

BIPES: PROGRAMAÇÃO DE DISPOSITIVOS ELETRÔNICOS EM BLOCOS PARA ATIVIDADES EDUCACIONAIS MÃO NA MASSA

BIPES: PROGRAMMING ELECTRONIC DEVICES IN BLOCKS FOR HANDS-ON EDUCATIONAL ACTIVITIES

Jorge André Gastmaier Marques^{1*} 

Wesley Flávio Gueta¹ 

Tatiana de Figueiredo Pereira Alves Taveira Pazelli¹ 

Rafael Vidal Aroca¹ 

RESUMO: O desenvolvimento de habilidades de programação de computadores é relevante em diversos contextos educacionais. As atividades práticas *maker*, de robótica e solução de problemas por projetos trazem resultados importantes para alunos que participam desses tipos de atividade. Entretanto, mesmo com tecnologias acessíveis, como o Arduino e o ESP8266, os custos e a dificuldade de instalação podem restringir o acesso a esses recursos educacionais. Nesse sentido, o Projeto *Block Based Integrated Platform for Embedded Systems* (BIPES) foi concebido no início do ano de 2020 com o objetivo de facilitar essas atividades, oferecendo recursos como programação por blocos, suporte a vários dispositivos embarcados e a flexibilidade de dispensar a instalação e configuração de qualquer *software* adicional. Este artigo descreve os aspectos operacionais do Projeto BIPES e apresenta alguns resultados do seu uso em oficinas e outras atividades. Atualmente, o projeto conta com mais de 10 mil usuários.

Palavras-chave: Programação por blocos. BIPES. Atividades *maker*.

ABSTRACT: The development of computer programming skills is relevant in various educational and curricular contexts. In addition, hands-on robotics, maker activities and project-based problem solving have provided relevant results for students who attend these activities. However, even with low-cost technologies and platforms such as Arduino and ESP8266, installation challenges and costs can often prevent access to these educational resources. In this sense, the Block Based Integrated Platform for Embedded Systems (BIPES) project was conceived at the beginning of 2020 with the objective of facilitating these activities, offering resources such as block programming, support for multiple embedded devices, and flexibility of not requiring the installation or configuration of any additional software. This article describes the operational aspects of the BIPES project and presents some aspects of its use in workshops and other activities. Currently, the project has more than 10,000 users.

Keywords: Block based programming. BIPES. Maker activities.

1. Universidade Federal de São Carlos – São Carlos (SP), Brasil.

*Autor correspondente: jorge.marques@estudante.ufscar.br

Número temático organizado por: Vânia Almeida Neris  e Alessandra Arce Hai 

Introdução

Cursos regulares do ensino fundamental e médio vêm passando por transformações, especialmente o enfoque no pensamento computacional, que é o conjunto de habilidades comumente utilizadas na criação de programas computacionais como uma metodologia para resolver problemas específicos nas mais diversas áreas (BUNDY, 2007).

O pensamento computacional é composto de quatro pilares: decomposição, reconhecimento de padrões, abstração e algoritmos. Este último é responsável pela solução e execução de um problema (BRACKMANN, 2017), porém o aprendizado é um processo difícil, sendo necessários alto nível de abstração e em geral compreensão de linguagem com sintaxe complexa (GOMES *et al.*, 2008).

A robótica pedagógica consiste em um artefato educacional de grande potencial para promover um ambiente favorável ao aprendizado das habilidades do pensamento computacional (ZANETTI; OLIVEIRA, 2015). Os estudantes são mais motivados a resolver problemas com a programação de dispositivos reais, porque não só interagem com o ambiente, controlando robôs, coletando dados de sensores e testando algoritmos, como também têm a oportunidade de discutir os problemas que surgem com os colegas e professores em busca de soluções.

Além disso, tais atividades capacitam os estudantes, permitindo que eles preparem projetos avançados para feiras, mostras, competições e olimpíadas científicas, pavimentando um caminho de conquistas pessoais e profissionais para estudantes e professores (MIRANDA; SUANNO, 2012).

No caso específico da robótica educacional, trabalhos recentes demonstram que atividades práticas de robótica educacional estimulam o desenvolvimento de várias competências, além das habilidades técnicas óbvias, incluindo trabalho em equipe, raciocínio lógico, resiliência, pensamento crítico e criatividade (HUANG; HUANG, 2011; CHUNG; CARTWRIGHT; CHUNG, 2014; EGUCHI, 2014; CUKURBASİ; KIYICI, 2018; HUANG; HU; WU, 2018; SOUZA *et al.*, 2018; SCHINA; USART; ESTEVE-GONZALEZ, 2019; UNESCO, 2020).

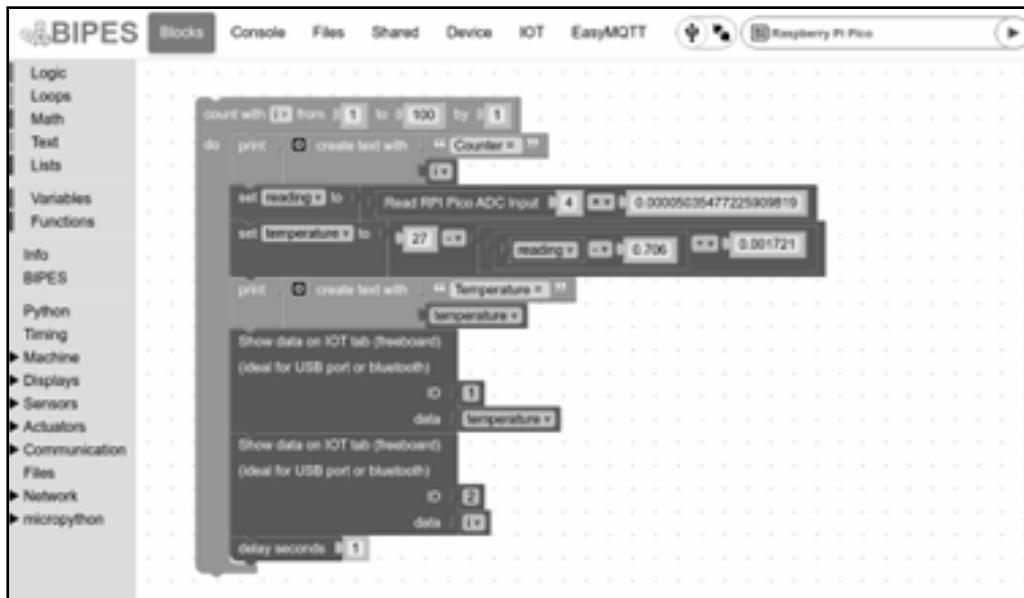
Nesse contexto, diversas ferramentas estão bem estabelecidas, tais como os *kits* de robótica Lego, *kits* baseados em Arduino ou na placa micro:bit, entre outras. Em alguns casos, o custo e a dificuldade de uso podem restringir o acesso a essas ferramentas (GALVAN *et al.*, 2006; ALVES *et al.*, 2011; CHELLA, 2012). Por outro lado, existem materiais e componentes de baixo custo, como as placas ESP32 e ESP8266, que podem ser aplicadas em atividades educacionais, demandando, entretanto, a instalação de *software* específico, *plugins* e/ou a realização de configurações especiais no computador de desenvolvimento. Além disso, essas placas são tipicamente programadas em linguagem com sintaxe complexa, como C ou Rust, exigindo maior capacitação de professores e monitores.

O projeto brasileiro *Block Based Integrated Platform for Embedded Systems* (BIPES) foi concebido no início do ano de 2020 (SILVA JUNIOR *et al.*, 2020; AROCA *et al.*, 2021a; 2021b) com o objetivo de ampliar o acesso a essas ferramentas, apresentando algumas características especiais:

- Projeto aberto, gratuito e de código totalmente aberto (licença *general public license* – GPL);
- Baseado na linguagem de programação visual em blocos do Google, o Blockly;
- Suporte a diversas placas, incluindo Arduino, ESP8266, ESP32, STM32, micro:bit, Raspberry Pi Pico, placas com Linux, entre outras;
- Funciona *online* ou *offline*, sem a necessidade de conexão à internet;
- Não precisa ser instalado nem configurado. Basta ter os arquivos em uma pasta para acessar o BIPES, sem ser necessário instalar nenhum *software* ou *plugin*;
- Baseado em plataforma *web*, funcionando diretamente em navegadores *web* como Google Chrome;
- Acesso para programação e monitoramento de placas via USB, *serial*, *Wi-Fi* ou *bluetooth*;
- Possibilidade de gerenciar arquivos nas placas eletrônicas utilizadas;

- Suporte à internet das coisas usando MQTT e opção de construção de painéis de visualização de dados;
- Compartilhamento de código por *link*;
- Suporte a vários idiomas;
- Documentação, fóruns de discussão e livros eletrônicos (*ebooks*) gratuitos;
- Blocos básicos e avançados, incluindo até mesmo controle PID em malha fechada e reconhecimento de imagens.

A Fig. 1 ilustra uma tela do Projeto BIPES, com um programa elaborado usando blocos.



Fonte: Autores.

Figura 1. Exemplo de tela do Projeto BIPES, com blocos sendo utilizados para programar uma placa Raspberry Pi Pico.

Nesses dois anos de existência do projeto, o BIPES ultrapassou os dez mil usuários em mais de cem países, com dezenas de milhares de aplicações desenvolvidas, e a adoção da ferramenta por escolas, bem como a sua integração com diversos produtos e *kits* educacionais. Para utilizar o BIPES, basta acessar <https://bipes.net.br/ide/>.

Este artigo apresenta um histórico do projeto, incluindo relatos de sucesso no seu uso aplicado a atividades didáticas, oficinas e até mesmo desenvolvimento de soluções profissionais. O artigo está organizado da seguinte forma: a seção “Trabalhos relacionados” apresenta e discute trabalhos relacionados com o Projeto BIPES; a seção “Projeto BIPES e sua abordagem” descreve a concepção e arquitetura do BIPES, suas funcionalidades e formas de uso; a seção “BIPES 3: desenvolvimento da nova versão” traz a nova arquitetura do BIPES em desenvolvimento; a seção “Resultados” detalha alguns resultados e relatos do uso do BIPES; e a seção “Conclusão” contém as considerações finais deste trabalho.

Trabalhos Relacionados

Uma linguagem de programação gráfica (VPL, do inglês *visual programming language*) manipula informação visual, suporta interação visual e/ou permite programação com expressões visuais (GOLIN; REISS, 1990). As linguagens de programação “regulares” são comumente a implementação de um paradigma de programação, por exemplo, a linguagem C (sem classes). Uma classificação proposta por Burnett e Baker

(1994) divide as VPLs em 11 paradigmas nos quais a linguagem de programação pode ser classificada. Mais notavelmente, são as linguagens de fluxo de dados e orientadas a objetos.

Como exemplo de linguagem visual de fluxo de dados, está o popular *software* LabVIEW. Na implementação do LabVIEW, o diagrama de blocos é uma estrutura orientada a dados (*dataflow*) de subprogramas e funções primitivas que representa a funcionalidade de um instrumento, uma hierarquia de módulos com um painel frontal interativo. Exemplos mais recentes de linguagem visual de fluxo de dados são Node-RED (<https://nodered.org/>) e NoFlow (<https://noflojs.org/>), bibliotecas baseadas na *web* e expansíveis para implementar ambientes de programação visual.

As linguagens visuais orientadas a objetos implementam um método mais convencional de programação visualmente, já que seu foco é orientar o desenvolvimento e simplificar o teste de rotinas orientadas a objetos. Em geral, o bloco é colocado visualmente na sequência de execução e não é reutilizável. Ou seja, várias entradas não podem ser conectadas ao mesmo bloco para acionar um método de forma independente. A biblioteca visual orientada a objetos mais usada é o projeto de código aberto do Google Blockly, do qual dependem outros 777 projetos (<https://github.com/google/blockly/network/dependencies>), incluindo o Scratch (<https://github.com/LLK/scratch-blocks>) e MIT App Inventor (<https://github.com/mit-cml/appinventor-sources>), aplicativos desenvolvidos como ponto de partida para aprender programação.

Considerando a popularidade de longo prazo das VPLs, tanto como um produto completo, como no caso do LabVIEW e Scratch, quanto como biblioteca, como no caso do Google Blockly, há demanda por VPLs que achatam a curva de aprendizado de uma linguagem de interface/programação, facilitam o desenvolvimento de rotinas e ajudam a lidar com dados.

Todos os *softwares*, bibliotecas e *firmwares* usados pelo BIPES são de código aberto. Portanto, é importante entender a percepção do público-alvo sobre esse tipo de *software*. O *software* de código aberto é um tema de interesse tanto para a comunidade de pesquisa quanto para a indústria, cujo interesse tem sido crescente, mas a adoção ainda parece ser limitada pela dificuldade de avaliar sua qualidade (LENARDUZZI *et al.*, 2020). Do ponto de vista educacional, a acessibilidade da ferramenta tem bastante peso no momento de seleção.

É importante notar que há casos em que uma fração do *software* é de código aberto ou que a implementação oficial tenha outra licença e políticas de privacidade distintas, que podem restringir o uso dependendo do contexto, por exemplo, não poder utilizá-la ou modificá-la para fins comerciais. Apesar disso, a percepção sobre o *software* de código aberto parece ser positiva. Em um estudo recente sobre a percepção dos estudantes de engenharia a respeito de um *software* de código aberto (NASCIMENTO; CHAVEZ; BITTENCOURT, 2018), concluiu-se que 93% dos formandos acreditavam que o projeto *open source* analisado era semelhante aos da indústria de *software* em relação a tamanho e complexidade. Já Carrington e Kim (2003) identificaram que 67% acreditam que o *software* de código aberto estudado fornece “uma experiência de *software* do mundo real”.

Várias plataformas com proposta semelhante à do BIPES estão disponíveis para uso, como MicroBlocks, Microsoft MakeCode, BlocklyDuino e XOD. As características de cada uma delas são resumidas nos Quadros 1 e 2 e então discutidas.

Quadro 1. Protocolos de comunicação.

Plataforma	TCP/IP	Serial	Bluetooth
MicroBlocks	Não	Sim	Não
Microsoft MakeCode	Não	Sim	Sim
BlocklyDuino	Não	Não	Não
XOD	Não	Sim	Não
BIPES	Sim	Sim	Sim

Fonte: Marques (2022).

Quadro 2. Características de execução.

Plataforma	Servidor	Local	Nuvem
MicroBlocks	Sim	Não	Não
Microsoft MakeCode	Sim	Não	Sim
BlockyDuino	Sim	Não	Não
XOD	Sim	Sim	Não
BIPES	Sim	Sim	Parcial

Fonte: Marques (2022).

MicroBlocks é uma solução educacional que permite programação lógica básica e controle de pequenos motores e telas via porta serial. A plataforma tem como alvo os dispositivos que executam o *firmware* NodeMCU, como, por exemplo, o ESP32, e está disponível sob a licença pública do Mozilla (Mozilla Public License v2.0).

Microsoft MakeCode consiste em uma plataforma proprietária baseada no Programming eXperience Toolkit, esta sob licença MIT (<https://github.com/Microsoft/pxt>), e, para embarcados, concentra-se em *kits* educacionais como Lego Mindstorms e micro:bit.

BlockyDuino2 é um projeto disponível sob a licença Apache (Versão 2.0) que gera código C++ direcionado às placas Arduino (<https://github.com/BlocklyDuino/BlocklyDuino-v2>). Ao contrário das outras soluções, o BlockyDuino2 não possui um protocolo de comunicação com outros dispositivos, mas comanda o ambiente de desenvolvimento do Arduino por meio de um servidor local configurado pelo usuário.

Finalmente, o XOD implementa sua própria linguagem de programação visual baseada em sinais elétricos, similarmente à analogia do LabVIEW, porém com os blocos fixados em grades, e está sob a licença AGPL-3.0 (<https://github.com/xodio/xod>).

Em contraste, o Projeto BIPES está disponível sob a licença GPL e é aberto a contribuições (<http://www.bipes.net.br/source.html>). Um grande diferencial é o suporte a diversas placas de desenvolvimento, incluindo ESP8266, ESP32, Arduino e Raspberry Pi Pico. A plataforma é capaz de executar em servidor, e seus recursos da nuvem, como compartilhamento de projetos e armazenamento de dados MQTT, são anônimos. Por decisão dos contribuidores, a plataforma não sincroniza projetos com a nuvem, pois não força o credenciamento do usuário nem coleta dados pessoais. Como resultado de Marques (2022), a plataforma é a única com suporte a várias instâncias simultâneas.

Projeto BIPES e sua Abordagem

Uma das principais premissas do Projeto BIPES é permitir a programação de dispositivos embarcados de forma intuitiva e simples para usuários leigos e usuários avançados utilizando uma plataforma que dispense a necessidade de instalação e configuração de qualquer *software*. Para atender a esse requisito de projeto, o BIPES foi concebido como uma aplicação *web* (HTML + JavaScript) que pode ser utilizada em qualquer dispositivo que possua um navegador *web*, tais quais o Google Chrome ou o Mozilla Firefox. Além disso, toda lógica de funcionamento do BIPES ocorre localmente, de forma que o BIPES funcione *offline* / sem conexão com a internet. Esse requisito foi pensado para que seja possível usar o BIPES em escolas com acesso restrito à internet.

Para viabilizar que o BIPES atenda a essa premissa, são utilizadas tecnologias consolidadas como MicroPython, HTML, Javascript, WebSockets e, especialmente, a tecnologia *web serial*, que permite que o navegador *web* tenha acesso direto a dispositivos seriais, tipicamente conectados via USB, sem a necessidade de *plugins* ou *drivers* especiais. De fato, esse recurso permite que o BIPES ofereça o console / terminal de interação com as placas pelo navegador *web*, seja em modo *online*, seja *offline*.

BIPES 3: desenvolvimento da nova versão

Com o intuito de facilitar implementações complexas no BIPES, uma nova arquitetura está sendo desenvolvida. Com essa arquitetura, não só contribuições para a versão oficial são facilitadas, como também será permitido que qualquer pessoa tenha sua própria versão funcional em poucos minutos. Desse modo, novas funcionalidades e mesmo áreas de pesquisa podem ser desenvolvidas como módulos independentes do código-fonte principal.

A nova arquitetura possibilitou uma atualização ao módulo easyMQTT, concebido por Silva (2020). O módulo deixa de ser executado em PHP e passa a ser integrado no *framework* Flask, centralizando os métodos e facilitando a manutenção do código. Também, é implementada conexão MQTT via WebSocket, que estabelece uma conexão *full-duplex* contínua com a plataforma, em vez de solicitações cíclicas via HTTP. Para que isso fosse possível, o MQTT broker Mosquitto (<https://mosquitto.org/>) foi reconfigurado e a biblioteca paho.js (<https://www.eclipse.org/paho/index.php?page=clients/js/index.php>) incluída como dependência.

Outra atualização importante é a utilização de modelos para a seleção de quais blocos estão disponíveis para cada dispositivo. Em vez de criar de modo estático arquivos repetitivos com a definição de cada bloco para cada dispositivo, os blocos são identificados por um código e cada dispositivo-alvo possui uma lista com esses códigos. Isso permite que seja muito fácil criar dispositivos-alvo, adaptando as necessidades de cada usuário e que haja melhor organização do código-fonte, com os blocos separados por categorias em vez de por dispositivos.

Uma funcionalidade exclusiva dessa nova versão do BIPES é o suporte a várias janelas de programação (instâncias) com o desenvolvimento de biblioteca própria. Isso permite que o usuário abra diferentes módulos e projetos em abas e janelas distintas, incluindo compartilhar o ciclo de leitura-avaliação-impressão (REPL, do inglês *read-eval-print-loop*) de um mesmo dispositivo entre as instâncias. Para que isso seja possível, a plataforma utiliza a Broadcast Channel API (https://developer.mozilla.org/en-US/docs/Web/API/Broadcast_Channel_API) para realizar o despacho de modificações entre as instâncias.

Resultados

Nos resultados do uso de BIPES em situações de aprendizagem, tivemos um longo período de exposição não-formal na 18ª Semana Nacional de Ciência e Tecnologia, realizada entre 3 e 10 de dezembro de 2021, quando tivemos a oportunidade de utilizar o BIPES em um estande prático da 1ª Olimpíada Brasileira de Satélites MCTI (AROCA, 2022), visto que os participantes do evento foram instigados a compreender o uso do BIPES para as aplicações aeroespaciais, como o caso da programação de pequenos protótipos de satélites educacionais (CubeSats e CanSats). Durante o evento, optou-se por aplicar uma abordagem construcionista, aquela pela qual o aluno/aprendiz constrói por intermédio do computador o seu conhecimento (PAPERT, 1986), como método para sua efetivação, de modo a aproximar as diferentes tecnologias presentes da realidade do visitante.

O estande foi construído de maneira que o visitante tivesse a oportunidade de interagir com os satélites educacionais, radiocomunicação, impressão em 3-D e programação via BIPES. Possibilitou-se a produção de diferentes níveis de abstração que poderiam proporcionar mudanças na estrutura cognitiva, como a abstração empírica, ao conceber a forma do satélite (pequeno, diferente dos habituais satélites de telecomunicações), refletir sobre o seu propósito e imaginar suas próprias aplicações, de modo a proporcionar uma abstração reflexiva em que há uma reorganização desses conhecimentos considerando seu conhecimento prévio (VALENTE, 2005).

Na via da programação, os alunos foram introduzidos a um programa que aferia, em tempo real, utilizando os sensores do satélite, a temperatura e a umidade do local, além de dados inerciais, realizando um registro temporal durante o período. Ao acessar um link por QR code, atuaram também em uma aplicação de Internet das Coisas (IoT) para o acionamento de um relê que ligava e desligava uma lâmpada LED. Nesse caso, os visitantes assumiram o papel de fornecer a informação para que o computador trabalhasse, não o contrário, como em um método instrucionalista.

Os visitantes ao longo do evento foram questionados, por um mediador treinado para a exposição, sobre esse tipo de interação entre a programação, o satélite e a Terra e levados à reflexão quanto ao seu uso. Dessa forma, puderam integrar suas diferentes esferas imaginativas e sociais ao pautar sua realidade como fonte de ideias e conhecimento para problemas a serem resolvidos pela interação entre computador e máquina, mediante a plataforma BIPES.

Houve dificuldades, por não ser uma prática uniforme, ainda mais considerando que o público era muito diverso, sendo necessário adaptar a exposição e interação com o estande a desde crianças (de 2 a 10 anos) a idosos com pouca vivência digital. Todavia, o BIPES atuou como um facilitador, por ser uma linguagem baseada em blocos e pela existência dos painéis de visualização de dados, que apresentava a visualização em tempo real dos dados coletados pelos sensores. Logo, foi um potencializador de uma prática construcionista nesse ambiente não-formal. A prática de Internet das Coisas gerou bastante empolgação, e os participantes continuavam a testar o sistema mesmo após terem saído do estande.

Conclusão

Com dois anos de existência, o Projeto BIPES torna a programação de dispositivos embarcados mais fácil e acessível, o que introduz e viabiliza aplicações de automação, Internet das Coisas e robótica para públicos diversificados.

As tecnologias nacionais desenvolvidas para a plataforma são contribuições abertas e de livre uso. Já os materiais auxiliares, como livros e minicursos, contribuem para tornar o tema mais didático e acessível.

O código-fonte encontra-se em <https://github.com/BIPES>, e uma nova versão está sendo desenvolvida para proporcionar que módulos mais complexos sejam implementados, permitindo que novas áreas de estudo sejam exploradas.

Dados analíticos providos pela *tag* do Google Analytics no BIPES indicam que 46% do público é feminino, e 27,5% tem até 24 anos. Em acordo com a internacionalização do projeto, os países com mais usuários recorrentes são, em ordem decrescente de uso: Alemanha, China, Brasil, Estados Unidos da América e Austrália. Além disso, sabe-se de empresas na Alemanha, Indonésia e China explorando o BIPES como plataforma de programação para *kits* didáticos.

Os próximos passos incluem, além do constante desenvolvimento da plataforma, a produção de cursos e *workshops* de treinamento e a aplicação para professores e monitores.

Conflito de Interesse

Nada a declarar.

Contribuição dos Autores

Conceitualização: Marques JAG; Aroca RV. **Metodologia:** Marques JAG; Gueta WF; Pazelli TFPAT; Aroca RV. **Investigação:** Marques JAG; Gueta WF; Aroca RV. **Redação – Primeira Versão:** Marques JAG; Gueta WF; Pazelli TFPAT; Aroca RV. **Redação – Revisão & Edição:** Marques JAG; Pazelli TFPAT; Aroca RV.

Financiamento

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
Código de financiamento 001

Disponibilidade de Dados da Pesquisa

Todos os dados foram gerados/analísados no presente artigo.

Agradecimentos

Não se aplica.

Referências

ALVES, S. F. R.; FERASOLI FILHO, H.; PEGORARO, R.; CALDEIRA, M. A. C.; ROSÁRIO, J. M.; YONEZAWA, W. M. Proposal of educational environments with mobile robots. In: INTERNATIONAL CONFERENCE ON ROBOTICS, AUTOMATION AND MECHATRONICS, 5., 2011. **Anais [...]**. 2011. p. 264-269. <https://doi.org/10.1109/RAMECH.2011.6070493>

AROCA, R. V.; GUETA, W. F.; JESUS, A. A.; TRUBIENE, C. S.; SILVA, J. P. V. B.; MARQUES, J. A. G. **Programação de CanSats PION e CubeSats PION por blocos usando BIPES**. Olimpíada Brasileira de Satélites, 2022.

AROCA, R. V.; GUETA, W. F.; MARQUES, J. A. G.; PAZELLI, T. F. P. A. T. **An introduction to the internet of things and embedded systems using block-based programming with BIPES and ESP8266/ESP32**. [s.l.: s.n.], 2021a.

AROCA, R. V.; GUETA, W. F.; MARQUES, J. A. G.; PAZELLI, T. F. P. A. T. **Uma introdução à internet das coisas e sistemas embarcados utilizando programação por blocos com BIPES e ESP8266/ESP32**. [S.l.: s.n.], 2021b.

BRACKMANN, C. P. **Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica**. 2017.

BUNDY, A. Computational thinking is pervasive. **Journal of Scientific and Practical Computing**, v. 1, p. 67-69, 2007.

BURNETT, M. M.; BAKER, M. J. A classification system for visual programming languages. **Journal of Visual Languages and Computing**, v. 5, n. 3, p. 287-300, 1994. <https://doi.org/10.1006/jvlc.1994.1015>

CARRINGTON, D.; KIM, S.-K. Teaching software design with open source software. *In: ANNUAL FRONTIERS IN EDUCATION*, 3., 2003. **Anais [...]**. IEEE, 2003. p. S1C-9.

CHELLA, M. T. SimRobô: simulador para robótica com propósito educacional. *In: WORKSHOP DE ROBÓTICA EDUCACIONAL*, 3., 2012. **Anais [...]**. 2012.

CHUNG, C. J.; CARTWRIGHT, C.; CHUNG, C. Robot music camp 2013: an experiment to promote STEM and computer science. *In: IEEE INTEGRATED STEM EDUCATION CONFERENCE*, 2014. **Anais [...]**. IEEE, 2014. p. 1-7.

CUKURBASI, B.; KIYICI, M. High school students' views on the PBL activities supported via flipped classroom and LEGO practices. **Journal of Educational Technology & Society**, v. 21, n. 2, p. 46-61, 2018.

EGUCHI, A. Educational robotics for promoting 21st century skills. **Journal of Automation, Mobile Robotics and Intelligent Systems**, v. 8, n. 1, p. 5-11, 2014. Disponível em: <https://www.jamris.org/index.php/JAMRIS/article/view/284/284>. Data de acesso: 11 de abril 2023.

GALVAN, S.; BOTTURI, D.; CASTELLANI, A.; FIORINI, P. Innovative robotics teaching using Lego sets. *In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION*, 2006. **Anais [...]**. IEEE, 2006. p. 721-726. <https://doi.org/10.1109/ROBOT.2006.1641795>

GOLIN, E. J.; REISS, S. P. The specification of visual language syntax. **Journal of Visual Languages & Computing**, v. 1, n. 2, p. 141-157, 1990. [https://doi.org/10.1016/S1045-926X\(05\)80013-8](https://doi.org/10.1016/S1045-926X(05)80013-8)

GOMES, A.; AREIAS, C.; HENRIQUES, J.; MENDES, A. J. Aprendizagem de programação de computadores: dificuldades e ferramentas de suporte. **Revista Portuguesa de Pedagogia**, v. 42, n. 2, p. 161-179, 2008. https://doi.org/10.14195/1647-8614_42-2_9

HUANG, K. H.; HUANG, P.-L. Lego robotics and group learning: Exploring the effects of gender, age, and family background. *In: INTERNATIONAL CONFERENCE ON COMMUNICATION SOFTWARE AND NETWORKS*, 3., 2011. **Anais [...]**. IEEE, 2011. p. 219-222.

HUANG, W.-Y.; HU, C.-F.; WU, C.-C. The use of different kinds of robots to spark student interest in learning computational thinking. *In: INTERNATIONAL CONFERENCE ON LEARNING AND TEACHING IN COMPUTING AND ENGINEERING*, 2018. **Anais [...]**. IEEE, 2018. p. 11-16.

LENARDUZZI, V.; TAIBI, D.; TOSI, D.; LAVAZZA, L.; MORASCA, S. Open source software evaluation, selection, and adoption: a systematic literature review. *In: EUROMICRO CONFERENCE ON SOFTWARE ENGINEERING AND ADVANCED APPLICATIONS*, 46., 2020. **Anais [...]**. IEEE, 2020. p. 437-444. <https://doi.org/10.1109/SEAA51224.2020.00076>

MARQUES, J. A. G. **Collaboration on the development of BIPES and the inclusion of Control Systems blocks**. 84f. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de São Carlos, São Carlos, 2022.

MIRANDA, J. R.; SUANNO, M. V. R. Robótica na escola: ferramenta pedagógica inovadora. *In: WORKSHOP OF ROBOTICS ON EDUCATION*, 3., 2012, Fortaleza. **Anais [...]**. 2012. p. 18-19.

NASCIMENTO, D. M. C.; CHAVEZ, C. F. G.; BITTENCOURT, R. A. The adoption of open source projects in engineering education: a real software development experience. *In: IEEE FRONTIERS IN EDUCATION CONFERENCE*, 2018. **Anais [...]**. IEEE, 2018. p. 1-9. <https://doi.org/10.1109/FIE.2018.8658908>

ORGANIZAÇÃO DAS NAÇÕES UNIDAS PARA A EDUCAÇÃO, A CIÊNCIA E A CULTURA (UNESCO). **Life skills**. UNESCO: International Bureau of Education, 2020. Disponível em: <http://www.ibe.unesco.org/en/glossary-curriculum-terminology/life-skills>. Acesso em: 16 jan. 2023.

PAPERT, S. **Constructionism: a new opportunity for elementary science education**. Massachusetts: Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group, 1986.

SCHINA, D.; USART, M.; ESTEVE-GONZALEZ, V. Participants' perceptions about their learning with first LEGO® league competition—a gender study. *In: INTERNATIONAL CONFERENCE ON ROBOTICS IN EDUCATION*, 2019. **Anais [...]**. Cham: Springer, 2019. p. 313-324.

SILVA, C. A. **Desenvolvimento e validação de módulo de comunicação MQTT para a plataforma BIPES para aplicações de internet das coisas**. 2020.

SILVA JUNIOR, A. G.; GONÇALVES, L. M. G.; CAURIN, G. A. P.; TAMANAKA, G. T. B.; HERNANDES, A. C.; AROCA, R. V. BIPES: block based integrated platform for embedded systems. **IEEE Access**, v. 8, p. 197955-197968, 2020. <https://doi.org/10.1109/ACCESS.2020.3035083>

SOUZA, I. M. L.; ANDRADE, W. L.; SAMPAIO, L. M. R.; ARAÚJO, A. L. S. O. A systematic review on the use of LEGO® robotics in education. *In: IEEE FRONTIERS IN EDUCATION CONFERENCE*, 2018. **Anais [...]**. IEEE, 2018. p. 1-9.

VALENTE, J. A. Informática na educação: instrucionismo x construcionismo. **Educação Pública**, v. 2, n. 1, 2005.

ZANETTI, H.; OLIVEIRA, C. Práticas de ensino de programação de computadores com robótica pedagógica e aplicação de pensamento computacional. *In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO*, 2015. **Anais [...]**. 2015. p. 1236.

Recebido: 1º set. 2022

Aprovado: 1º mar. 2023

Editores Associados:

Ana Clara Bortoleto Nery  e Eduardo Alessandro Kawamura 