

Uma Análise Comparativa entre Ambientes de Programação em Blocos para a Interação com o Arduino

Luiz Kevin de Souza Rios¹, João Paulo F. Lima¹, Almir de Oliveira Costa Junior¹, Elloá B. Guedes¹

Universidade do Estado do Amazonas¹

e-mail: lkdsr.lic16@uea.edu.br, jpfls.lic17@uea.edu.br, adjunior@uea.edu.br, ebgcosta@uea.edu.br

Resumo. Quais aspectos são relevantes e precisam ser considerados na escolha de um ambiente de programação em bloco para interagir com o Arduino? Em essência, estes ambientes dispõem dos mesmos recursos que aqueles voltados apenas para o ensino e aprendizagem de lógica de programação. Porém, além de possuírem uma notação visual clara e intuitiva, devem oferecer blocos dedicados a realizar a interação com as portas do Arduino. Diante disso, este trabalho apresenta e analisa comparativamente as características de três ambientes de programação baseados no Scratch, considerando diferentes aspectos nesta análise, tais como: interface, integração com o Arduino, firmware, integração dos blocos, dentre outros. Além disso, outra contribuição deste artigo é fornecer informações para o processo de tomada de decisão na escolha dos ambientes de programação a serem adotados em contextos reais de atividades em sala de aula.

Palavras Chave: Scratch 2.0, Scratchduino, Kittenblock, Tecnologias Educacionais, Robótica Educacional.

Introdução

Os ambientes de programação em blocos têm atraído o interesse de professores e pesquisadores por se caracterizar como uma ferramenta introdutória a uma linguagem de programação, com grande potencial para o desenvolvimento cognitivo de estudantes de diversos níveis, especialmente as crianças (Alves et al., 2014). Em particular, empoderam este público alvo como agente participativo e autor dessas novas tecnologias, não apenas atuando como meros consumidores, corroborando também para a aquisição de habilidades transversais e multidisciplinares (Silva et al., 2017) e (Porvir, 2019).

Ainda nesta perspectiva e considerando os benefícios e práticas preconizados pela Robótica Educacional, muitos ambientes de programação têm incorporado a integração com plataformas de criação de dispositivos eletrônicos, tais como o Arduino. A ideia é diminuir a distância entre programação e a abstração dos conceitos demandados por estar utilizando tais dispositivos como meio para visualização concreta de conceitos, especialmente pelos mais iniciantes. Juntamente com a filosofia "*Do it yourself*", este ferramental permite a construção, experimentação, personalização e livre criação (Rodrigues et al., 2016).

Levando em conta estas características e o fato de tais ambientes ainda serem emergentes, é essencial conduzir uma análise das ferramentas atualmente disponíveis e mais disseminadas, com vistas a elencar e evidenciar suas características e delimitar suas limitações, favorecendo a identificação dos cenários mais propícios para sua utilização. Em particular, tomou-se como referência a plataforma Arduino, por ser amplamente adotada, aberta e de baixo-custo. Considerando este contexto, o presente trabalho se propõe a analisar comparativamente três destes ambientes quanto aos recursos que possuem. Nesta análise foram levados em considerações alguns fatores, tais como: sistemas operacionais, aspectos da integração com o Arduino, usabilidade, disponibilidade de material pedagógico, dentre outros.

Para apresentar os resultados obtidos, este artigo está organizado como segue. Os ambientes de

programação em blocos e seus conceitos são apresentados na Seção 2. A metodologia de comparação e os critérios analisados são apresentados na Seção 3. Em seguida, os ambientes considerados são enumerados e descritos na Seção 4. Segue-se então a análise comparativa, na Seção 5, na qual os ambientes serão contrastados. Por fim, são apresentadas as considerações finais e sugestões de trabalhos futuros.

Ensino de Programação em Bloco

A programação em si sempre foi vista como um obstáculo para alunos que não possuem facilidade em programar e também para aqueles que estão tendo um primeiro contato com a mesma, principalmente a programação textual em linhas de código (Souza e França, 2013). Diante disso, na perspectiva de diminuir estas dificuldades, foram criados os ambientes de programação visuais, sendo um de seus ramos os ambientes de programação em blocos.

A ideia destes ambientes de programação em blocos consiste em tornar a programação, naturalmente atrelada à conceitos da Matemática e da Lógica, uma experiência mais intuitiva, de natureza essencialmente visual. Os comandos e valores comportam-se de maneira análoga à blocos de encaixe, fazendo com que o ato de programar torne-se essencialmente menos complexo, de mais alto nível, aproximando o público infantil e um público que outrora não tinha experiência com programação, tais como os professores da educação básica (Silva et al. 2017).

O pontapé para a proposição de ambientes desta natureza iniciou-se em 1967, quando Feurzeig e Papert propuseram a linguagem de programação LOGO, com vistas a apoiar o ensino regular de programação para crianças (Papert, 1994). No LOGO, as crianças eram estimuladas a controlar os movimentos de uma tartaruga, que desenhava traços na tela de acordo com comandos pré-estabelecidos, conforme exemplificado na Figura 1. Neste cenário, a criança era estimulada a exercitar o pensamento computacional por meio da interação, podendo realizar diversas tentativas e aprender com seus próprios erros até alcançar um objetivo desejado.

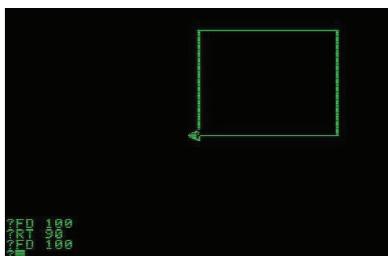


Figura 1. Ambiente de Programação LOGO. Fonte: (Turtle Academy, 2019)

Com o passar dos anos e um maior acesso e disponibilidade da tecnologia, esta ideia seminal foi evoluindo e outros ambientes de programação em blocos foram propostos, incorporando elementos gráficos mais complexos e atrativos. Nesta perspectiva, citam-se o DuinoBlock for Kids, o Google Blockly, ilustrado na Figura 2, e, mais predominantemente, o Scratch, que está entre umas das 20 linguagens de programação mais populares no mundo.

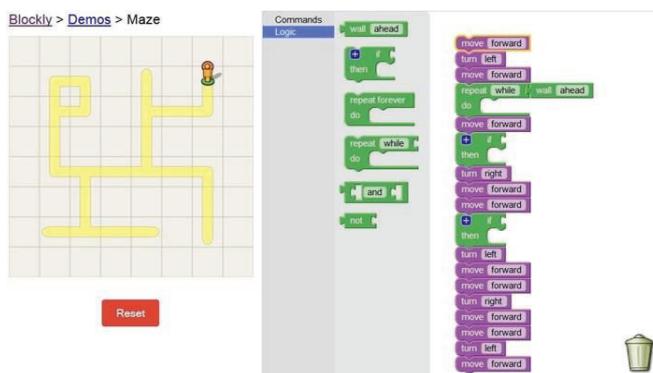


Figura 2. Ambiente Google Blockly. Fonte: (Blockly, 2019)

Tendo em vista este contexto e o perfil do público-alvo, é perceptível que a programação em blocos pode ser uma ferramenta que venha a colaborar no desenvolvimento de habilidades e competências que são necessárias em muitas disciplinas curriculares, tais como Matemática e Física, além de exercitar o pensamento computacional (Garlet et al., 2016).

Metodologia e Critérios Analisados

Com vistas a conduzir uma análise comparativa, inicialmente foi feito um levantamento dos ambientes de programação em blocos que dispunham de integração com a plataforma Arduino. Para este levantamento, foram consultados os manuais disponíveis, páginas de divulgação, artigos, livros e fóruns de discussão.

Após o procedimento de identificação dos ambientes, seguiu-se então com a enumeração de características a serem consideradas. Primeiramente, considerou-se aspectos que favoreçam a sua utilização, tais como os sistemas operacionais em que o ambiente está disponível e também a portabilidade, isto é, a capacidade de ser executado em diferentes plataformas. Em ambos os critérios, entende-se que quanto maior a quantidade de opções disponíveis, maior é o público-alvo capaz de abranger.

Considerando a utilização por crianças, o critério de idiomas que o ambiente dispõe também foi elencado, pois este pode ser um limitador para o público-alvo a que se destina, cuja habilidade com línguas estrangeiras ainda está em desenvolvimento. Um outro critério que pode ser crucial para a ampla adoção do ambiente também foi investigado: os custos para aquisição e utilização, quando haviam.

Em se tratando da existência de material pedagógico, um critério especialmente útil para uma adoção de mais curto prazo em ambiente escolar, por exemplo, considerou a produção de tais artefatos pelos proponentes do ambiente e/ou pela comunidade usuária, aspecto relevante para propiciar a realização de atividades junto aos estudantes.

No tocante às questões mais técnicas, avaliou-se o grau de integração com a plataforma Arduino, a existência de blocos específicos e livremente disponíveis para projetos com esta plataforma e ainda aspectos de usabilidade, relevantes para um bom aproveitamento no uso destes ambientes. Estes critérios foram ponderados com o uso da Escala Likert de 5 níveis, variando de Péssimo a Excelente ou de Muito Ruim a Muito Bom, contemplando também argumentos que justifiquem a classificação estabelecida.

As seções a seguir apresentam os resultados obtidos decorrentes do cumprimento desta metodologia.

Ambientes de Programação em Blocos

Esta seção apresenta os três ambientes de programação em blocos com foco na interação com a plataforma Arduino identificados conforme procedimento metodológico descrito. Estes ambientes, a citar: S4A, Scratchduino e KittenBlock, são baseados na linguagem de programação visual Scratch e são detalhados a seguir.

Scratch For Arduino - S4A

Trata-se de um ambiente baseado no Scratch que permite programar de forma simples a plataforma de prototipagem aberta Arduino. Com uma interface gráfica similar ao do Scratch 1.4, o ambiente tem como objetivo principal, atrair pessoas para o mundo da programação por meio da utilização de blocos, sem que exista a necessidade de digitar linhas de códigos para interagir com o Arduino. Na Figura 3 é apresentada a interface principal do ambiente.

Uma característica bem peculiar ao S4A, é a disponibilidade de um monitor sobre o palco do ambiente onde são disponibilizadas informações importantes sobre o status de comunicação entre o ambiente de programação S4A e a placa Arduino como mostra a Figura 3.

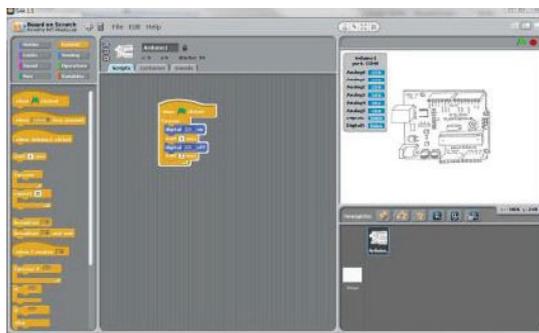


Figura 3. Ambiente de programação do S4A. Fonte: (S4A, 2019)

Desevolvido pelo Citilab do Edutec Research Group, trata-se de um ambiente baseado no Scratch que permite trabalhar com programação em blocos.

Ele apresenta recursos e ferramentas para a integração com o Arduino, inclusive possui um *Sprite* dedicado a dar suporte à interação, já adicionado automaticamente ao palco do Software (S4A, 2019).

Atualmente, o ambiente está na versão 1.6 e é distribuído gratuitamente com suporte ao português do Brasil, sendo compatível com os principais sistemas operacionais disponíveis atualmente: Windows, Linux, Mac e Raspbian (Debian para RaspberryPi). No site do S4A, são disponibilizados alguns modelos de projetos e um ambiente com respostas a perguntas frequentes. É importante ressaltar que os programas desenvolvidos em S4A não são suportados em outros ambientes (S4A, 2019).

Uma das características particulares do S4A, é a obrigatoriedade de conectar motores, sensores e atuadores em portas específicas do Arduino. Isso ocorre porque o ambiente define blocos específicos para interagir com 6 entradas analógicas (pinos analógicos), 2 entradas digitais (pinos digitais 2 e 3), 3 saídas analógicas (pinos digitais 5, 6 e 9), 3 saídas digitais (pinos 10, 11 e 13) e 4 saídas especiais para conectar os servomotores de rotação contínua Parallax (pinos digitais 4, 7, 8 e 12) (S4A, 2019). Diante disso, a adoção deste ambiente impõe requisitos restritivos no desenvolvimento de projetos de desenvolvimento eletrônicos na plataforma Arduino.

Scratchduino

Em sua essência, o Scratchduino não se trata de um ambiente de programação derivado do Scratch, mas sim um servidor de comunicação multiplataforma que permite realizar a integração da versão nativa do Scratch 2.0 com projetos de robôs e dispositivos de computação física baseados em Arduino (Computação na Escola, 2019). O servidor Scratchduino é desenvolvido pela Computação na Escola, que é uma iniciativa do INCoD/INE/UFSC. Na Figura 4a é apresentado o modelo de interação do Scratchduino.

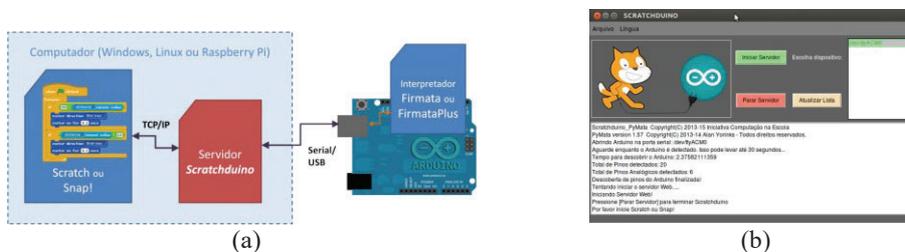


Figura 4. Esquema de funcionamento do Scratchduino e interface do servidor de comunicação. Fonte: (Computação na Escola, 2019)

A interface do servidor Scratchduino é bastante simples e leve. Conforme ilustrado na Figura 4b, a interface dispõe de três botões básicos que permitem realizar as seguintes tarefas: Iniciar o Servidor, Parar o Servidor e Atualizar Lista. Esta última tarefa, em especial, serve para identificar as portas de comunicação que estão com controladores Arduino conectados a ela.

O servidor Scratchduino, é compatível com os sistemas operacionais: Windows (XP e superiores), Linux e Raspbian (Raspberry Pi) e é disponibilizado de forma gratuita. No site da iniciativa, ainda é possível encontrar os manuais de instalação nos diferentes sistemas operacionais e algumas instruções para eventuais erros possíveis de ocorrer ao instalar o software. Além disso, estão disponíveis alguns exemplos de projetos desenvolvidos de forma integrada com materiais alternativos para construção de protótipos robóticos e materiais de cunho pedagógico para o professor, contendo instruções de como preparar as atividades práticas. Atualmente, o servidor possui suporte a língua portuguesa e inglesa (Computação na Escola, 2019).

Para criar um novo projeto neste ambiente, é necessário sempre utilizar *templates* de projeto fornecidos pela iniciativa. Isto é necessário para que os blocos específicos de interação com o Arduino fiquem disponíveis para serem utilizados (Computação na Escola, 2019). Uma das particularidades desses blocos de interação, é a possibilidade de realizar a personalização do número da porta que poderá ser controlada no Arduino, conforme Figura 5. Dessa forma, fica a critério do usuário personalizar as portas da melhor maneira para atender suas necessidades.



Figura 5. Blocos especiais para o Scratch interagir com o Arduino via Scratchduino. Fonte: (Computação na Escola, 2019)

KittenBlock

O KittenBlock é um ambiente de programação que permite a interação com placas de programação por meio de uma linguagem de blocos de encaixe. Suas funcionalidades e interface gráfica são baseadas em uma prévia do Scratch 3.0 (KittenBlock, 2019). A ideia por trás da iniciativa, é permitir que pessoas iniciantes possam aprender a programar a partir do Scratch até se sentirem mais seguras em utilizar linha de código por meio do *Python* e a linguagem *C* para criar programas que possam interagir com as placas microcontroladoras.

O ambiente é desenvolvido e distribuído de forma gratuita pela Shenzhen Kitten Technology Co., LTD que foi fundada em 2016 na China. Além deste ambiente, a iniciativa também desenvolve um modelo de hardware denominado *KittenBot Robot* para os *makers* interessados em desenvolver projetos de robótica. Em sua interface, o ambiente já apresenta opções no menu dedicadas aos blocos de interação com as placas microcontroladoras Arduino e *KittenBot*, além destas placas o ambiente ainda permite realizar interações com a *Microbit* (KittenBlock, 2019).

O KittenBlock é disponibilizado no site do *KittenBot*, e atualmente está na versão 1.8.3 com suporte aos idiomas inglês, chinês e coreano, com interface gráfica de interação com usuário conforme ilustrada na Figura 6a, o ambiente é compatível com os sistemas operacionais Windows e Mac. No site da iniciativa, pode-se encontrar tutoriais com exemplos práticos e materiais de suporte ao usuário. Porém, como trata-se de uma iniciativa recente, ainda não existem muitos exemplos disponíveis (KittenBlock, 2019). Na Figura 6b são ilustrados os blocos de interação com as placas controladoras do Arduino.

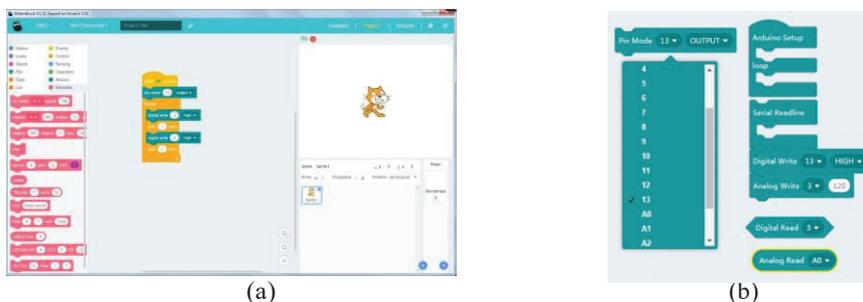


Figura 6. Ambiente de programação do KittenBlock. Fonte: (KittenBlock, 2019)

Este ambiente possui algumas características particularmente atrativas. A primeira delas consiste de funcionalidade para tradução automática da programação feita em blocos para linha de código suportada pelo Arduino. De certo modo, isto acaba permitindo que as instruções construídas dentro do ambiente possam ser executadas na placa Arduino de modo *Offline*, ou seja, sem a necessidade de estar com a placa conectada via cabo USB após a transferência da programação. Outra característica importante do ambiente, é a possibilidade de realizar a personalização do número da porta no bloco que permitirá a interação com o Arduino. Além disso, o ambiente já apresenta blocos dedicados a interagir com LEDs, LCDs, sensores, motores e servomotores conectados ao Arduino, o que facilita a identificação dos blocos utilizados em cada projeto.

Análise Comparativa

Conforme apresentado, existem diferentes ambientes de programação em blocos integráveis à plataforma Arduino com características bem particulares. A escolha de um determinado ambiente para adoção em certo contexto precisa levar em conta estas diferenças, visando identificar aquele que

melhor atende às necessidades. Desta maneira, considerando os critérios estabelecidos, fez-se uma avaliação dos ambientes identificados, sintetizando suas características de maneira mais objetiva, conforme apresentado na Tabela 1.

	S4A (<i>Scratch 1.0</i>)	Scratchduino (<i>Scratch 2.0</i>)	KittenBlock (<i>Scratch 3.0</i>)
Sistemas Operacionais	Windows, Linux, Mac e Raspbian	Windows, Linux e Raspbian	Windows e Mac
Portabilidade	Boa	Boa	Excelente
Idiomas Suportados	50	74	3
Custo	Gratuito	Gratuito	Gratuito
Material Pedagógico	Muito Bom	Muito Bom	Muito Ruim
Grau de Integração com o Arduino	Excelente	Boa	Excelente
Integração dos Blocos	Excelente	Regular	Excelente
Usabilidade	Excelente	Excelente	Excelente

Tabela 1. Quadro comparativos com os diferentes aspectos considerados em cada ambiente.

No que se refere a quantidade de sistemas operacionais suportados, destacam-se o S4A que pode ser instalado nos sistemas operacionais Windows, Linux, Mac e Raspbian (Debian para RaspberryPi) e o Scratchduino que pode ser instalado no Windows (XP e superiores), Linux e Raspbian (Raspberry Pi). Como se trata de um ambiente recente, o KittenBlock ficou abaixo das expectativas, tendo em vista que o mesmo ainda não é suportado nos sistemas operacionais mais conhecidos, como é o caso do Linux, que é gratuito.

Em se tratando da portabilidade, todos os ambientes avaliados podem ser instalados em computadores com os respectivos sistemas operacionais. Com relação às animações/programações construídas por meio dos ambientes, destaca-se o KittenBlock, que possibilita a transcrição dos códigos em blocos para outras linguagens, como as suportadas nativamente pelo Arduino e o Python. Além disso, permite que os códigos criados, sejam transferidos para outros robôs independente do modelo de placa Arduino. No S4A, o número de modelos de Arduino suportados é limitado, além da impossibilidade de utilizar o código em outras plataformas. O Scratchduino, por sua vez, também é compatível com uma grande quantidade de modelos de Arduino, porém, acaba não disponibilizando uma ferramenta para que os programas possam ser executados em outras plataformas.

No item idiomas, os ambientes Scratchduino e S4A destacam-se por disponibilizar uma quantidade relativamente grande de opções de idiomas, entre elas o português do Brasil. Por outro lado, o KittenBlock ficou com uma classificação abaixo das expectativas, por disponibilizar apenas três opções de idiomas ao usuário: inglês, chinês e coreano.

Em termos de custo, todos os ambientes de programação analisados são disponibilizados de forma gratuita pelos seus desenvolvedores.

No caso do KittenBlock, em especial, os proponentes disponibilizam uma loja online para aquisição de diversos modelos de hardware compatíveis com este ambiente. Isto pode ser uma conveniência em comparação com as demais alternativas no tocante ao desenvolvimento dos componentes de hardware.

Com relação ao material de uso pedagógico, ou seja, aqueles que oferecem algum tipo de material de instrução para aplicação em contextos educacionais, o S4A recebeu uma classificação “Muito bom”. Nesta análise, pôde-se verificar que o ambiente possui muitos manuais disponíveis na internet que apresentam exemplos concretos de programações que podem ser criadas. De certo modo, a grande quantidade de material disponível se deve ao fato de o S4A já estar sendo utilizado há mais tempo que os demais, o que sugere a existência de uma comunidade maior de pessoas utilizando, recriando e compartilhando os seus projetos nesta plataforma.

Em se tratando da integração do ambiente com o Arduino, ou seja, a possibilidade de realizar a comunicação entre o ambiente de programação e hardware, os ambientes KittenBlock e S4A puderam ser classificados como sendo excelentes. Isto se deve ao fato de ambos não demandarem software intermediário para que a comunicação possa ser estabelecida entre ambas as partes. Como mencionado anteriormente, o Scratchduino trata-se de um servidor de comunicação entre ambiente Scratch 2.0 e a placa Arduino, e por este motivo, acaba tornando o processo de integração um pouco mais burocrático, o que pode ensejar dificuldades ao público iniciante.

Em termos de integração dos blocos, ou seja, a disponibilização de blocos personalizáveis e dedicados à interação com as portas do controlador Arduino, o KittenBlock recebeu uma classificação excelente, tendo em vista que os botões já estão integrados de forma nativa na interface do ambiente. No S4A, os blocos também já estão disponíveis de forma nativa, porém o usuário acaba sendo forçado a utilizar portas específicas para acionar determinados recursos. Por outro lado, o Scratchduino até permite que os números das portas sejam editados nos blocos, porém os blocos não estão disponíveis de forma nativa no ambiente de programação.

À usabilidade, isto é, a facilidade de utilizar a linguagem de programação gráfica, também foi um aspecto a ser analisado. Neste item, todos os ambientes foram classificados com “Excelente” tendo em vista que ambos se utilizam de recursos e estratégias semelhantes. O destaque fica para o KittenBlock, pois apresenta blocos de programação com descrições mais detalhadas e intuitivas sobre as suas funções e interações com as portas do Arduino. A classificação positiva neste aspecto também decorre do fato de todos se basearem no ambiente Scratch, que possui uma interface intuitiva e amigável, projetada para o público-alvo infantil e iniciante em programação.

Considerações Finais

Uma das preocupações ao adotar ambientes de programação em blocos para interagir com o Arduino, diz respeito a qual ambiente deverá ser utilizado, pois há diferentes opções disponíveis com diferentes características. Na tentativa de auxiliar neste processo de escolha, o artigo em questão apresentou e caracterizou três ambientes de programação em blocos que permitem a interação com o Arduino. Após esta apresentação, foi feita uma análise comparativa considerando oito aspectos diferentes, tais como, a integração com o Arduino, o material pedagógico, dentre outros.

Após esta análise, foi verificado que os ambientes se assemelham quanto a alguns aspectos, já que utilizam como base o Scratch, mas que se diferenciam em termos de outros. Diante disso, não houve o intuito de elencar um ambiente como sendo superior aos demais, mas sim elencar estas características com vistas a uma melhor tomada de decisão a respeito de qual deles adotar em determinado contexto. Por exemplo, o S4A e Scratchduino se mostram bastante interessantes no que diz respeito a quantidade de material disponível na internet para auxiliar iniciantes no mundo da programação. Já o KittenBlock pode ser mais interessante para aqueles que estão um passo à frente e tem interesse em introduzir uma linguagem de programação textual.

Como sugestões de trabalhos futuros, almeja-se uma expansão do número de ambientes de programação em blocos considerados na análise comparativa. Além disso, realizar análises comparativas de ambientes que não sejam baseadas em alguma versão do Scratch e Ambientes que sejam executados em Nuvem.

Agradecimentos

A autora E.B.G. agradece o apoio financeiro e material provido pela Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM) por meio do PPP 04/2017.

Referências

ALVES, Rafael Machado; SAMPAIO, Fábio Ferrentini. Duinoblocks: Desenho e implementação de um ambiente de programação visual para robótica educacional baseado no hardware arduino. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2014. p. 11.

Computação na Escola (2019). Iniciativa Computação na Escola. <http://www.computacaonaescola.ufsc.br/>. Acessado em 29 de março de 2019.

GARLET, D.; BIGOLIN, N. M.; SILVEIRA, S. R. Uma proposta para o ensino de programação de computadores na educação básica. **Departamento de Tecnologia da Informação, Universidade Federal de Santa Maria, RS**, 2016.

KittenBlock (2019). Kitten Block. <http://www.kittenbot.cc/kittenblock/>. Acessado em 29 de março de 2019.

Por Vir (2019). Competências para o Século XXI. <http://porvir.org/conheca-competencias-para-seculo-21/>. Acessado em 29 de março de 2019.

S4A (2019). Scratch for Arduino. <http://s4a.cat/>. Acessado em 29 de março de 2019.

BATISTA, Esteic Janaina Santos et al. Poredu: um ambiente de programação em blocos. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2017. p. 144.

DE SOUZA, Márcia Valéria Rocha; FRANÇA, A. César C. Um estudo sobre as dificuldades no processo de aprendizagem de programação no Curso de Análise e Desenvolvimento de Sistemas na FAFICA–Faculdade de Filosofia, Ciências e Letras de Caruaru-PE. **Revista da Escola Regional de Informática**, v. 2, n. 2, p. 19-27, 2013.