familiarizados com computadores, mas capazes de criar tecnologias.

No ECC, a habilidade de pensamento computacional é desenvolvida. Esse tipo de pensamento deve ser considerado uma habilidade fundamental para estudantes no século 21 (assim como leitura, escrita, aritmética, etc.), e programação deve ser uma habilidade que todo estudante deve adquirir. Em diversos países, tais como Estados Unidos, Coreia do Sul e Inglaterra, o ensino de programação nas escolas já é realidade.

Para o ensino de conceitos simples de programação, tais como contadores e acumuladores, há evidências de sucesso usando ferramentas simples, como calculadoras

DOI: 10.5753/cbie.wcbie.2017.490

mecânicas [SOVIC, 2014]. Por outro lado, para o ensino de conceitos mais avançados de programação procedural ou baseada em eventos, diversas ferramentas de programação visuais foram propostas na literatura, tais como o Scratch [RESNICK et al., 2009], Alice [PAUSCH, 1995], Kodu [STOLEE, 2010], Greenfoot [KOLLING, 2010] e App Inventor.[MIT app Inventor, 2017]

Em Bombasar et al. (2015), várias ferramentas utilizadas para o ensino e aprendizagem do pensamento computacional foram exploradas por meio de uma revisão sistemática na literatura. De acordo com eles, as ferramentas mais populares são: Scratch, Alice, App Inventor e Lego Mindstorms [Lego Mindstorms, 2017]. Por outro lado, as ferramentas não foram caracterizadas e a eficácia das mesmas não foi investigada. Além disso, o estudo limitou-se a avaliar ferramentas apresentadas na literatura. Por exemplo, o Code.org [Code.org, 2017], uma popular ferramenta para ensino de programação com registro de mais de 300 milhões de acessos, não foi encontrada nesse estudo. Por outro lado, ferramentas como Scratch e App Inventor estão sendo estudadas por diversos autores, seja em levantamentos bibliográficos, seja em estudos de caso, a exemplo de [Batista et al., 2016], [Silva et al., 2016] e [Gomes, Melo, 2013].

Nessa pesquisa, busca-se realizar um levantamento e a caracterização dessas e de outras ferramentas existentes para o ensino de programação para crianças e adolescentes. Além disso, busca-se explorar a literatura para identificar evidências acerca do uso e da eficácia dessas ferramentas.

A Seção 2 desse artigo possui a metodologia que será abordada no trabalho. A Seção 3 descreve o que foi levantado até este momento, alguns resultados obtidos e o que será feito em seguida. A Seção 4 possui algumas conclusões de acordo com os dados já obtidos.

2. Metodologia

Para a realização da pesquisa serão executadas as seguintes atividades:

Levantamento de ferramentas: realizar uma busca na internet para identificar ferramentas disponíveis para o ensino de programação para crianças e adolescentes;

Análise das ferramentas: caracterizar e comparar as ferramentas de acordo com suas principais funcionalidades e paradigmas de programação.

Revisão Sistemática da Literatura (RSL): o protocolo da RSL será definido e a RSL será executada de acordo com o mesmo, para identificar o estado da arte da literatura acerca do tema da pesquisa;

Elaboração de relatório com as conclusões: as conclusões finais serão levantadas, onde serão observadas as ferramentas existentes no mercado que estão de fato sendo estudadas pela literatura e quais delas já têm alguma comprovação de eficácia baseadas nos estudos já feitos. Além disso, será verificado, dessas ferramentas, se existem algumas que estão sendo utilizadas, mas não estão sendo estudadas pela literatura.

3. Resultados e Discussão

Nesse primeiro momento, foram feitos levantamentos mais detalhados sobre cada ferramenta encontrada utilizando uma lista de características previamente definidas. Essas características foram: ter uma linguagem visual ou não, podendo ser 2D ou 3D; suportar uma linguagem profissional, explicitando qual; se a ferramenta é paga ou não, já que isso influencia a possibilidade de uso; quais idiomas suporta, verificando sua possível abrangência no mundo e no Brasil; e como ela contextualiza o raciocínio de programação,

podendo ser através de jogos, histórias, vídeos, ou criação direta de programas. Com essas informações, foi possível gerar uma tabela, que resume e facilita a visualização comparativa de cada uma dessas características, podendo ser utilizada como guia para auxílio na escolha e utilização das ferramentas. Além disso, foi possível observar que algumas características têm correlação entre si.

Tabela 1. Comparativo de Ferramentas

Ferramenta	Público alvo*	Visual	Linguagem	Nível de abstração	Contexto (Criação)	Idioma	Plataforma de ensino	Plataforma de desenvolvimento
Robomind	F/M/S	Não	Java / ROBO	Baixo	Programar robôs.	22 idiomas	Não	Sim
Scratch	F	Sim	Scratch	Alto	Histórias, jogos	40 idiomas.	Sim	Não
Kodu	F	Sim	Kodu	Alto	Jogos	23 idiomas	Não	Sim
Greenfoot	M/S	Sim	Java	Baixo	Jogos e vídeos	Inglês	Não	Sim
StarLogo TNG	F/M	Sim	StartLogo	Alto	Jogos.	Inglês	Não	Sim
AppInventor	F	Sim	Java	Baixo	APP's android	Inglês, Português	Não	Sim
Kodable	F/M	Sim	Baseia-se em JavaScript	Alto	Aplicativos, jogos e sites	Inglês	Sim	Não
Alice	M/S	Sim	Java	Baixo	Histórias, animações	Inglês	Sim	Não
Hackety Hack	M/S	Não	Ruby	Baixo	Atividades em Ruby.	Inglês	Não	Sim
Code. org	F/M	Sim	JavaScript	Baixo	Vídeos, jogos e app.	Inglês	Sim	Não
Sparki	F/M/S	Sim	C/C++	Baixo/ Alto	Programar robôs	Inglês	Não	Sim
Programaê	F	Sim	Code. Org, Scratch	Alto	Histórias.	Português Inglês	Sim	Não
Robotopia	F	Sim	Google Blockly	Alto	Programar robôs	Inglês	Sim	Não
CodeCombat	F/M	Não	Python, JavaScript	Baixo	É um Jogo	50 Idiomas	Sim	Não
Lego Mindstorms	M	Sim	Lego Mindstorms NXT	Alto	Programar robôs	Inglês	Não	Sim

*F-Fundamental, M-Médio, S-Superior

Foram observadas as correlações de que as ferramentas utilizadas para o ensino fundamental têm uma relação direta com linguagens visuais, alta abstração em termos de linguagem de programação e, geralmente, são plataformas focadas no ensino e não no desenvolvimento em si. Nas ferramentas utilidadas para o ensino médio e superior, percebeu-se uma relação com abstração mais baixa em termos de programação. Além disso, elas são plataformas de desenvolvimento que utilizam uma linguagem de programação profissional. Esse segundo grupo de ferramentas não necessariamente exclui as outras características, podendo assim ter características para o ensino de programação para crianças e, ao mesmo tempo, ter as características necessárias para o

ensino de programação para adolescentes e adultos, tornando-se ferramentas mais "amplas".

O próximo passo do trabalho é fazer uma revisão sistemática na literatura para verificar, quais dessas ferramentas estão sendo estudadas e citadas nos estudos sobre esse assunto, quais já têm relatos de eficácia de uso e quais ainda não aparecem nos estudos acadêmicos.

4. Conclusões

Com base nos estudos feitos até agora, foi possível fazer um levantamento das ferramentas existentes para o ensino de programação à crianças e adolescentes, gerando uma tabela comparativa entre elas através de características observadas nas mesmas. Com essas comparações, já foi possível verificar relação entre algumas características, de tal forma que se observou levantamentos padrões entre ferramentas que tentam atingir o mesmo público-alvo, sendo esse fundamental, médio ou superior.

Observou-se também que algumas delas aparecem na literatura acadêmica, sendo necessário, agora, uma revisão sistemática a fim de descobrir quais delas estão sendo estudadas pela a academia e quais não.

Referências

KOLLING, M. (2010) "The Greenfoot Programming Environment". ACM Trans. Comput. Educ., v. 10, n. 3, 14.

PAUSCH, R., BURNETTE, T., CAPEHEART, A.C., CONWAY, M., COSGROVE, D., DELINE, R., DURBIN, J., GOSSWEILER, R., KOGA, S., WHITE, J. (1995) IEEE Computer Graphics and Applications.

RESNICK M., MALONEY, J., MONROY-HERNÁNDEZ, A., EASTMOND, E., BRENNAN, K., MILLNER, A., ROSENBAUM, E., SILVER, J., SILVERMAN, B., KAFAI, Y., (2009) "Scratch: Programming for all. Communications of the ACM", v. 52, n. 11, p. 60-67.

STOLEE, K. (2010) Kodu Language and Grammar Specification. Microsoft Resarch whitepaper.

SOVIC, A., JAGUST, T., SERSIC, D. (2014) "How to teach basic university-level programming concepts to first graders?" IEEE Integrated STEM Education Conference (ISEC), Princeton, p. 1-6.

MIT APP INVENTOR. D. (2017) disponível em: <http://appinventor.mit.edu/explore/>. Acesso em 14 de julho.

BOMBASAR, J., RAABE, A., MIRANDA, E., SANTIAGO, R. (2015) "Ferramentas para o Ensino-Aprendizagem do Pensamento Computacional: onde está Alan Turing?" In: Anais do XXVI Simpósio Brasileiro de Informática na Educação - SBIE, p. 81-90.

LEGO MINDSTORMS. (2017) Disponível em: <https://www.lego.com/en-us/mindstorms>. Acesso em 14 de julho.

CODE.ORG - Anybody can learn. (2017) Disponível em:<https://code.org>. Acesso em 14 de julho.

BATISTA, E. J. S.; CASTRO JÚNIOR, A.; CANTERO, S.; BOGARIM, C. A. C.; LARREA, A. A. (2016) “Uso do Scratch no ensino de programação em Ponta Porã: das séries iniciais ao ensino superior”. In: Anais do XXII Workshop de Informática na Escola. Uberlândia, MG.

SILVA, G. T.; SOUZA, J. L.; SILVA, L. A. M. (2016) “Aplicação da Ferramenta Scratch para o Aprendizado de Programação no Ensino Fundamental I”. In: Anais do XXII Workshop de Informática na Escola. Uberlândia, MG.

GOMES, T. C. S.; MELO, J. C. B. (2013) “App Inventor for Android: Uma Nova Possibilidade para o Ensino de Lógica de Programação”. In: Anais dos Workshops do II Congresso Brasileiro de Informática na Educação. Campinas, SP.