



UM ESTUDO SOBRE AVALIAÇÃO DE SOFTWARES EDUCATIVOS PARA O ENSINO DE PROGRAMAÇÃO

A STUDY ON EDUCATIONAL SOFTWARE EVALUATION FOR PROGRAMMING TEACHING

- **Romário Ferreira de Pontes** (Universidade Federal da Paraíba/Campus IV – romario.pontes@dce.ufpb.br)
- **Williane Rodrigues de Almeida Silva** (Universidade Federal da Paraíba/Campus IV – williane.rodrigues@dce.ufpb.br)
- **Thereza Patrícia Pereira Padilha** (Universidade Federal da Paraíba/Campus IV – thereza@dce.ufpb.br)

Resumo:

Cada vez mais as mídias interativas, sobretudo softwares educativos, estão sendo inseridos no ambiente escolar possibilitando, assim, aos professores o uso de recursos extras no processo de ensino-aprendizagem. Avaliar um software educativo significa analisar as características de sua interface, bem como os seus efeitos no processo de aprendizagem. Este artigo apresenta os resultados de um estudo de avaliação de quatro softwares educativos (Code.org, Scratch, RoboMind e LightBot) para o ensino de Programação, utilizando o método de Reeves, com o intuito de subsidiar a escolha desses recursos pelos professores. Para cada software avaliado, é mostrado o quanto cada critério (pedagógico e de interface) se aplica. De modo geral, os resultados da avaliação dos critérios pedagógicos nos softwares citados se mostraram similares na determinação dos conceitos do Método de Reeves. Já em relação à avaliação dos critérios de interface, percebeu-se que a maioria deles recebeu um conceito positivo mas os resultados dos softwares não foram tão semelhantes.

Palavras-chave: software educativo, avaliação, ensino de programação.

Abstract:

Increasingly interactive media, mainly educational software, are being available in the educational environment and allowing teachers use additional resources in the teaching-learning process. To evaluate educational software means to analyze its interface and effects on the learning process. This paper presents an evaluation of four educational software (Code.org, Scratch, RoboMind and LightBot) to programming teaching using Reeves method, in order to provide support in the resources choosing by teachers. It also shows how much each (pedagogical and interface) criterion is valued in the educational software. Overall, the results of the pedagogical criteria evaluation in the cited software, using Reeves method, shown similar. But the interface criteria evaluation in the four software, it was noticed that most of them received a positive concept but the results among the software were not so similar.

Keywords: educational software, evaluation, programming teaching.





1. Introdução

Uma preocupação nas Instituições de Ensino Superior é a evasão universitária, definida como a saída do aluno de seu curso, sem concluí-lo. Sabe-se, ainda, que os índices de evasão são maiores nos primeiros períodos dos cursos (BARDAGI e HUTZ, 2009). No caso dos cursos da área de Computação e Engenharia, outra preocupação é o alto índice de retenção em disciplinas de programação. Tanto a evasão quanto à retenção, ocorrem em todas as instituições de ensino públicas e particulares brasileiras. No caso das instituições federais de ensino superior, o fracasso do aluno é um assunto importante e que merece atenção (CAMPELLO e LINS, 2008).

Há anos os altos índices de evasão e retenção nas disciplinas iniciais de programação têm despertado o interesse e a preocupação de pesquisadores e educadores em todo o mundo, como relata (SCHULTZ, 2003). Sabe-se, segundo (BRANCO NETO e SCHUVARTZ, 2007), que alunos iniciantes, ao se depararem com a disciplina de programação, sentem-se incapazes de programar devido ao conjunto de habilidades que a programação exige, que são: resolução de problemas, habilidade matemática, abstração, entre outras.

Além das aulas presenciais ou a distância, professores de disciplinas de programação podem recomendar a seus alunos, como material complementar, livros, softwares educativos (SEs), apostilas, vídeo-aulas, entre outros. No caso de SEs, muitos professores têm dificuldades para indicá-los porque há uma diversidade na Internet e, também, devido muitos softwares se apresentarem como educativo, no entanto, não cumprem o papel de aprendizagem. Diante disso, este trabalho visa avaliar quatro SEs da área de Programação a fim de orientar professores na indicação deste tipo de material, utilizando o método de Reeves, que engloba critérios pedagógicos e de interface.

2. Softwares educativos na área de programação

Um software educativo é todo aquele software que possa ser usado com algum objetivo educacional, pedagogicamente defensável, por professores e alunos, independente do objetivo para o qual ele foi criado (TEIXEIRA e BRANDÃO, 2003). Um SE, conforme suas características pedagógicas, é classificado como jogo, simulação, tutoriais, programação, entre outros (VALENTE, 1999). Os SEs do tipo programação, objetos de estudo deste trabalho, estão voltados para a criação de protótipos de programas, estimulando o raciocínio quanto à ordem de ações a serem executadas para resolver determinado problema. Dentre os softwares educativos on-line que se propõem auxiliar a aprendizagem de programação, destacam-se:

- **Code.org:** dispõe de uma série de tutoriais para desenvolver uma sequência correta de ações a fim de atingir o objetivo (CODE, 2016);
- **Scratch:** permite a programação de suas próprias histórias, jogos e animações interativas, e ainda partilhar as suas criações (MALONEY et al., 2008);
- **RoboMind:** voltado para a programação das ações (mover itens, andar, olhar para os lados e pintar) de um robô em um ambiente (HALMA, 2016);
- **LightBot:** permite a programação de um robô através blocos para navegar por plataformas labirínticas, seguindo instruções pré-determinadas (SOUZA e PRADO, 2015).





Para este trabalho, foram considerados os quatro SEs citados, por serem plataformas livres que oferecem fácil manuseio e atividades relacionadas a jogos, o que incentiva o aprendizado da programação.

3. Método de Reeves

Avaliar um software vai além da preocupação com defeitos de funcionamento, dependendo do tipo de software e de seu grupo de usuário, diferentes fatores podem ser mais (ou menos) importantes (PFLEEGER, 2004; PRESSMAN, 2006). Na literatura, existem várias abordagens para avaliação de SEs abordando aspectos pedagógicos e também de usabilidade do software. Como o método de Reeves contempla estes dois aspectos, e também se apresenta como um *checklist* simplificado, este foi escolhido para o desenvolvimento deste trabalho.

O método de Reeves consiste em 24 critérios, sendo 10 relacionados à interface e 14 pedagógicos (CAMPOS, 1996). Para cada critério, há uma escala representada por uma seta dupla em que deve ser realizada uma marcação do quanto este critério se aplica ao software. Na extremidade esquerda da seta fica situado o conceito mais negativo e à direita o mais positivo. Os conceitos variam de acordo com cada critério avaliado. A Figura 1 apresenta os critérios de forma enumerada, sendo os de interface representado pela letra inicial “I” e os pedagógicos pela letra “P”, bem como os conceitos envolvidos em cada critério.

Critério	Conceitos envolvidos
I1 - facilidade de uso	Difícil (Dif) /Fácil (Fac)
I2 - navegação	Difícil (Dif) /Fácil (Fac)
I3 - carga cognitiva	Não (Não) /Gerenciável (Ger)
I4 - mapeamento	Nenhum (Nen) /Poderoso (Pod)
I5 - design de tela	Princípios violados (Pvi) /Princípios respeitados (Pre)
I6 - compatibilidade especial do conhecimento	Incompatível (Inc) /Compatível (Com)
I7 - apresentação da informação	Confusa (Con) /Clara (Cla)
I8 - integração das mídias	Não coordenada (NCo) / Coordenada (Coo)
I9 - estética	Desagradável (Des) /Agradável (Agr)
I10 - funcionalidade geral	Não funcional (NFu) /Altamente funcional (Afu)
P1 - epistemologia	Objetivista (Obj) /Construtivista (Con)
P2 - filosofia pedagógica	Instrutivista (Ins) /Construtivista (Con)
P3 - psicologia subjacente	Comportamental (Com) /Cognitiva (Cog)
P4 - objetividade	Precisamente focalizado (Pre) /N-Focalizado (N-F)
P5 - sequenciamento instrucional	Reducionista (Red) /Construtivista (Con)
P6 - validade experimental	Abstrato (Abs) /Concreto (Con)
P7 - papel do instrutor	Provedor de Materiais (Pro) /Agente (Age)
P8 - valorização do erro	Aprendizado sem Erro (ASE) /Aprendizado com a experiência (ACE)
P9 – motivação	Extrínseca (Ext) /Intrínseca (Int)
P10 - estruturação	Alta (Alt) /Baixa (Bai)
P11 - acomodação de diferenças individuais	Não existente (Nex) /Multi-facetada (Mul)
P12 - controle do aluno	Não existente (Nex) /Irrestrito (Irr)
P13 - atividade do usuário	Matemagênico (Mat) /Generativo (Ger)
P14 - aprendizado cooperativo	Não suportado (Nsu) /Integral (Int)

Figura 1. Critérios do método de Reeves.

A conclusão da avaliação se dá pela ligação de todas as marcações realizadas nas setas, resultando, assim, em um gráfico.

4. Resultados obtidos

As avaliações dos quatro SEs são resultado de atividades da disciplina de Softwares Educacionais, do curso de Licenciatura em Ciência da Computação, da Universidade X. Em cada avaliação, buscou-se verificar e registrar o quanto cada critério do método de Reeves se aplica ao software. Os resultados das avaliações estão disponibilizados na Figura 2.

Crt	Code.org	RoboMind	Scratch	LightBot
I1	Dif ← → Fac			
I2	Dif ← → Fac			
I3	Não ← → Ger			
I4	Nen ← → Pod			
I5	Pvi ← → Pre			
I6	Inc ← → Com			
I7	Con ← → Cla			
I8	NCo ← → Coo			
I9	Des ← → Agr			
I10	NFu ← → Afu			
P1	Obj ← → Con			
P2	Ins ← → Con			
P3	Com ← → Cog			
P4	Pre ← → N-F			
P5	Red ← → Con			
P6	Abs ← → Con			
P7	Pro ← → Age			
P8	ASE ← → ACE			
P9	Ext ← → Int			
P10	Alt ← → Bai			
P11	Nex ← → Mul			
P12	Nex ← → Irr			
P13	Mat ← → Gen			
P14	Nsu ← → Int			

Figura 2. Avaliação de SEs da área de Programação utilizando o método de Reeves.

Diante desses resultados, pôde-se observar que a maioria dos critérios de interface obteve conceitos mais relacionados à positivo do que negativo. Já o conceito mais negativo atribuído foi para o critério I8, considerando o software Code.org, porque este não dispõe de integração de mídias. De modo geral, o software LightBot foi o que apresentou melhor avaliação em termos de interface se comparados aos demais SEs. Considerando os critérios pedagógicos, as avaliações dos quatro SEs tiveram um comportamento similar na determinação dos conceitos. O critério P9, por exemplo, que obteve conceitos bastante negativos, foi devido estes SEs apresentarem uma motivação muito maior fora do ambiente de aprendizado.

5. Considerações finais

Devido aos altos índices de evasão e retenção em disciplinas iniciais de programação e, também, a diversidade de SEs disponíveis, torna-se necessário um estudo mais profundo sobre o propósito dos SEs. Diante deste cenário, este trabalho analisou quatro SEs com ênfase no ensino de programação, observando características de interface e pedagógicas,



segundo o método de Reeves, para orientar a escolha de um SE pelos professores. Apesar de uma abrangência restrita, este estudo serviu para obter conhecimento e experiência na observação de características presentes em SEs. De modo geral, reforça-se que o uso destes SEs tende a levar um desenvolvimento de habilidades para resolução de problemas devido a necessidade de estruturar uma sequência de ações dado um problema. Com relação aos SEs avaliados sob o método de Reeves, LightBot e Scratch foram considerados com maior potencial para o aprendizado de programação devido à facilidade de uso e diversidade de conteúdo.

Referências Bibliográficas

BARDAGI, M. P.; HUTZ, C. S. "Não havia outra saída": percepções de alunos evadidos sobre o abandono do curso superior. **Psico-USF (Impr.)** Itatiba, v. 14, n. 1, p. 95-105, Apr. 2009. Disponível em <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-82712009000100010>. Acesso em: 30/04/2016.

BRANCO NETO, W. C.; SCHUVARTZ, A. A. Ferramenta Computacional de Apoio ao Processo de Ensino-Aprendizagem dos Fundamentos de Programação de Computadores. In: 18º Simpósio Brasileiro de Informática na Educação, 2007, São Paulo. **Anais...** São Paulo: Mackenzi, 2007. p. 520-528

CAMPELLO, A. V. C.; LINS, L. N. Metodologia de análise e tratamento da evasão e retenção em cursos de graduação de instituições federais de ensino superior. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 28., 2008, Rio de Janeiro. **Anais eletrônicos...** Disponível em: <http://www.abepro.org.br/biblioteca/enegep2008_TN_STO_078_545_11614.pdf>. Acesso em: 21 mai. 2016.

CAMPOS, G. H. B. **Avaliação da Qualidade de Software Educacional**. Tese (Doutorado em Engenharia). Universidade Federal do Rio de Janeiro, Rio de Janeiro, 1996.

CODE.ORG. "CODE". Disponível em: <<https://code.org/about>>. Acesso em: 04/04/2016.

HALMA, A. "Robomind.net – Welcome to Robomind.net", the new way to learn programming. Disponível em: <<http://www.robomind.net>>. Acesso em: 04/04/2016.

MALONEY, J.; PEPPLER, K.; KAFAI, Y. B.; RESNICK, M.; RUSK, N. Programming by Choice. Urban Youth Learning Programming with Scratch. In: Symposium on Computer Science Education, 2008, Oregon. **Anais...** Oregon, 2008. p. 367 - 371.

PFLEEGER, S. L. **Engenharia de Software: Teoria e prática**. 2. ed. São Paulo: Prentice Hall, 2004.

PRESSMAN, R. S. **Engenharia de Software**. 6. ed. Porto Alegre: Bookman, 2006.





SCHULTZ, M. R. O. **Metodologias para Ensino de Lógica de Programação de Computadores**. Monografia de Especialização (Ciência da Computação). Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, 2003.

SOUZA, E. C.; PRADO, R. C. Lightbot: uma possibilidade para analisar a estratégia de resolução de problemas dos alunos e a apropriação de conceitos matemáticos. In: XII Congresso Nacional de Educação, 2015, Curitiba. **Anais...** Curitiba, 2015. p. 4983-4994.

TEIXEIRA, A. C.; BRANDÃO, E. J. R. Software educacional: difícil começo. **Revista Novas Tecnologias na Educação – RENOTE**, Porto Alegre, V. 1. 2003.

VALENTE, J. A. (org). **O computador na sociedade do conhecimento**. Campinas: Unicamp/NIED, 1999.

