UpRobotics: Robótica Educacional Utilizando Linguagem Visual Baseada em Blocos

Francisco Marinho, Edwin Monteiro, Raimundo Barreto
Instituto de Computação (IComp)
Universidade Federal do Amazonas (UFAM)
Manaus, Brasil
{francisco, edwin, rbarreto}@icomp.ufam.edu.br

***Resumo*—A tecnologia é um recurso fundamental que traz impacto em todos os setores da vida cotidiana. Dessa forma, há hipóteses que apontam para uma demanda crescente por profissionais com habilidades e competências na área de computação, tanto no setor industrial quanto científico. No entanto, ainda há uma carência latente no que tange ao desenvolvimento do pensamento computacional no ensino básico, particularmente em sociedades com baixo nível socioeconômico. Existem diversas razões que vão muito além de recursos limitados, ou falta de interesse dos alunos, que inclui a dificuldade de entendimento e abstração dessas tecnologias, tanto por parte dos educadores, quanto dos alunos. Desta forma, este artigo propõe um método para exercitar o pensamento computacional utilizando a Matemática de maneira conjunta com duas estratégias bem sucedidas: a robótica educacional e a utilização de linguagem visual baseada em blocos gráficos arrastáveis. O UpRobotics é o artefato principal desta pesquisa, uma ferramenta voltada ao público escolar infantil, capaz de explorar conhecimentos científicos dos alunos, tanto matemáticos quanto computacionais, capacitando-os para manipular um braço robótico usando uma linguagem visual baseada em blocos.**

***Index Terms*—Robótica Educacional, Linguagem Visual Baseada em Blocos, Geração Automática de Código, Ângulos, Braço Robótico**

I. Introdução

A tecnologia tem um forte impacto na força de trabalho, no desenvolvimento e na prosperidade socioeconômica. Deste modo, a computação torna-se uma prioridade educacional desde o ensino básico. É necessário equipar a força de trabalho do futuro por intermédio de conhecimentos e habilidades de ciência da computação, engenharias e robótica. Todavia, essa integração da computação não acontece devido à falta de recursos e de treinamento profissional para apoiar o aprendizado dos alunos. Segundo Miller et. al. [1] um fator que também influencia na carência do ensino de computação nas escolas é a falta de interesse de alunos, principalmente aqueles com baixo nível socioeconômico pois, de acordo com a pesquisa dos autores, para medir o interesse de 148 educandos de 6ª série ao terceiro ano do ensino médio, os alunos de baixa renda tiveram desempenhos inferiores a seus pares de alta renda durante uma lição apresentada sobre robótica computacional em duas escolas públicas. Ao final da pesquisa, após análise dos resultados, os autores sugerem que a integração de aulas introdutórias de computação e robótica em sala de aula com alunos de baixo nível socioeconômico pode ajudá-los a alcançar níveis semelhantes de interesses e atitudes de engenharias em relação a alunos com maior renda familiar. Dessa forma, essas habilidades e competências adquiridas corroboram e se enquadram em um termo bastante difundido na área da ciência da computação, denominado Pensamento Computacional, isto é, uma maneira de resolver problemas comuns do dia a dia de maneira sistemática baseada em um processo que se inicia desde a análise do problema e organização dos pensamentos, representação de dados através de abstrações que podem ser representadas em modelos ou simulações, e por fim, automatização de problemas de maneira eficiente visando criar uma forma de padronizar algumas soluções para problemas diversos [3] [4].

Assim sendo, baseado em aspectos levantados na literatura e no estado da arte, chegou-se à ideia do principal desafio desse trabalho: a criação de um método que facilite o exercício do pensamento computacional na educação básica que possa contribuir para a minimização da problemática, além da contribuição científica na área de robótica e linguagens de programação de fácil assimilação. Para isso, a ideia foi utilizar duas estratégias bem sucedidas: a robótica educacional e a utilização de linguagem visual baseada em blocos gráficos arrastáveis, com a finalidade de aplicar o conhecimentos teóricos de ângulos em articulações de um braço robótico de baixo custo. Dentre os fatores que podem ser considerados diferenciais nesse trabalho, destaca-se a possibilidade de alunos que nunca tiveram contato com programação conseguirem programar um braço robótico em poucos minutos utilizando apenas seus conhecimentos de lógica e matemática, principalmente no que se refere aos assuntos de ângulos retos, obtusos, agudos ou rasos. Para isso, foi necessário criar uma linguagem de programação baseada em blocos com aspectos semelhantes à linguagem natural, capaz de gerar código-fonte automático, visando a sua execução a partir de um microcontrolador Arduino, acessório de baixo custo e necessário para movimentar o braço robótico.

As seções subsequentes deste artigo estão organizadas da seguinte maneira: A Seção 2 descreve a Fundamentação Teórica elencando os principais conceitos para compreensão do trabalho. A Seção 3 comenta os trabalhos relacionados, visando apresentar o estado da arte e alguns resultados atingidos nos respectivos trabalhos. A seção 4 apresenta o método proposto detalhando suas devidas fases. A Seção 5 descreve o planejamento e detalhes do experimento juntamente com os resultados preliminares. Por fim, a Seção 6 discorre sobre as

considerações finais e trabalhos futuros.

II. Fundamentação Teórica

A. O Pensamento Computacional

Em linhas gerais, o Pensamento Computacional (PC) é um dos instrumentos mais importantes da educação moderna, pois é o processo de elaborar problemas e soluções de uma forma que um computador possa efetivamente entender e realizar [4]. Ele é requisitado por ter a capacidade de desenvolver habilidades cognitivas em todas as áreas da matriz curricular. Swaid, e Suid [5] afirmam que o PC é uma habilidade que toda criança do século XXI deve ter. Nesse sentido a comunidade científica trabalha para desenvolver e expandir esse conhecimento para que todas as pessoas tenham a oportunidade de desenvolver suas habilidades cognitivas.

B. A Robótica Educacional

A robótica é a ciência responsável pelo estudo dos robôs, sendo estes, máquinas que podem ser usadas para realizar trabalhos específicos. Alguns robôs podem trabalhar sozinhos, outros precisam ser controlados o tempo todo por seres humanos para que o trabalho seja realizado. Deste modo, os robôs se tornaram instrumentos importantes para o avanço da ciência e humanidade em diversos cenários. É importante frisar a robótica está presente em outras áreas, como na medicina, no setor de fábrica, indústria automotiva e na educação.

A robótica educacional é uma metodologia de ensino que visa estimular o aluno a praticar os conhecimentos aprendidos no conteúdo curricular, possibilitando e incentivando o trabalho em equipe, raciocínio lógico e estimulando a criatividade para desenvolver soluções [2] [12]. Essa metodologia tem embasamento teórico na teoria construtivista de Piaget [6], que considera a manipulação de objetos um método chave para que a criança, por si só, seja capaz de construir o seu próprio conhecimento. Além do mais, Papert [7] corrobora com essa teoria afirmando que essa construção ocorre de forma mais afetiva quando a criança se esforça de forma consciente na construção de algo que ela possa manipular.

Assim sendo, são criadas algumas estratégias dentro da robótica educacional, dentre elas, a robótica desplugada, isto é, um novo formato didático para o ensino da robótica sem a necessidade de robô ou programação em linguagens mais complexas para programá-lo. Esse conceito é utilizado no trabalho de Miller et. al. [1] na próxima Seção.

C. Linguagem de Programação Visual (LPV)

A LPV é uma linguagem que permite ao usuário criar programas por intermédio de manipulação gráfica [9]. Alguns exemplos mais comuns de utilização dessas linguagens são:

- arrastar blocos ao redor de uma tela (exemplos: MatLab e Scratch);
- diagramas de fluxo ou de estados;
- usar ícones ou representação não textual.

Para alcançar os objetivos desta pesquisa foi necessária a criação de uma linguagem de programação personalizada inteiramente baseada em blocos gráficos arrastáveis. A Seção V detalha as características da linguagem.

III. Trabalhos relacionados

Sales et. al. [10] relataram a experiência de uso do Arduino programado com Scratch para o desenvolvimento de conteúdos de matemática. O estudo visou compreender se a programação com Scratch poderia potencializar o interesse dos alunos pelas aulas de geometria. Como metodologia, eles propuseram uma sequência didática na qual alunos da educação básica com faixa etária entre 11 e 16 anos, de uma escola de rede pública municipal, criaram e programaram cubos de LED nas aulas de matemática, desenvolveram aprendizagens relacionadas ao pensamento computacional e a geometria.

O trabalho de Vidal, Sampaio e Pontes [2] apresentou uma ferramenta de programação visual voltada para a robótica educacional com a utilização de componentes de baixo custo como o Arduino e tecnologias gratuitas. O objetivo dos pesquisadores foi apoiar o ensino-aprendizagem de lógica de programação para promover o pensamento computacional fazendo experimentos em escolas públicas do Rio de Janeiro.

O trabalho de Miller et. al. [1] teve o intuito de aumentar o interesse dos alunos do ensino fundamental em atividades de ciência, tecnologia, engenharia e matemática. Para isso os autores utilizaram robótica desplugada com interface de usuário magnética tangível, isto é, blocos conectáveis de movimentos cortados em uma máquina a laser. Como resultado, eles atingiram o objetivo proposto, permitindo que alunos de baixo nível socieconômico alcancem interesse níveis de aprendizagem semelhantes a seus pares de alto nível socieconômico.

Todos os trabalhos supracitados nessa seção possuem relação com esta proposta devido a utilização de linguagens visuais para o ensino de áreas da ciência focadas no contexto de pensamento computacional no ensino básico.

IV. Método Proposto

Esta seção apresenta as fases e subfases envolvidas no método adotado para esse trabalho.

A. Levantamento Bibliográfico

Foi realizado um levantamento bibliográfico em fontes publicadas no período de 2017 a 2020, nas bibliotecas IEEE Explore, anais dos eventos disponibilizados na Biblioteca Digital Brasileira de Computação (BDBComp), livros científicos e sites governamentais.

B. Planejamento

Essa fase é composta por três subfases importantes que envolvem desde nivelamento teórico dos alunos até a conclusão com testes práticos.

1) Treinamento: a estratégia inicial foi elaborar uma aula básica sobre ângulos matemáticos com o intuito de familiarizar os participantes da pesquisa ao conteúdo.

2) Linguagem: Foi utilizada a tecnologia livre Google Blocklys que segundo Pasternak, Fenichel e Marshall [11] é uma biblioteca que adiciona um editor de código visual à aplicações web ou mobile, isto é, uma maneira intuitiva e visual de construir código-fonte, pois o editor utiliza blocos gráficos interligáveis para representar conceitos de código

como variáveis, expressões lógicas, decisões e laços.. Ele permite a criação de blocos personalizados (*Blockly Developer Tools*) permitindo que os desenvolvedores criem sua própria linguagem de programação. Outra característica louvável dessa ferramenta é permitir, por padrão, a geração de código-fonte em Javascript, Python, PHP, Lua e Dart. Para os blocos personalizados, a biblioteca permite que o desenvolvedor programe um código-fonte personalizado de acordo com sua necessidade, deste modo, a linguagem de programação escolhida nessa proposta foi a Linguagem C.

3) Validação: A validação consiste em verificar se o objetivo principal dessa proposta foi atingido com base em métricas pré-estabelecidas. Para a realização dos experimentos finais foi necessário a aquisição de um Arduino, um kit de braço robótico MeArm em MDF e outro em Acrílico e a definição dos movimentos básicos necessários para que o braço robótico possa realizar algumas tarefa simples, como pegar um objeto de uma posição A e colocá-lo em uma posição B.

C. Execução dos Experimentos

Essa fase é a união de testes feitos em laboratório com o braço visando compreender os movimentos, além de testar o UpRobotics em um experimento feito com os alunos do 7° ano. Esses experimentos estão descritos na Seção V.

D. Implantação do Projeto Atual

A implantação consistiu em disponibilizar a primeira versão da ferramenta na internet visando apoiar o ensino-aprendizagem da robótica educacional, permitindo aos educadores promover o pensamento computacional com seus alunos em salas de aula, viabilizando experimentos práticos sobre o conteúdo de geometria no que se refere a ângulos.

V. UpRobotics: Criação, Experimento e Resultados Preliminares

A. A Linguagem Visual desenvolvida

Visando representar a linguagem criada com sintaxe bem definida e semântica precisa, foi elaborada uma linguagem formal e um Autômato Finito Determinístico (AFD) Incompleto apresentado na Figura 1. Esse autômato é um formalismo particularmente adequado ao reconhecimento dessa linguagem. Assim sendo, as características principais desse ADF incompleto são: o autômato recebe uma palavra p de entrada, composta por uma sequência arbitrária de símbolos pertencentes ao alfabeto Σ. A leitura de cada símbolo em p implica na mudança de estado no autômato [13].

A Linguagem Visual de Programação Baseada em Blocos construída para o UpRobotics é definida da seguinte maneira: L é uma linguagem pertencente ao alfabeto Σ de modo que $\Sigma = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ tal que L = { $p \in \{\{1^k\ 2^k\ 3^k\ 4^k\}\ U\ \{1^k\ 3^k\ 2^k\ 4^k\}\}\ 5^k\ \{6, 7, 8, 9\}^n \mid k = 1,\ n \geq 1\}$ onde p é uma palavra gerada pela linguagem L.

A Tabela I descreve os significado de cada estado no AFD Incompleto e a Tabela II descreve o significado de cada símbolo do alfabeto. A Figura 2 mostra um pequeno exemplo

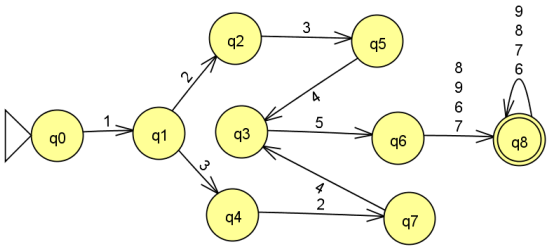

Figura 1. Autômato Finito da Linguagem Baseada em Bloco.

de blocos necessários para fazer o braço apontar um objeto a frente dele e logo em seguida voltar a posição anterior.

Tabela I
SIGNIFICADO DE CADA ESTADO DO AUTÔMATO

Estados	Significado
{q0}	Nenhum bloco foi adicionado
{q1}	Os motores foram importados
{q2,q7}	Os motores foram ativados
{q4,q5}	As portas seriais foram definidas
{q3}	O bloco para inicializar todos os movimentos foi adicionado
{q6}	A posição inicial do braço foi definida
{q8}	A garra foi aberta; Ou A garra foi fechada; Ou ângulo do braço foi movido da posição atual até a nova posição informada; Ou a altura do braço foi movida da posição A até a posição B; Ou a base girou o braço para a direita ou esquerda;

Tabela II
SIGNIFICADO DE CADA SÍMBOLO DO ALFABETO

Alfabeto (Σ)	Significado
1	Importar motores
2	Ativar motores
3	Definir as portas no Arduino
4	Inicializar todos os movimentos
5	Inicializar o robô a partir de um posição inicial padrão
6	Movimentar fechamento/abertura da garra robótica
7	Movimentar o ângulo do braço
8	Movimentar a altura do braço
9	Girar o braço para a direita ou para a esquerda

Na Figura 2 é possível notar alguns aspectos semelhantes à didática bem sucedida abordada no trabalho de Miller et al. [1] , visto que os blocos gráficos do UpRobotics possuem algumas características inspiradas nos blocos conectáveis utilizados pelos autores. O diferencial é que os blocos do UpRobotics geram código-fonte em linguagem C, que é compilado e embarcado em um Arduino por meio de uma IDE, movimentando o braço robótico.

A Figura 3 mostra a parte principal da ferramenta UpRobotics, a qual é dividida em 3 partes:

1) Caixa de ferramentas: local dividido em 4 categorias, onde cada categoria possui um conjunto de blocos. As categorias são: (i) Movimentos, que possui todos

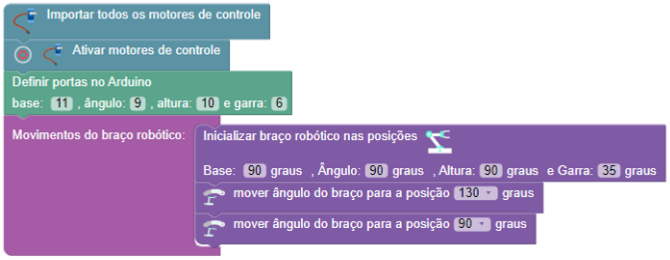

Figura 2. Exemplo de programação em blocos com o intuito de fazer o braço robótico apontar um objeto a frente dele e logo em seguida voltar a posição anterior.

os blocos principais que controlam as articulações do braço robótico, isto é, os quatro servomotores (garra, base, altura e ângulo); (ii) Funções, onde estão todos os códigos responsáveis por importações de bibliotecas do Arduino necessários, além de conter funções responsáveis por manipular a velocidade de movimentação do braço e controle das posições de um ponto A ao ponto B; (iii) Controle, onde está um bloco responsável por inicializar todos os movimentos ou ações que devem ser executadas; (iv) Arduino, contém um único bloco para definir as portas seriais no Arduino aonde serão conectados os servomotores.

2) Área de Trabalho: local para onde os blocos devem ser arrastados, montados, excluídos, preenchidos e manipulados.
3) Tela de saída do código gerado: uma tela onde os códigos gerados são mostrados de acordo com a montagem dos blocos na área de trabalho.

B. O Experimento

Para validar a ferramenta verificamos se código gerado possibilita a realização dos mesmos movimentos utilizados com o uso de dois *joysticks*. Então, antes da realização do experimento com os alunos, foi realizado dois experimentos com o braço robótico em laboratório. No primeiro experimento, o braço robótico deveria pegar um objeto de um lugar, girar e colocá-lo em outro lugar. No segundo experimento, ele deveria pegar um objeto de um lugar baixo, isto é, a superfície onde ele estava e colocar em cima de uma pilha de livros. Todos os passos para que ele fizesse as ações foram feitos por blocos interconectados na ferramenta proposta. O resultado foi promissor, pois foi feito exatamente como foi programado, inclusive respeitando a velocidade suave utilizada com *joystick*. A Figura 4 mostra como funciona o controle do braço por intermédio de dois *joysticks*.

Assim sendo, devido a situação de pandemia existente durante todo o período de realização desse projeto, não houve escolas abertas disponíveis para a aplicação do experimento para um número expressivo de alunos. Contudo, dois alunos (ambos com 12 anos de idade) do 7° ano do ensino fundamental se propuseram a participar do experimento, com todas os devidos cuidados básicos recomendados pela Organização Mundial de Saúde (OMS).

Tabela III
LISTA DE PASSOS UTILIZADOS NO EXPERIMENTO

Passos	Movimentos a serem feitos
1	inicializar robô com a posição padrão definida no bloco
2	abrir garra 35 graus
3	mover altura para a posição 50 graus
4	mover ângulo para a posição 180 graus
5	fechar a garra 160 graus
6	mover ângulo para a posição 90 graus
7	girar para a direita 110 graus
8	abrir garra 35 graus
9	girar o braço esquerda 90 graus
10	mover a altura para a posição 90 graus

1) O Nivelamento Teórico: Foi lecionado uma lição que durou cerca de 45 minutos com o intuito de familiarizar os alunos ao conteúdo Ângulos Matemáticos. Foi utilizado um slide com alguns exercícios simples ao final da apresentação, exercícios baseados nos livros distribuídos pelo governo para serem utilizados nas aulas sobre ângulos. Durante essas aulas foram apresentados alguns exemplos das aplicações práticas desses conhecimentos, como por exemplo, os braço robóticos utilizados pela NASA [8] para explorar o espaço. Além do mais, foi explicado como esses conceitos seriam utilizados no experimento para manipular o braço robótico.

2) A Demostração da Ferramenta: Após a finalização do breve nivelamento, foi mostrado como é o funcionamento do braço quando se utiliza dois *joysticks*. Em seguida, foi apresentado a ferramenta e como os blocos disponíveis poderiam ser utilizados para realizar exatamente os mesmos movimentos, mas sem o auxílio de *joysticks*, apenas utilizando o conceito de ângulos matemáticos.

3) O Experimento Prático: Foram utilizados dois braços robóticos iguais e de baixo custo no experimento. A diferença é que o primeiro podia ser controlado por dois *joysticks* e o segundo poderia ser programado utilizando a ferramenta UpRobotics, mas sem o auxílio de *joysticks*. Deste modo, foi dado o mesmo desafio para ambos os participantes.

Desafio: Na mesa onde estavam os robôs existia um pequeno objeto, fácil de ser agarrado pela garra robótica. Assim sendo, cada participante teve que realizar um conjunto de movimentos com o braço primeiramente utilizando *joysticks* e o objeto da mesa. Após fazer isso, o participante teria que fazer exatamente os mesmos movimentos no robô sem *joysticks* programando-o com a linguagem visual baseada em blocos do UpRobotics e utilizando o mesmo o objeto disponibilizado.

Deste modo, cada um dos participantes realizou o experimento sem o auxílio de qualquer pessoa no que tange a exercitar o pensamento computacional explorando seus conhecimentos adquiridos nas lições. Assim sendo, como base, foi apresentado a eles uma lista com 10 movimentos necessários para o braço carregar um objeto de uma posição A até uma posição B. A Tabela III é a lista de movimentos utilizada no experimento.

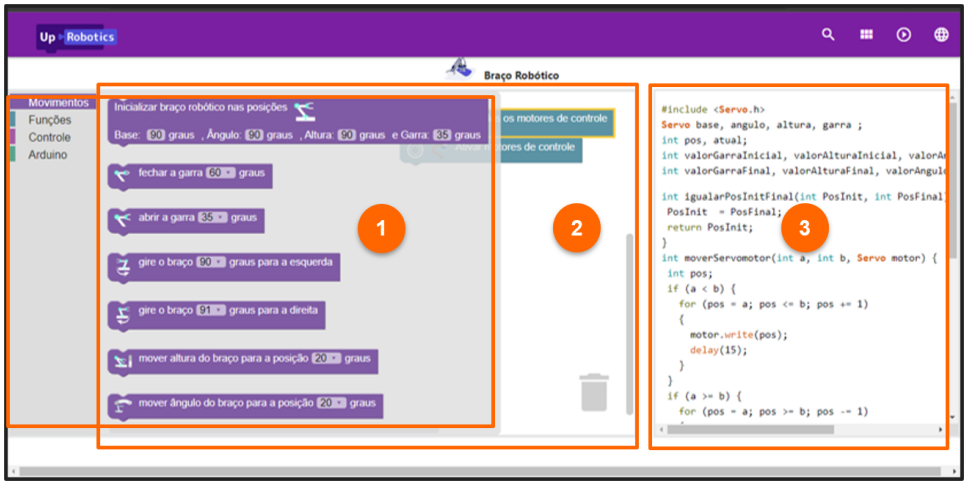

Figura 3. Parte principal da ferramenta criada.

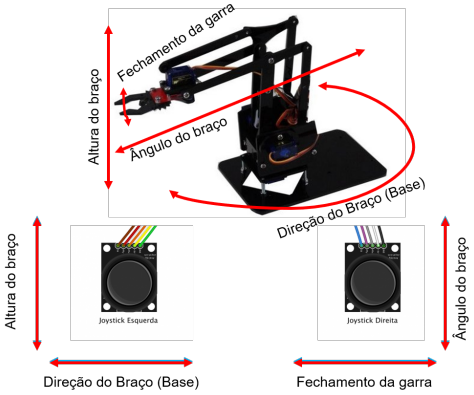

Figura 4. Relação entre *Joysticks* controladores e movimentos do braço.

C. Resultados Preliminares e Discussões

A métrica que decide se o experimento obteve resultados nesse cenário positivos ou negativos é baseada em um método de avaliação utilizado na obra de Jurafsky e Martin [9], no que tange a definição de precisão geral do modelo em Aprendizagem de Máquina. Nesse caso, avaliamos a precisão dos movimentos que usaram apenas a linguagem visual do UpRobotics em relação aos movimentos realizados pelo braço robótico utilizando *joysticks*. A precisão foca nos valores classificados como positivos:

$$prec = \frac{VP}{VP + VN} \tag{1}$$

Adaptando essa definição para o nosso problema, temos:

- VP (Verdadeiros Positivos): Movimentos produzidos do braço robótico exatamente como deveria acontecer se estivesse utilizando dois *joysticks*.
- VN (Verdadeiros Negativos): Movimentos produzidos do braço robótico diferente em relação à mesma tarefa utilizando dois *joysticks*.

O primeiro participante obteve 8 VPs e o segundo obteve 9 VPs. Temos então:

$$prec = \frac{17}{17 + 3} \tag{2}$$

Deste modo, a precisão de 0.85 nesse cenário realista é considerada ótima. Deste modo, o resultado para esse experimento foi considerado aceitável e então foi decidido que essa primeira versão da ferramenta apoia o ensino-aprendizagem da robótica educacional, com grande potencial de contribuição ao exercício do pensamento computacional em escolas do ensino básico, trazendo boas contribuições para o avanço na fronteira de conhecimento nessa área da educação. Em ambos os experimentos, tanto em testes em laboratório, quanto durante a participação dos alunos, foi possível realizar todos as tarefas definidas, obedecendo quase tudo o que foi programado através dos blocos. Foi possível perceber também que os blocos são fáceis de manusear, o que é uma característica bastante aceita por estudantes, comparado às construções de uma linguagem de programação convencional. Algumas características se destacam na ferramenta, como a rapidez na realização de todas as suas funcionalidades e principalmente na geração automática de códigos e corretude comparado com o método tradicional.

Há melhorias que podem ser feitas no bloco de abertura e fechamento da garra robótica, pois os alunos ainda tiveram um pouco de dificuldade em saber quantos graus seriam necessários para a garra encaixar perfeitamente no objeto, sem apertar muito ou afrouxar o fechamento, tal ação é muito mais fácil com a utilização de *joysticks*.

Além do mais, com o decorrer desse projeto, desde as pesquisas na fase de levantamento bibliográfico ao desenvolvimento da ferramenta proposta, foi notório perceber o poder que as linguagens visuais podem agregar a educação, ensino e aprendizagem no contexto da internet das coisas. Partindo desse mesmo princípio, assim como foi possível gerar código para um microcontrolador Arduino UNO, é possível

também programar outros componentes como por exemplo um Raspberry Pi ou ESP8266.

VI. Considerações Finais e Trabalhos Futuros

O projeto UpRobotics foi desenvolvido com o intuito de promover o exercício do pensamento computacional dos alunos, aplicando o ensino de ângulos matemáticos presente na geometria e com o auxílio da robótica educacional por meio de uma ferramenta gráfica de programação, a qual utiliza como mecanismo principal o conceito de blocos gráficos arrastáveis, onde toda a lógica de programação torna-se transparente para o usuário, pois a programação passa a ser a ligação adequada dos blocos funcionais.

O apelo visual e o encaixe correto das peças induz o aluno a percepção das ligações corretas das partes da programação do início ao fim do programa, facilitando a manipulação dos elementos da lógica de programação. Os resultados preliminares mostraram ser satisfatórios, visto que essa primeira versão da ferramenta possibilita a geração automática de código embarcável em um Arduino suficiente para o controle de movimentos do braço robótico. Deste modo, esse trabalho tem potencial para dar subsídios a um estudo mais aprofundado sobre ensino de robótica, dando base a trabalhos vindouros com experimentos de campo aplicados a um número maior de alunos.

Como trabalhos futuros, a ferramenta tem potencial de expandir suas funcionalidades com a inclusão de outros componentes relacionados à matemática, internet das coisas e robótica. Por exemplo, pode haver a inclusão de sensores e atuadores, outros microcontroladores, conceitos de outras áreas da ciência e educação, poderá servir de base para a elaboração de outros experimentos cujos os resultados poderão servir de base para construir para o avanço da fronteira de conhecimento nesse ramo científico.

Agradecimentos

Esta pesquisa, conforme previsto no Art. 48 do decreto nº 6.008/2006, foi financiada pela Samsung Eletrônica da Amazônia Ltda, nos termos da Lei Federal nº 8.387/1991, através de convênio nº 003, firmado com o ICOMP/UFAM. Este trabalho foi parcialmente financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior e Fundação de Amparo à Pesquisa do Estado do Amazonas (Edital Universal 002/2018).

Referências

[1] B. Miller, A. Kirn, M. Anderson, J. C. Major, D. Feil-Seifer e M. Jurkiewicz, "Unplugged Robotics to Increase K-12 Students' Engineering Interest and Attitudes,"2018 IEEE Frontiers in Education Conference (FIE), San Jose, CA, USA, 2018, pp. 1-5, doi: 10.1109/FIE.2018.8658959.

[2] J. Vidal, F. Sampaio e A. Pontes. "DuinoBlocksII: Uma ferramenta de apoio ao ensino-aprendizagem de lógica de programação e robótica educacional para alunos da rede pública do Ensino Médio"8th Workshop of Robotics in Education, Curitiba-PR - Brazil, November, 08-10, 2017

[3] A. Gerosa, V. Koleszar, L. Gómez-Sena, G. Tejera e A. Carboni, "Educational Robotics and Computational Thinking Development in Preschool,"2019 XIV Latin American Conference on Learning Technologies (LACLO), San Jose Del Cabo, Mexico, 2019, pp. 226-230, doi: 10.1109/LACLO49268.2019.00046.

[4] T. Nakamura e T. Kawasaki, "Computer Science Unplugged for Developing Computational Thinking and Mathematical Thinking,"2019 International Joint Conference on Information, Media and Engineering (IJCIME), Osaka, Japan, 2019, pp. 305-308, doi: 10.1109/IJCIME49369.2019.00108.

[5] S. Swaid e T. Suid, "Computational Thinking Education: Who Let the Dog Out?,"2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2019, pp. 788-792, doi: 10.1109/CSCI49370.2019.00150.

[6] J. Piaget (1973). To understand is to invent: The future of education.

[7] S. Papert (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc.

[8] S. May. 2017. What Is Robotics?. Disponível em: <https://www.nasa.gov/stem/forstudents/5-8/index.html>. Acesso em: 7 ago. 2020.

[9] D. Jurafsky e J. Martin. Speech and Language Processing. 3. ed. [S.l.: s.n.], 2018. 26, 30, 33, 36

[10] S. Sales, R. Silva, E. Sobreira e M. Nascimento. "Utilizando Scratch e Arduino como recursos para o ensino da Matemática". VI Congresso Brasileiro de Informática na Educação (CBIE, 2017). Em: Anais do XXVII Workshop de Informática na Escola (WIE 2017) pp. 538-547, 2017.

[11] E. Pasternak, R. Fenichel e A. N. Marshall, "Tips for creating a block language with blockly,"2017 IEEE Blocks and Beyond Workshop (B&B), Raleigh, NC, 2017, pp. 21-24, doi: 10.1109/BLOCKS.2017.8120404.

[12] R. E. Patiño-Escarcina, D. Barrios-Aranibar, L. S. Bernedo-Flores, P. Javier Alsina e L. M. Garcia Gonçalves, "EDUROSC-Kids: An Educational Robotics Standard Curriculum for Kids,"2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE), Rio Grande, Brazil, 2019, pp. 471-476, doi: 10.1109/LARS-SBR-WRE48964.2019.00089.

[13] Hopcroft, J. E., Ullman, J. D. e Motwani, R. (2002). Introdução à teoria de autômatos, linguagens e computação. Rio de Janeiro: Campus.