

## **Avaliação de linguagens visuais de programação no ensino médio a partir da utilização do conceito de Robótica Pedagógica**

**José Inaldo dos Anjos Junior<sup>1</sup>, Péricles de Lima Sobreira<sup>1</sup>, Jauberth Weyll Abijaude<sup>1</sup>, Hellan Dellamycow Gomes Viana<sup>1</sup>, Levy Marlon Souza Santiago<sup>1</sup>**

<sup>1</sup>Departamento de Ciências Exatas e Tecnológicas  
Universidade Estadual de Santa Cruz (UESC) – Ilhéus, BA – Brazil

inaldoanjosjr@gmail.com, plsobreira@uesc.br, jauberth@uesc.br

dellamycow@gmail.com, levysantiago@gmail.com

**Abstract.** *In this paper, a study of software based on visual languages for programming teaching through the concept of Pedagogical Robotics is carried out. The analysis of the characteristics of these platforms helped the authors of this research to classify them according to usability criteria necessary for their future target audience: high school students beginning in the subjects of algorithms and programming languages.*

**Resumo.** *Neste trabalho realiza-se um estudo de softwares que utilizam como base linguagens visuais para o ensino de programação através do conceito de Robótica Pedagógica. A análise das características destas plataformas deu suporte para classificá-las de acordo com critérios de usabilidade necessários ao seu futuro público alvo: estudantes do ensino médio iniciantes nos assuntos de algoritmos e linguagens de programação.*

### **1. Introdução**

A popularização dos conhecimentos em ciência e tecnologia é relevante no mundo atual, pois no cotidiano das pessoas é muito difícil não se deparar com situações onde estes temas não estejam inseridos. Não obstante, o advento da Web 2.0 e a inclusão de novas tecnologias nas relações entre alunos e professores ganham força no cenário educacional [De Miranda et al. 2007].

Muitos alunos do Ensino Médio (EM) que ingressam em um curso superior de Ciência da Computação, e afins, demonstram grande dificuldade na abstração e execução de atividades inerentes a estas áreas de formação, principalmente nas disciplinas iniciais de Linguagem de Programação e Algoritmos, o que pode acarretar baixo rendimento, mudança de curso, alta retenção na disciplina e evasão escolar [Gomes et al. 2008, Gomes et al. 2012, Queiroz et al. 2016].

Com base nisto, a implementação do pensamento computacional em estudantes do EM poderia representar uma nova forma de se desenvolver, neste público, competências e habilidades que facilitassem a construção de soluções de problemas nas mais variadas áreas do conhecimento. Entretanto, a formação do adolescente no ensino de lógica e da programação computacional se apresenta como um desafio, visto que há uma grande complexidade nos termos que circundam este conteúdo [Barr and Stephenson 2011].

Trabalhos que analisam o uso do Scratch no processo de ensino-aprendizagem de programação apontam as dificuldades enfrentadas pelos estudantes na absorção dos conceitos transmitidos por esta plataforma [Aureliano and Tedesco 2012], e assim, outras técnicas de ensino para programação surgem com o objetivo de contornar as dificuldades e atrair, de forma mais perceptível, as ações tomadas pelos alunos em suas práticas computacionais.

Destas técnicas, podemos citar a implementação da Robótica Pedagógica (RP) como uma ferramenta promissora na transmissão de conhecimentos [D'Abreu et al. 2013, Trentin et al. 2013]. A Adoção da RP como diferencial de aprendizagem proporciona um conjunto de recursos para o desenvolvimento de práticas pedagógicas, estimulando o ensino de programação [Zanetti and Oliveira 2015].

Para tanto, é necessário o uso de uma ferramenta de programação visual e de um componente físico (hardware) que possibilite a interação da lógica implementada no software com o ambiente externo. Como plataforma de suporte de hardware a esta técnica, percebe-se uma tendência mundial na utilização de kits que oferecem experiências de aprendizagem estimulantes [Papavlasopoulou et al. 2017], e entre estas, destaca-se a plataforma *open source* Arduino [Arduino 2017], por proporcionar uma experiência agradável tanto para novatos quanto para especialistas em programação e eletrônica, através de experimentos do seu ambiente de desenvolvimento multiplataforma [Evans 2011].

Esta pesquisa, então, apresenta algumas ferramentas voltadas à RP que se encontram atualmente disponíveis no mercado, analisando suas características e realizando um comparativo entre elas. A organização deste trabalho segue o seguinte roteiro: Na seção 2 apresenta-se um conjunto de softwares que podem ser utilizados para a programação visual em contextos de RP; na seção 3 encontram-se descritas a metodologia e o protocolo da experimentação realizada para a análise de tais plataformas; na seção 4 discutem-se os resultados obtidos das experimentações realizadas, e; na seção 5 encontram-se descritas as conclusões e considerações finais do trabalho.

## **2. Softwares utilizados no ensino de RP**

Por meio da robótica o aluno analisa, manuseia, executa, depura, corrige, reexecuta e pratica. Deste modo, ele abstrai conteúdos puramente teóricos de forma prática. Ao se deparar com problemas propostos pelo educador, o aluno pode estabelecer uma linha de raciocínio capaz de executar tarefas de maneira mais hábil e objetiva.

Atualmente no Brasil existem vários trabalhos direcionados à aplicação de RP em sala de aula [Cardoso and Antonello 2015, Zanetti and Oliveira 2015, De Souza et al. 2016, Queiroz et al. 2016]. Estas iniciativas proporcionam ao estudante uma nova percepção da aprendizagem, permitindo que se adquira novas habilidades no contexto da robótica através de manipulações tecnológicas virtuais e reais.

Após um levantamento na literatura com ênfase nas principais plataformas atualmente disponíveis para programação visual voltados a robótica [Booth and Stumpf 2013, De Miranda et al. 2007, Aureliano and Tedesco 2012, Queiroz et al. 2016], cinco softwares merecem destaque: Ardublock [Ardublock 2017], Minibloq [Minibloq 2017], ModKit [ModKitMicro 2017], Physical Etoys [GIRA 2017] e S4A [Edutec 2017]. Estas ferramentas se encontram descritas nas próximas subseções, onde serão identificadas as suas

funcionalidades, ferramentas, comunidades, licenças comerciais e aspectos positivos e negativos.

### **2.1. Ardublock**

O Ardublock é uma ferramenta utilizada conjuntamente com a IDE (*Integrated Development Environment*) do Arduino como forma de plugin, possuindo suporte multilíngue, inclusive português. É compatível com os principais Sistemas Operacionais do mercado. Após ativado, permite que o usuário deixe de programar em linhas de código e passe a combinar blocos lógicos (repetição, decisão, etc.), facilitando o trabalho das pessoas que não tem familiaridade com linguagens de programação não-visuais.

### **2.2. Minibloq**

Software gratuito e *open source*, o Minibloq tem como principal objetivo aproximar a programação em componentes físicos e plataformas de robótica para iniciantes a partir da manipulação de blocos gráficos. Esta ferramenta, apesar de algumas funcionalidades interessantes (como a opção de geração de linhas de código em uma outra janela configurável pelo usuário), apresenta problemas de metaforização de alguns de seus blocos, percebidos durante a manipulação de sua interface (a programação de seus blocos se mostra um pouco confusa devido aos ícones serem poucos significativos).

### **2.3. ModKit Micro**

O ModKit Micro (MM) permite que os usuários resolvam problemas computacionais sem conhecimento ligado às linguagens de programação [Millner and Baafi 2011]. Sua interface se assemelha à do Ardublock pelo fato de sua programação ser realizada por meio de blocos sequenciais. Esta ferramenta permite programar o Arduino e é também compatível com outras plataformas de hardware.

### **2.4. Physical Etoys**

Physical Etoys é um ambiente de programação desenvolvido a partir da linguagem Small-talk. Possui interface gráfica onde os componentes reais físicos (sensores, fios, motores etc.) são metaforizados como objetos virtualizados prontos para manipulação. A interface de programação, diferentemente dos outros softwares já apresentados, e a lógica de programação, apesar de serem visuais, não são realizadas por meio de blocos, mas sim através de scripts, o que pode ocasionar confusão na realização de suas atividades. Atualmente a ferramenta é compatível com Windows e Linux e disponibilizada em vários idiomas, incluindo o português.

### **2.5. S4A**

O S4A (*Scratch for Arduino*) é uma ferramenta baseada no Scratch. A programação é feita por meio de blocos e há interação com a plataforma Arduino. A interface de programação é bastante intuitiva e de fácil manuseio, sendo capaz de detectar automaticamente placa(s) Arduino conectada(s) via cabo USB ao computador de desenvolvimento. Esta comunicação é dada pelo envio dos estados dos atuadores e a receptividade dos estados dos sensores, no intervalo de microssegundos, através de um protocolo cujas rotinas de tratamento devem se encontrar pré-instaladas na(s) placa(s) conectada(s).

### 3. Metodologia de análise dos softwares utilizados para a programação visual em contextos de RP

A Metodologia usada neste trabalho é dividida em duas partes. A primeira, discutida na subseção 3.1, consiste em analisar a técnica com base em atributos oferecidos pelos softwares analisados. A segunda, discutida na subseção 3.2, é composta por um protocolo de pesquisa que avalia a manipulação das interfaces das plataformas através de informações quantificáveis de métricas de usabilidade.

#### 3.1. Abordagem dos atributos dos softwares analisados

A análise dos softwares seguiu cinco critérios que os autores julgaram pertinentes para a presente pesquisa, enumerados de C1 a C5: C1 – O software é gratuito?; C2 – É de código aberto?; C3 – O aplicativo é compatível com a plataforma Arduino?; C4 – Sua linguagem é traduzida para o Português-Brasileiro (PT-BR)?; C5 – A ferramenta tem compatibilidade com quais Sistemas Operacionais (SOs)?. As questões empregadas, e as suas respostas, encontram-se descritas a seguir (questionário abaixo e Tabela 1, respectivamente).

Tais critérios foram elaborados de forma a ajudar futuros pesquisadores, professores e/ou administradores de instituições de ensino na escolha de plataformas que utilizem linguagens visuais de programação para a construção de projetos de robótica no ensino médio, de acordo com a possibilidade de: alcançar uma maior abrangência em seu uso (C1 e C5); redução de custos, com a utilização de plataformas de software livre (C1); multiplataforma (C5); comunicação com tecnologias de hardware que possam ser utilizadas em contextos educacionais, como o Arduino, por exemplo (C3), e; integração de estudantes que possuam certas necessidades linguísticas e/ou culturais para a manipulação de tais ambientes, como no caso da língua portuguesa, para estudantes brasileiros (C4).

**Tabela 1. Análise de softwares para a programação visual em contextos de RP**

<b>Critérios</b>	<b>Ardublock</b>	<b>Miniblock</b>	<b>ModKit Micro</b>	<b>Physical Etoys</b>	<b>S4A</b>
<b>C1</b>	Sim	Sim	Sim <sup>2</sup>	Sim	Sim
<b>C2</b>	Sim	Sim	Não	Sim	Sim
<b>C3</b>	Sim	Sim	Sim	Sim	Sim
<b>C4</b>	Sim	Sim	Não	Sim <sup>3</sup>	Sim
<b>C5<sup>4</sup></b>	W/L/M	W/L	W/M	W/L	W/L/M

Após a avaliação, observa-se que todos os softwares responderam satisfatoriamente ao critério C1, com uma ressalva para o ModKit Micro, que não disponibiliza todos os recursos em sua versão gratuita. Para o critério C2, observa-se que apenas o ModKit Micro é uma plataforma de código fechado. O Critério C3 indica que todos os softwares são compatíveis com a plataforma Arduino. Na sequência, o critério C4 mostrou que apenas o ModKit Micro não permite que a sua plataforma seja traduzida para o português (observa-se, entretanto, que apesar da interface do Physical Etoys poder ser

<sup>2</sup>O MM possui sua versão gratuita, mas exige uma licença para a utilização de alguns de seus recursos.

<sup>3</sup>Apesar do menu do PE pode ser traduzido para o português, seus exemplos e tutoriais estão em inglês.

<sup>4</sup>Legenda para Windows (W), Linux (L) e Mac (M).

traduzida para português, todos os seus exemplos e tutorias estão disponíveis somente em inglês). Por último, o critério C5 indicou que todas as plataformas possuem versões para o Windows, que apenas o ModKit Micro não é compatível com o Linux, e que somente o Ardublock e o S4A podem ser utilizadas no Windows, Linux e Mac.

### 3.2. Protocolo para avaliação da usabilidade das interfaces dos softwares analisados

A Tabela 1 foi capaz de indicar, segundo os critérios estabelecidos e analisados, que as ferramentas S4A e Ardublock foram as melhores classificadas, e que o Modkit Micro obteve os piores resultados. Entretanto, uma análise para a escolha de uma plataforma mais adequada, com tais dados, poderia induzir este estudo a resultados inconclusivos, uma vez que os critérios subjetivos de usabilidade devem ser analisados cuidadosamente nesta pesquisa, que tem como objetivo a utilização de programação visual em contextos de RP. Desta forma, torna-se necessário a continuação deste estudo através da avaliação das ferramentas em relação às suas facilidades de uso.

Para isto, uma revisão bibliográfica acerca da abordagem de usabilidade de sistemas interativos foi realizada [Nielsen and Landauer 1993, Nielsen and Mack 1994, Preece et al. 2002, Prates and Barbosa 2003], e requisitos como satisfação do usuário, facilidade de uso/aprendizagem e qualidade da interface gráfica foram levados em consideração durante a elaboração de mais dois questionamentos e respectivos desdobramentos da segunda etapa da metodologia desta pesquisa:

- A interface gráfica da plataforma é inteligível (ou seja, de fácil entendimento)?
- Sua manipulação é descomplicada?

Para a análise destes critérios a avaliação torna-se mais subjetiva, não podendo assim ser julgada com apenas uma resposta do tipo SIM/NÃO, pois existe um grau de relatividade em cada um dos quesitos que pode distinguir opiniões e induzir a investigação a resultados incoerentes. Para tanto, realizou-se uma pesquisa de opinião, na forma de entrevista, com 5 indivíduos, de forma individual e com base em um questionário (que será descrito posteriormente).

O estudo da usabilidade de um software baseado em pesquisa com apenas 5 usuários parece pouco demonstrativo, mas para [Nielsen 2000], testes com apenas um usuário já são capazes de detectar cerca de um terço (33.3%) de possíveis problemas que a interface de um sistema possa vir a apresentar, e quando este número passa para cinco pessoas, tal valor chega a 85% (sendo, na maioria das vezes, desnecessária a utilização de mais usuários para continuar os testes devido aos custos e tempo associados).

Neste trabalho, os autores (estudantes e professores universitários da área de Ciência da Computação) participaram como avaliadores das enquetes sobre as plataformas em análise. A justificativa para isso se dá por dois motivos. Primeiro, por estes colaboradores possuírem as características técnicas desejadas dos futuros usuários que utilizarão os resultados desta proposta. Segundo, pois existem alguns trabalhos na área de Interface Homem-Máquina que apontam a importância da avaliação da interação de plataformas computacionais por avaliadores especialistas nas áreas de usabilidade, desenvolvimento de software, ou ainda, de certos domínios de interesse dos programas sob análise [Nielsen and Mack 1994].

As respostas da segunda etapa foram analisadas e classificadas segundo a Escala Likert, que permite mensurar as atitudes e conhecer o grau de conformidade que um

usuário entrevistado tem com a afirmação proposta a ele por meio da escolha de um item entre cinco [Likert 1932], como apresentado a seguir:

- (1) Discordo totalmente
- (2) Discordo
- (3) Indiferente
- (4) De acordo
- (5) Totalmente de acordo

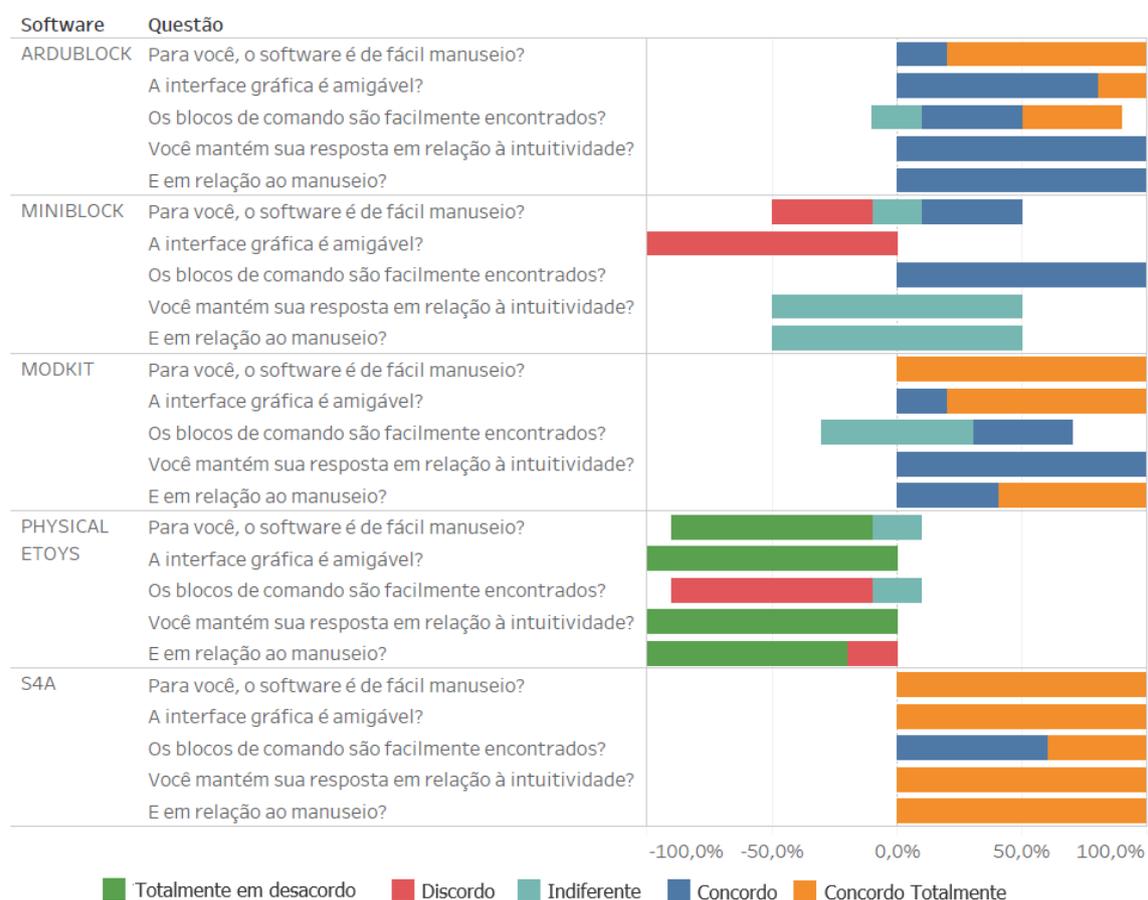
Para realização da pesquisa entre os usuários, utiliza-se um protocolo elaborado e testado em outros dois experimentos realizados pelos autores deste trabalho [Viana et al. 2017, Santiago et al. 2017]. O roteiro deve ser seguido por todos os entrevistados envolvidos de modo que um dos autores seja o responsável por aplicar o teste. O mesmo é aplicado de forma individual e igualitária, para cada uma das plataformas em análise, de acordo com os seguintes passos:

- Inicialmente, o responsável pelos testes apresenta a ferramenta ao usuário que irá testá-la, demonstrando as suas principais funcionalidades: criar projeto; localizar os principais blocos de comando do software; identificar a área de construção da modelagem do código, onde tais blocos serão depositados; e salvar projeto (se possível).
- Em seguida, o usuário é convidado a manusear o aplicativo a partir do seguinte roteiro: 1) Crie um novo projeto; 2) Encontre o bloco de comando que representa uma estrutura de repetição e o arraste para área de construção de código; 3) Crie uma variável e a insira ao seu bloco de comandos; 4) Associe esta variável ao pino digital 13 da estrutura microcontrolada da unidade de processamento do software (no modo ON); 5) Se possível, salve seu projeto. Após a finalização desta atividade, o responsável pelos testes recupera o roteiro entregue ao avaliador, afim de que o mesmo não o utilize como aparato para realização das experiências 1 e 2, descritas a seguir.
- O usuário recebe uma segunda folha contendo a descrição da primeira experiência e é convidado a implementar a seguinte lógica, em uma estrutura de *loop* infinito: Acenda um LED, espere um segundo, desligue-o, espere mais um segundo e o ligue novamente.
- Após a finalização deste experimento, o usuário é convidado a responder às seguintes questões, baseadas nos dois questionamentos realizados na segunda parte da metodologia desta pesquisa (“A interface gráfica da plataforma é inteligível (ou seja, de fácil entendimento)?” e “Sua manipulação é descomplicada?”) – as suas respostas foram quantificadas segundo a Escala Likert:
  - Para você, o software é de fácil manuseio?
  - A interface gráfica é amigável?
  - Os blocos de comando são facilmente encontrados?
- Em seguida, o usuário recebe uma terceira folha contendo a descrição da segunda experiência e é convidado a implementar a seguinte lógica, novamente em uma estrutura de *loop* infinito: Insira dois LEDs e faça com que os dois sejam acesos alternadamente, no intervalo de um segundo para cada acendimento.
- Solicita-se que o usuário responda a um novo questionário referente ao segundo experimento, cujas respostas foram quantificadas segundo a Escala Likert:

- Você mantém sua resposta em relação à intuitividade? (Se sim, deixe em branco ou repita a resposta anterior).
- E em relação ao manuseio? (Se sim, deixe em branco ou repita a resposta anterior).

#### 4. Discussão

Após finalizar a experimentação de usabilidade das plataformas, os resultados foram contabilizados e exibidos na Figura 1. Foram realizadas cinco perguntas às quais foram atribuídas notas em uma grandeza de 1 a 5, sendo a 1 (Totalmente em desacordo) a avaliação mais baixa e 5 (Concordo totalmente), a mais alta. Para a interpretação do gráfico é conveniente ressaltar que sua escala vai de -100% à 100%, onde os valores negativos demonstram as menores notas (1 - Totalmente em desacordo e 2 - Desacordo) e os positivos, as maiores (4 - Concordo e 5 - Concordo Totalmente).



**Figura 1. Respostas à experimentação de usabilidade baseadas na Escala Likert**

Levando-se em consideração a quantidade de entrevistados para um mesmo questionamento (5 usuários neste experimento), a proporção para cada possível resposta equivale a um total de 20%. Respostas que receberam uma avaliação 3 ('Indiferente', representada pela cor azul-clara na Figura 1) são posicionadas no centro de sua respectiva barra de exibição (0%), com uma distribuição equitativa deste valor em seus lados negativo e positivo. Assim, se uma pergunta recebeu uma única resposta Indiferente, esta será

representada com 10% à esquerda do centro da barra (lado negativo), e os outros 10%, a direita deste (lado positivo). Este raciocínio é mantido caso mais usuários sejam Indiferentes a um mesmo questionamento: para duas respostas Indiferentes neste experimento, tem-se 20% para o lado esquerdo e 20% para o lado direito; e assim sucessivamente.

Observando-se a Figura 1, pode-se perceber uma grande vantagem da ferramenta S4A sobre as demais plataformas analisadas: em quatro dos cinco questionamentos ela obteve 100% de notas 5 ('Concordo totalmente', representada pela cor laranja); e em apenas um questionamento as opiniões foram divididas, porém todas apresentando um caráter positivo (40% de notas 5 – 'Concordo totalmente', e 60% de notas 4 – 'Concordo', representada pela cor azul-escuro). Ainda na Figura 1 é possível observar que: as ferramentas Ardublock e ModKit Micro mostram uma certa equivalência nas impressões recebidas por seus avaliadores; e que os softwares MiniBloq e Physical Etoys foram os únicos a receber notas inferiores a 3 (para o Physical Etoys é válido salientar a grande quantidade de notas 1 recebida, mostrando o grande distanciamento desta plataforma no comparativo entre as ferramentas analisadas no presente trabalho).

As informações coletadas na segunda etapa da metodologia desta pesquisa servem para corroborar os resultados da primeira, que analisou somente critérios técnicos e classificou as ferramentas S4A e Ardublock como as duas mais interessantes das cinco analisadas (cf. Tabela 1). Além disto, esta segunda etapa fez uso de uma experimentação com critérios de usabilidade e serviu finalmente para identificar, de forma inequívoca, a superioridade da plataforma S4A em relação às demais analisadas, quando levada em consideração a problemática estudada neste trabalho: uso de ambientes de linguagem visual, em contextos de Robótica Pedagógica (RP), para o ensino de estudantes secundaristas na área de programação.

## 5. Considerações Finais

Este trabalho realizou uma pesquisa buscando selecionar uma ferramenta que possibilitasse o ensino de estudantes secundaristas na área de programação, através de linguagens visuais e utilizando-se o conceito de RP, de uma forma simples, agradável e eficaz. Pesquisou-se um aplicativo que possuísse características que tornassem mais fácil a sua disseminação entre os seus usuários, evitando incompatibilidades de sistemas operacionais, dificuldades em sua linguagem de programação, problemas de manipulação de suas interfaces gráficas, dentre outros critérios discutidos nas seções anteriores.

Após estabelecer um comparativo entre os softwares pré-selecionados pelos autores, a ferramenta S4A destacou-se em relação às demais de forma positiva. Este aplicativo mostrou-se bem mais preparado para ser utilizado em projetos futuros que venham a utilizar como proposta o tema abordado nesta pesquisa. Por conseguinte, o presente trabalho pode ser usado como referência para uma melhor implementação das técnicas de RP com o S4A, em turmas do ensino médio, auxiliando na inserção de linguagens de programação no currículo de escolas secundaristas.

Como outra vertente de pesquisa em trabalhos futuros, propõe-se outras possibilidades de análise dos softwares sob investigação utilizando o público alvo para realizar os devidos testes, de modo que seja feita uma avaliação da efetividade do aprendizado de conceitos da robótica pedagógica, além de ampliar a quantidade de avaliadores que possa afirmar a técnica usada no presente trabalho. Sugere-se também uma elaboração de per-

guntas subjetivas aos avaliadores, de maneira que as mesmas sirvam como uma análise particular e mais profunda das ferramentas analisadas.

## Referências

- Ardublock (2017). Ardublock – A Graphical Programming Language for Arduino. <http://blog.ardublock.com>. Acessado em 15.07.2017.
- Arduino (2017). Arduino. <https://www.arduino.cc>. Acessado em 15.07.2017.
- Aureliano, V. C. O. and Tedesco, P. C. d. A. R. (2012). Avaliando o uso do scratch como abordagem alternativa para o processo de ensino-aprendizagem de programação. In *XX Workshop sobre Educação em Computação*.
- Barr, V. and Stephenson, C. (2011). Bringing computational thinking to k-12: what is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1):48–54.
- Booth, T. and Stumpf, S. (2013). End-user experiences of visual and textual programming environments for arduino. In *International Symposium on End User Development*, pages 25–39. Springer.
- Cardoso, R. and Antonello, S. (2015). Interdisciplinaridade, programação visual e robótica educacional: relato de experiência sobre o ensino inicial de programação. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 4, pages 1255–1262.
- De Miranda, L. C., Sampaio, F. F., and dos Santos Borges, J. A. (2007). Programefácil: Ambiente de programação visual para o kit de robótica educacional robofácil. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação)*, volume 1, pages 260–269.
- De Souza, I. M. L., da Silva Rodrigues, R., and Andrade, W. (2016). Introdução do pensamento computacional na formação docente para ensino de robótica educacional. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 5, pages 1265–1274.
- D’Abreu, J. V. V., Ramos, J. J., Mirisola, L. G., and Bernardi, N. (2013). Robótica educativa/pedagógica na era digital. In *II Congresso Internacional TIC e Educação*. Disponível em: Acesso em, volume 15.
- EduTec (2017). S4A – Scratch for Arduino. <http://s4a.cat>. Acessado em 15.07.2017.
- Evans, B. (2011). *Beginning Arduino Programming*. Apress.
- GIRA (2017). Physical Etoys. <http://tecnodacta.com.ar/gira/projects/physical-etoys>. Acessado em 15.07.2017.
- Gomes, A., Henriques, J., and Mendes, A. (2008). Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. *Educação, Formação & Tecnologias-ISSN 1646-933X*, 1(1):93–103.
- Gomes, G., Martinho, J., Bernardo, M., Matos, F., and Abrantes, P. (2012). Dificuldades na aprendizagem da programação no ensino profissional–perspetiva dos alunos. In *Anais II Congresso Internacional TIC e Educação, Lisboa, Portugal*, pages 438–448.

- Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):5–55.
- Millner, A. and Baafi, E. (2011). Modkit: blending and extending approachable platforms for creating computer programs and interactive objects. In *Proceedings of the 10th International Conference on Interaction Design and Children*, pages 250–253. ACM.
- Minibloq (2017). Minibloq. <http://blog.minibloq.org>. Acessado em 15.07.2017.
- ModKitMicro (2017). ModKit Micro. <http://www.modkit.com/micro>. Acessado em 15.07.2017.
- Nielsen, J. (2000). Why you only need to test with 5 users. <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users>. Acessado em 15.07.2017.
- Nielsen, J. and Landauer, T. K. (1993). A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 206–213. ACM.
- Nielsen, J. and Mack, R. L. (1994). *Usability Inspection Methods*. John Wiley & Sons, New York, NY.
- Papavlasopoulou, S., Giannakos, M., and Jaccheri, L. (2017). Reviewing the affordances of tangible programming languages: Implications for design and practice. In *Global Engineering Education Conference (EDUCON)*, pages 1811–1816. IEEE.
- Prates, R. O. and Barbosa, S. D. J. (2003). Avaliação de interfaces de usuário - conceitos e métodos.
- Preece, J., Rogers, Y., and Sharp, H. (2002). *Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, Inc., New York, NY, USA.
- Queiroz, R., Sampaio, F. F., and dos Santos, M. P. (2016). Duinoblocks4kids: Ensinando conceitos básicos de programação a crianças do ensino fundamental i por meio da robótica educacional. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 5, pages 1169–1178.
- Santiago, L., Jr., J. A., Viana, H., Louro, A., Abijaude, J., and Sobreira, P. (2017). Análise de softwares de virtualização de objetos eletrônicos para utilização em projetos de ensino que utilizem o conceito de internet das coisas: Protoboard virtual e plataforma arduino. In *ERBASE 2017 – WEIBASE*.
- Trentin, M. A. S., Teixeira, A. C., Da Rosa, C. T. W., and Da Rosa, Á. B. (2013). Robótica como recurso no ensino de ciências. In *Proceedings of International Conference on Engineering and Computer Education*, volume 8, pages 231–235.
- Viana, H., Sobreira, P., Abijaude, J., Louro, A., Jr., J. A., and Santiago, L. (2017). Redes sociais na educação - revisão de softwares de redes sociais como ferramentas educacionais. In *ERBASE 2017 – WEIBASE*.
- Zanetti, H. and Oliveira, C. (2015). Práticas de ensino de programação de computadores com robótica pedagógica e aplicação de pensamento computacional. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 4, pages 1236–1245.