



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
UNIDADE ACADÊMICA DE SERRA TALHADA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ALAN LIMEIRA BRITO

**Desenvolvimento de um Ambiente Educacional
para Auxiliar o Ensino de Lógica de
Programação: uma abordagem com arduino,
blocos de montar de plástico interligados e
programação em blocos.**

Serra Talhada,
Dezembro/2019

Alan Limeira Brito

**Desenvolvimento de um Ambiente Educacional
para Auxiliar o Ensino de Lógica de
Programação: uma abordagem com arduino,
blocos de montar de plástico interligados e
programação em blocos.**

Trabalho de Conclusão de Curso apresentado ao
Curso de Bacharelado em Sistemas de Informação
da Unidade Acadêmica de Serra Talhada da
Universidade Federal Rural de Pernambuco como
requisito parcial à obtenção do grau de Bacharel.

Orientador: Prof. Dr. Richarlyson Alves D'Emery

Serra Talhada,
Dezembro/2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

- B862d Brito, Alan Limeira
Desenvolvimento de um ambiente educacional para auxiliar o ensino de lógica de programação: uma abordagem com arduino, blocos de montar de plástico interligados e programação em blocos / Alan Limeira Brito. - 2019.
57 f. : il.
- Orientador: Richarlyson Alves D D Emery.
Inclui referências e apêndice(s).
- Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco, Bacharelado em Sistemas da Informação, Serra Talhada, 2020.
1. Programação em blocos. 2. Lógica de programação. 3. Blocos de montagem. 4. Robótica educacional. I. D Emery, Richarlyson Alves D, orient. II. Título

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
UNIDADE ACADÊMICA DE SERRA TALHADA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ALAN LIMEIRA BRITO

Desenvolvimento de um Ambiente Educacional para Auxiliar o Ensino de Lógica de Programação: uma abordagem com arduino, blocos de montar de plástico interligados e programação em blocos.

Trabalho de Conclusão de Curso julgado adequado para obtenção do título de Bacharel em Sistemas de Informação, defendida e aprovada por unanimidade em 19/12/2019 pela banca examinadora.

Banca Examinadora:

Prof. Dr. Richarlyson Alves D'Emery
Orientador
Universidade Federal Rural de Pernambuco

Prof. M.e Glauber Magalhães Pires
Universidade Federal Rural de Pernambuco

Prof. M.e Hidelberg Oliveira Albuquerque
Universidade Federal Rural de Pernambuco

DEDICATÓRIA

Dedico este trabalho aos meus pais, Isaac Amaral de Brito e Socorro Limeira Brito, e a meu avô Amaro Francisco do Amaral (*In Memoriam*).

AGRADECIMENTOS

Agradeço, primeiramente, a Deus por proporcionar a realização deste sonho. Agradeço aos meus pais, Isaac Amaral de Brito e Socorro Limeira Brito, por serem meu alicerce; aos meus irmãos: Alexandro Limeira e Aline Limeira. Agradeço também a Rafaela Mendes por todo apoio, incentivo e companheirismo e a Dona Lindalva Mendes pela motivação e por sempre ascender a minha fé. E a toda minha família e amigos.

Agradeço, imensamente, a meu orientador Richarlyson Alves D'Emery, por acreditar em mim e me fazer persistir, colaborando com seu conhecimento, até mesmo quando o tempo era escasso, com paciência e dedicação. Vindo a ficar na memória como ótimo orientador e excelente professor.

Agradeço a todos os professores que fizeram parte da minha jornada acadêmica, em especial aos professores Glauber Magalhães e Hidemberg Oliveira, os quais marcaram de forma significativa essa trajetória.

Agradeço, por fim, a todos que contribuíram, de forma direta ou indireta, para a finalização deste trabalho.

Muito Obrigado!

“O sucesso é ir de fracasso em fracasso sem perder o entusiasmo”.

Winston Churchill

RESUMO

Atualmente, as escolas passam por transformações em seus métodos de ensino, algumas ainda utilizam estratégias tradicionais, por outro lado, outras procuram atualizar-se. A tecnologia nas escolas pode desenvolver novas competências, como o raciocínio lógico, o que beneficiaria os alunos tanto no âmbito escolar quanto em relação a atividades cotidianas e em sua vida profissional. O ensino de lógica de programação já faz parte do currículo básico de vários países, que estão atentos à importância da lógica como base para o desenvolvimento das crianças. Na sociedade da informação, o uso de tecnologias como ferramenta de auxílio a professores vem ganhando notoriedade, nesse sentido, a Robótica Educacional passa a ser um meio alternativo para o ensino e, conseqüentemente, pode melhorar o desempenho de crianças no estudo de lógica de programação. É nesse contexto que se decorre esta pesquisa, voltada para o desenvolvimento de um ambiente educacional para o ensino de lógica de programação. O ambiente desenvolvido usa como método a programação via código baseada na linguagem JavaScript e a programação em blocos baseada em Blockly para acionar um *hardware* desenvolvido a partir da combinação de blocos de montar de plástico interligados e componentes eletrônicos acionados por Arduino. A interface eletrônica é encapsulada de maneira a dirimir eventuais complexidades. Destarte, como contribuição principal, este trabalho traz o desenvolvimento de uma solução para auxiliar professores e alunos no processo de ensino e aprendizagem da lógica de programação, principalmente pelo fato de tratar-se de uma solução de baixo custo que pode ser utilizada em cenários domésticos e/ou em ambiente escolar.

Palavras-chave: Ambiente Educacional, Aprendizagem, Lógica de Programação, Robótica Educacional, Blocos de Montagem, Programação em Blocos.

ABSTRACT

Currently, as schools undergo transformations in their teaching methods, some still use traditional tools, on the other hand, other updated updates. Technology in schools can develop new skills, such as logical thinking, or who welcomes students both at school level and in relation to everyday activities and their working life. The teaching of programming logic is already part of the basic curriculum of many countries, which are related to the importance of logic as a basis for the development of children. In the information society, the use of technologies as a tool to help teachers has gained notoriety, in this sense, Educational Robotics becomes an alternative means for teaching and, consequently, can improve the performance of children in the study of programming logic. . It is in this context that we decorate this research, focused on the development of an educational environment for the teaching of programming logic. The developed environment uses the JavaScript code programming method and block programming, which are used in blocks to drive hardware developed from the combination of interlocking plastic assembly blocks and Arduino-powered electronics. An electronic interface is encapsulated to address any complexities. Thus, as a major contribution, this work brings the development of a solution for teachers and students assisting in the process of teaching and learning the logic of programming, mainly because it is a low cost solution that can be used in work situations and / or in a school environment.

Keywords: Educational Environment, Learning, Programming Logic, Educational Robotics, Assembly Blocks, Block-based Programming

LISTA DE FIGURAS

Figura 2.1 – Resultado de uma sequência de comandos no SuperLogo	19
Figura 2.2 – Laços de repetição no Scratch	20
Figura 3.1 – Layout do Ambiente DuinoBlocks	22
Figura 3.2 – Layout da aplicação Raspiblocos	23
Figura 4.1 – Tela Inicial ProBlockly	28
Figura 4.2 – Tela principal do <i>software</i> ProBlockly	28
Figura 4.3 – Área de trabalho ProBlockly	29
Figura 4.4 – Área de exibição e edição de código	30
Figura 4.5 – Área de simulação para acionamento de: (a) LEDs (SimLed); e (b) servo motor (SimServo)	30
Figura 4.6 – Diagrama de comunicação do <i>software</i> ProBlockly	32
Figura 4.7 – Exemplo de bloco	33
Figura 4.8 – Código para criação de um bloco	34
Figura 4.9 – Função que retorna o código do bloco	34
Figura 4.10 – Código de exemplo para criação da categoria Lógica	35
Figura 4.11 – Exemplo de caixa de ferramentas	36
Figura 4.12 – Microcontrolador Arduino encapsulado: (a) aparência externa; (b) aparência interna	37
Figura 4.13 – Servo Motor integrado a um bloco de montar de plástico interligável	38
Figura 4.14 – Sensor ultrassônico integrado a um bloco de montar de plástico interligável	38
Figura 4.15 – LED integrado a um bloco de montar de plástico interligável	39
Figura 4.16 – Simulação de uma garagem construída no ambiente proposto	39
Figura 4.17 – Programação em blocos da simulação de uma garagem construída no ambiente proposto	40
Figura 4.18 – JavaScript da simulação de uma garagem construída no ambiente proposto	40
Figura A.1 – Telas do <i>software</i> do ProBlockly: (a) menu principal expandido; (b) opção experimente expandida; (c) opção salvar; (d) opção abrir arquivo; e (e) opção configuração.	49
Figura B.1 – Opções do menu de componentes eletrônicos: (a) LED; (b) Servo Motor; (c) Motor DC; (d) Ultrassom; (e) Pinos; e (f) Infravermelho	52
Figura C.1 – Programação em blocos e JavaScript para acionamento de um LED conectado ao Arduino	55
Figura C.2 – Programação em blocos e JavaScript para acionamento de LEDs a partir da leitura de um ultrassom conectado ao Arduino	55
Figura C.3 – Programação em blocos e JavaScript para acionamento de um servo motor conectado ao Arduino	56
Figura C.4 – Programação em blocos e JavaScript para acionamento dos LEDs do Simulador SimLed	56
Figura C.5 – Programação em blocos para acionamento do servo motor do Simulador SimServo	57

LISTA DE QUADROS

Quadro 3.1 – Análise comparativa com trabalhos relacionados

25

LISTA DE TABELAS

Tabela 4.1 – Investimento financeiro para construção do *hardware* do ambiente ProBlockly

41

LISTA DE ABREVIATURAS E SIGLAS

CSS	<i>Cascading Style Sheets</i>
HTML	<i>HyperText Markup Language</i>
IDE	<i>Integrated Development Environment</i>
IoT	<i>Internet of Things</i>
RE	Robótica Educacional
TI	Tecnologia da Informação
XML	<i>Extendible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Contextualização	13
1.2	Problema, Motivação e Justificativa	15
1.3	Objetivos	15
1.4	Estrutura do Documento	16
2	REFERENCIAL TEÓRICO	17
2.1	Educação e Tecnologia	17
2.1.1	Ensino de Lógica nas Escolas	18
2.2	Robótica Educacional	20
2.3	Considerações	21
3	TRABALHOS RELACIONADOS	22
3.1	Estudo Comparativo	24
4	AMBIENTE PROPOSTO	27
4.1	Introdução	27
4.2	Ambiente ProBlockly	27
4.2.1	Blocos Lógicos	33
4.2.2	Menu Lateral ou Caixa de Ferramentas	34
4.2.3	O Hardware	36
4.3	Considerações	41
4.4	Dificuldades Encontradas	42
5	CONCLUSÃO	44
5.1	Considerações Finais	44
5.2	Contribuições deste Trabalho	45
5.3	Proposta para Trabalhos Futuros	45
	REFERÊNCIAS	46
	APÊNDICE A – TELAS DO PROBLOCKLY	49
	APÊNDICE B – COMANDOS RELACIONADOS AOS COMPONENTES ELETRÔNICOS DISPONIBILIZADOS NA VERSÃO 1.0 DO PROBLOCKLY	52
	APÊNDICE C – EXEMPLOS DE PROGRAMAÇÃO EM BLOCOS NO PROBLOCKLY	55

1 Introdução

Neste capítulo é apresentada a motivação desta monografia. Na Seção 1.1 expõe-se brevemente uma introdução ao cenário atual da educação no ensino de lógica de programação. Na Seção 1.2 é apresentado problema, motivação e justificativa deste trabalho. Na Seção 1.3 demarcam-se os objetivos deste trabalho. E na Seção 1.4 é fornecida uma visão dos capítulos da monografia.

1.1 Contextualização

Há alguns anos, estudiosos vêm colocando em debate a discussão sobre a integração de Tecnologias da Informação e Comunicação (TICs) nas escolas. O uso destas tecnologias melhora a qualidade do ensino e torna a aprendizagem dos alunos mais motivadora (MIRANDA; SAMPAIO; BORGES, 2011). De acordo com os autores, percebe-se a importância de um ensino diversificado e inovador, o qual facilitará a assimilação dos conteúdos pelos educandos.

Além disso, é evidente a evolução da Informática nos últimos anos e a carência de profissionais na área vem aumentando consideravelmente (BEZERRA; DIAS, 2012 apud GARLET; BIGOLIN; SILVEIRA, 2016). Contudo, para o ingresso nesta área é preciso conhecimento de lógica de programação, o que demanda dedicação (PEREIRA; JOSÉ; RAPKIEWICZ, 2004 apud GARLET; BIGOLIN; SILVEIRA, 2016). Com isso, é perceptível o empenho de especialistas nessa área e a imprescindibilidade de estímulos para incentivá-los.

Tal preocupação não é recente, Moran (2000) já salientava que é preciso unificar as dinâmicas tradicionais e as inovadoras, pois as tecnologias contribuem na formação de um trabalho conjunto entre professor e aluno, o acesso a essas novas tecnologias deve ser frequente e para isto é preciso torná-lo viável, proporcionando um ensino mais produtivo.

Na educação básica, é importante o uso de estratégias para facilitar a aquisição, armazenamento ou também a utilização da informação. Alguns estudos mostram que o baixo desempenho escolar pode estar associado à má utilização de estratégias de aprendizagem (INÁCIO; OLIVEIRA; MARIANO, 2017). Juntamente a isso, os recursos tecnológicos têm

sido utilizados cada vez mais em salas de aula, desta forma, mostra-se necessário a utilização de ferramentas estratégicas no ensino infantil.

O uso de TICs para auxílio em sala de aula, bem como o ensino de lógica de programação, está aumentando exponencialmente no Brasil. A programação pode ajudar no desenvolvimento de habilidades valiosas na atualidade, como o trabalho em equipe, a autonomia e a criatividade (NOVA ESCOLA, 2016).

O ensino de lógica de programação também pode ajudar a criança com dificuldades em outras disciplinas, pois com este conteúdo ela pode desenvolver o pensamento computacional, melhorando o raciocínio lógico e facilitando a resolução de problemas (GARLET; BIGOLIN; SILVEIRA, 2016).

Apesar do Brasil ainda não possuir a obrigatoriedade deste tipo de conteúdo nos Parâmetros Curriculares Nacionais (BRASIL, 1997) e nas Diretrizes Curriculares Nacionais Gerais da Educação (BRASIL, 2013), diversos países já têm em seu currículo básico o ensino de programação. Segundo Garlet, Bigolin e Silveira (2016), no Reino Unido, as aulas de programação são obrigatórias para as crianças. E iniciativas como esta estão sendo discutidas e implementadas, a exemplo dos Estados Unidos da América que está atualizando seus currículos com a disciplina de programação no ensino básico (RICHTEL, 2014).

Nesse novo cenário, o ensino de lógica de programação desperta o interesse do aluno por outros tipos de tecnologias e facilita a utilização de novas ferramentas nas escolas, como, por exemplo, o uso da Robótica na Educação. A Robótica na Educação vem ganhando espaço nos últimos anos, embora sua aplicação com crianças ainda seja pouco explorada, ela possui um potencial como ferramenta interdisciplinar, em seu uso o aluno pode adquirir vários conhecimentos que, comumente, não são ensinados nas disciplinas, fazendo com que esse método prático ganhe força, sendo o elo entre o conhecimento passado em sala de aula e a resolução de problemas do mundo real presentes no dia a dia. Contudo, o acesso a esses recursos muitas vezes é tido como um grande desafio, já que a maior parte da população não possui condições de adquirir esses equipamentos devido ao alto custo de comercialização (SANTOS; NASCIMENTO; BEZERRA, 2010).

Desta forma, o desenvolvimento de *software* para fins educacionais, que faça a inserção do conteúdo de lógica de programação nas escolas, com a utilização da Robótica como método de ensino para crianças, faz com que facilite o aprendizado do aluno e esse consiga assimilar melhor a base de programação, que é uma necessidade cada vez maior, já que é escasso o número de profissionais nesta área.

1.2 Problema, Motivação e Justificativa

Diante da problemática de soluções que melhorem o processo de ensino e aprendizagem, em especial no âmbito da programação, impulsiona-se o desenvolvimento desta pesquisa. Assim, ela será direcionada aos estudos de práticas educacionais que auxiliem o processo de ensino e aprendizagem de lógica de programação.

O cenário atual da Tecnologia da Informação (TI) apresenta dificuldades na formação de profissionais, o que está relacionado, em grande parte, a falta de preparação prévia na educação básica. O jovem ingressante do ensino superior, quando criança, geralmente não apresenta conhecimentos básicos sobre TI, no que diz respeito à programação, ou não.

Além disso, o custo de kits de Robótica Educacional (RE) comerciais tem preços elevados, fazendo com que se procurem soluções mais acessíveis economicamente.

Devido à dificuldade no ensino de crianças e a necessidade de meios que facilitem o acesso à informação de forma inclusiva, torna-se mais notável o aumento de pesquisas sobre métodos que auxiliem o professor a promover um ensino de qualidade. Tendo em vista que muitas dessas dificuldades de aprendizagem têm a atenção como uma de suas principais causas, e sabendo que muitos alunos se sentem desmotivados com os métodos tradicionais de ensino empregados nas escolas, é importante a análise e desenvolvimento de ferramentas que ajudem a passar o conhecimento de forma lúdica.

Assim, essa pesquisa se justifica através do desenvolvimento de um ambiente educacional para auxiliar o ensino de lógica para crianças, em especial de programação, agregando valor tanto para o meio acadêmico, reforçando e contribuindo com o estudo e pesquisa na área de Educação, quanto para o meio social, esperando a melhora da qualidade do processo de ensino e aprendizagem.

1.3 Objetivos

Como objetivo geral, pretende-se desenvolver um ambiente educacional para auxiliar o processo de ensino e aprendizagem de lógica de programação para crianças.

Para alcançar esse objetivo, têm-se os seguintes objetivos específicos:

- Analisar a aplicação de kits educacionais como metodologia no auxílio de ensino de lógica de programação;
- Integrar componentes eletrônicos em peças de blocos de montar de plástico interligados;
- Desenvolver um módulo utilizando Arduino para acionar os componentes eletrônicos;
- Desenvolver um *software* baseado em programação em blocos com suporte aos componentes eletrônicos integrados; e
- Integrar os objetivos anteriores.

1.4 Estrutura do Documento

Esta monografia está estruturada da seguinte forma:

- Capítulo 1: Apresenta-se a contextualização da situação atual da educação e os desafios de implementar novas formas de ensino utilizando lógica de programação;
- Capítulo 2: Discute-se a utilização de RE no ensino de lógica de programação nas escolas;
- Capítulo 3: Apresentam-se algumas pesquisas relacionadas à RE no ensino de lógica de programação;
- Capítulo 4: Apresentam-se os métodos e ferramentas utilizadas; e
- Capítulo 5: Têm-se as considerações finais do presente trabalho.

2 Referencial Teórico

Neste capítulo é apresentada uma breve explanação sobre a importância da tecnologia para aquisição de conhecimento. Na Seção 2.1 é apresentado um resumo sobre a necessidade de integração entre Educação e Tecnologia. Na Seção 2.2 são descritos os benefícios do uso de robótica educacional nas escolas. Na Seção 2.3 são feitas algumas considerações acerca do capítulo.

2.1 Educação e Tecnologia

A Educação é fundamental para o desenvolvimento de uma sociedade e tem sido assegurada como direito a todos os cidadãos. Embora haja diversas pesquisas que melhorem a educação, ela passa por uma reforma em questão didático-pedagógico e organização curricular, favorecida pelos modelos e conteúdos diversificados.

A Internet propicia o acesso aos fatos, conteúdos e acontecimentos, tornando urgente a integração das TICs no currículo escolar, já que o uso de tecnologias como ferramentas pedagógicas possibilita o desenvolvimento de novas competências, como a informação e o pensamento em rede (LÉVY, 1993 apud TEZANI, 2011).

Segundo Oliveira, Libâneo e Toschi (2017), no capítulo “A educação escolar no contexto das transformações da sociedade contemporânea”, a sociedade contemporânea está sendo denominada por estudiosos como sociedade do conhecimento, em que o conhecimento e a ciência tornam-se de fundamental importância na atualidade, as pessoas podem adquirir informações de diversas fontes como televisão, computador, rádio, entre outros, nesse sentido a escola não é o único meio de se conseguir conhecimento.

A tecnologia influencia e é influenciada pela sociedade, o que representa para as escolas uma exigência para mudanças em suas políticas, adaptando seus currículos e práticas para ampliar as chances de inserção dos alunos na sociedade. Embora exista uma necessidade de adaptação das escolas com esta realidade, também se vê necessário um olhar mais aprofundado com relação às quais tecnologias serão usadas, como o professor as utiliza e se está preparado para usá-las (LÉVY, 1999 apud TEZANI, 2011).

2.1.1 Ensino de Lógica nas Escolas

As crianças de hoje, crescem imersas em novas tecnologias, contribuindo para o ensino de lógica de programação, o que facilita o uso das TICs em sala de aula, posto que já possuem um certo conhecimento no manuseio destas ferramentas.

Nessa nova realidade, onde se mostra cada dia mais primordial a utilização da tecnologia na educação como ferramenta auxiliadora, o ensino de lógica pode ser uma ponte para implementar a aplicação de TIC nas escolas. É importante que as pessoas tenham uma boa base sobre conhecimentos computacionais e pela dificuldade envolvida neste aprendizado, até mesmo no ensino superior, incentivar o ensino desde cedo pode ajudar os alunos que decidirem seguir na área, ao mesmo tempo os que não seguirem, a adquirir essa competência, que no futuro pode ajudar os indivíduos a solucionarem os problemas com mais facilidade (JÚNIOR et al., 2005 apud OLIVEIRA et al., 2014).

Pensando nisso, vários países já se anteciparam e estão adicionando a programação nos currículos escolares, como, por exemplo, no Reino Unido as aulas de programação já são obrigatórias (GARLET; BIGOLIN; SILVEIRA, 2016). O ensino de lógica de programação pode facilitar o uso das TICs na educação, e introduzir novos métodos de ensino, como a RE.

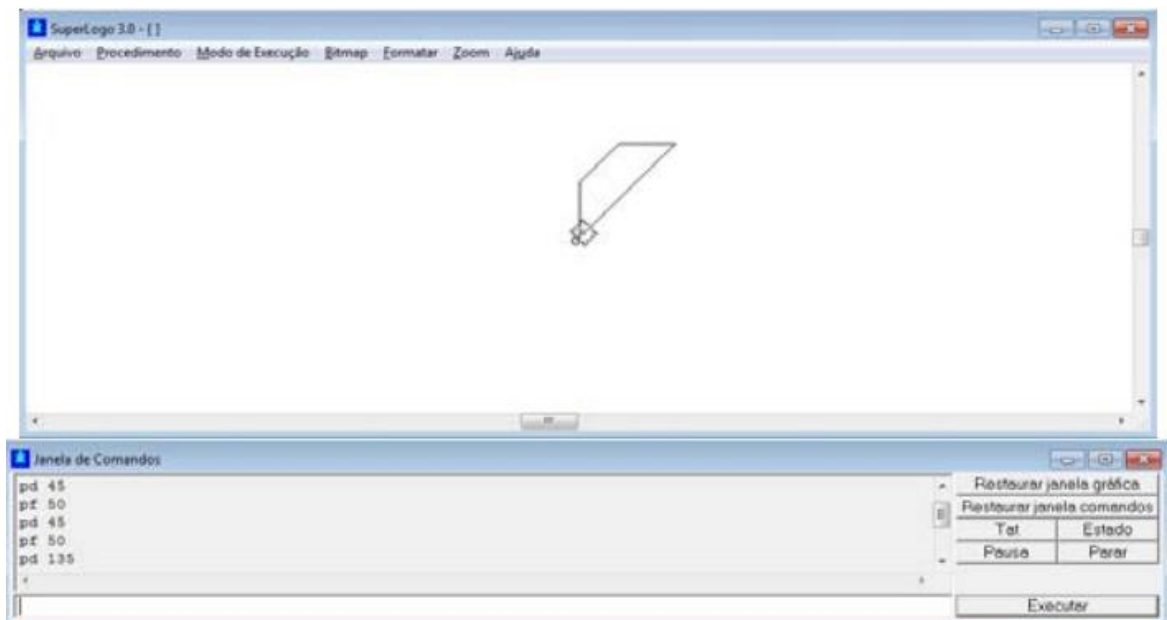
Diante dos benefícios que o ensino de lógica de programação proporciona, destaca-se a melhora do desempenho dos alunos em outras disciplinas, pois a lógica é considerada uma matéria interdisciplinar (PUGA; RISSETTI, 2009 apud SILVA et al., 2017).

Existem projetos para o incentivo a programação, a exemplo da linguagem de programação LOGO, a qual não necessita de uma base de conhecimento para programá-la, considerada uma linguagem de fácil entendimento que proporciona ao aluno satisfação para continuar o estudo de programação (MIRANDA; PEREIRA; RISSETTI, 2016).

Ademais, LOGO é utilizada, principalmente, para iniciantes na programação, devido a simplicidade de seus comandos. O ambiente SuperLogo é utilizado no ensino da Matemática, por exemplo, onde o aluno pode criar formas geométricas com comandos para movimentação de uma tartaruga, que desenha retas de acordo com as instruções passadas. O SuperLogo é simples e possui comandos básicos como “PARATRAS” (ou equivalente PT), “PARAFRENTE” (ou PF), “PARADIREITA” (ou PD) e “PARAESQUERDA” (ou PE); os comandos PF e PT fazem a tartaruga se movimentar, enquanto que PD e PE fazem a mesma girar em torno do seu eixo (OLIVEIRA; MARCELINO, 2015). Mesmo sendo utilizado na Matemática, o SuperLogo trabalha de forma imperceptível conceitos de programação.

A Figura 2.1 ilustra uma sequência de comandos sendo executados pelo SuperLogo, resultando uma forma geométrica. Conclui-se então, que a criança deve estabelecer uma sequência de passos para chegar ao resultado esperado, ensinando implicitamente, o básico sobre programação. Por exemplo, para se desenhar um quadrado a criança poderia utilizar a sequência de comandos: PF 100, PD 90, PF 100, PD 90, PF 100, PD 90, PF 100 e PD 90. Nessa codificação observam-se o tamanho do quadro e os ângulos internos, através do comando PF 100 desenha-se uma linha reta para frente de tamanho 100, enquanto que PD 90 faz a tartaruga girar 90 graus para direita. Mas, existe uma sequência lógica de repetição e a criança poderia pensar em outra codificação para obter o mesmo resultado, ou seja, repita 4 vezes [PF 100 PD 90].

Figura 2.1 – Resultado de uma sequência de comandos no SuperLogo



Fonte: Oliveira; Marcelino (2015)

Outro projeto para ensino de lógica de programação é o Scratch, uma linguagem gráfica baseada na linguagem LOGO (OLIVEIRA et al., 2014). Ainda, segundo os autores, o ambiente Scratch proporciona uma interação diversificada, no qual o aluno pode criar animações e histórias, por exemplo.

A Figura 2.2 apresenta uma animação feita no ambiente Scratch. O aluno monta a sequência lógica através de blocos de comando e faz a interação com imagens, associando cada componente de imagem a comandos específicos. A interação é realizada graficamente, a criança faz toda a programação sem a necessidade de digitação de código e conhecimentos de linguagens de programação.

Figura 2.2 – Laços de repetição no Scratch



Fonte: Oliveira et al. (2014)

2.2 Robótica Educacional

A Robótica e a tecnologia, quando utilizadas na educação, podem alcançar resultados considerados melhores que aulas de exposição de conteúdo, tendo em vista que essa metodologia busca estimular o interesse dos alunos, consequentemente melhorando seu aprendizado (PAPERT, 1980 apud HEINEN et al., 2015).

Por meio da Robótica, os alunos podem desenvolver várias habilidades e competências como a formulação de testes de hipóteses e o raciocínio lógico, além de ensinar de forma lúdica, pois o aluno pode manusear, construir, verificar o que está errado e consertar, assim o aluno se sente mais estimulado com o ensino (ALVES; SAMPAIO; ELIA, 2013).

Alguns dos conteúdos trabalhados em sala são difíceis para os alunos aprenderem através de aulas teóricas, contudo a utilização da RE permite que as crianças possam aplicar os conhecimentos adquiridos em sala de forma prática (ZILLI, 2004). Nesse sentido, estudos crescem a cada dia e a demanda por novas tecnologias nesta área também. No mercado já existem várias soluções, tanto nacionais como internacionais, dentre os projetos brasileiros, estão o Super Robby e o Robótica Fácil, já no contexto internacional destaca-se o Lego MindStorms (SASAHARA; CRUZ, 2007).

Ainda que diante de tantos benefícios que a Robótica no âmbito educacional pode proporcionar, os custos de produção, principalmente no contexto comercial, ainda são consideravelmente elevados. Posto isso, aumentou-se a procura por alternativas que possam viabilizar o uso dessas tecnologias em sala de aula (ALVES; SAMPAIO; ELIA, 2013).

O projeto Arduino é exemplo de alternativa, disponibilizando uma plataforma de *software* e *hardware* de código aberto e que pode ser utilizado como equipamento de baixo custo na RE.

2.3 Considerações

O mundo está se atualizando, os países desenvolvidos estão preparando suas crianças para que elas não sejam apenas futuros consumidores de tecnologias, mas que possam também criá-las e desenvolvê-las. Para isto, a adequação dos currículos escolares, adicionando-se conteúdos de programação, torna-se necessária.

O uso da Robótica em salas de aula pode estimular o aprendizado do aluno e motivá-lo, apesar disso, a RE ainda é pouco explorada. Isto se explica pelo fato das pesquisas nesta área serem relativamente novas e pelo elevado custo financeiro atrelado.

Uma possível solução é o uso da Robótica de baixo custo, que pode tornar viável seu uso nas escolas. Essa ideia pode ser viabilizada levando em consideração a possibilidade de integração de componentes eletrônicos de baixo custo inseridos no contexto de blocos de montar de plástico interligados. Esse *hardware* associado a interfaces de *software* de fácil usabilidade poderá favorecer um ambiente lúdico, em que o ensino de lógica de programação é ensinado por meio de comandos de programação, mas sem a abstração requerida em linguagens de programação oriundas de cursos de Computação.

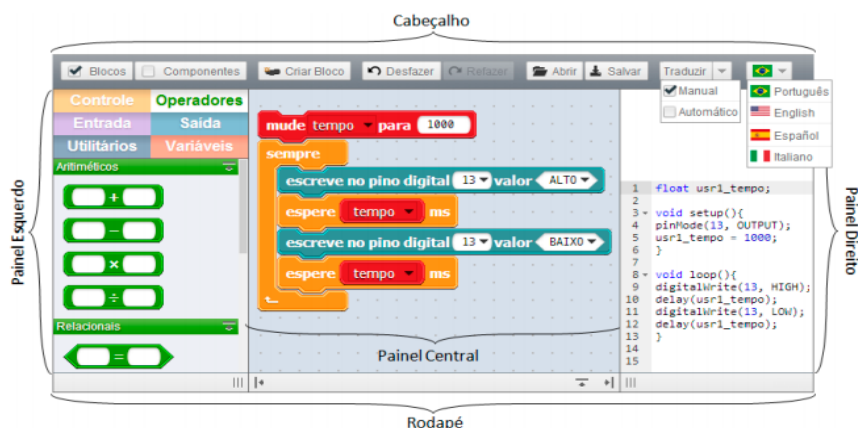
3 Trabalhos Relacionados

Neste capítulo é apresentado os trabalhos relacionados que serviram de base para desenvolvimento desta monografia. Na Seção 3.1 é apresentado o trabalho proposto.

Neste capítulo, apresentam-se algumas pesquisas presentes na literatura direcionada aos estudos da RE.

Na pesquisa intitulada "**DuinoBlocks: Desenho e implementação de um ambiente de programação visual para robótica educacional**", Alves, Sampaio e Elia (2013) implementaram um ambiente de programação, chamado DuinoBlocks, que apresenta como objetivo facilitar a programação de equipamentos de Robótica, como o Arduino, entre alunos e professores com pouco conhecimento na área. Ele possui um método de programação baseado em uma interface com blocos de comandos, em que cada bloco está dividido nas categorias relacionadas a controle, operadores, entrada, saída, unitários e variáveis. Também possui a função de recursividade, permitindo a criação e chamada de procedimentos, como podemos observar na Figura 3.1. A interação com o usuário é dada ao clicar, segurar e arrastar os componentes (blocos) para montar sequências lógicas. Neste modelo, o aluno não precisa saber comandos específicos de nenhuma linguagem.

Figura 3.1 – Layout do Ambiente DuinoBlocks

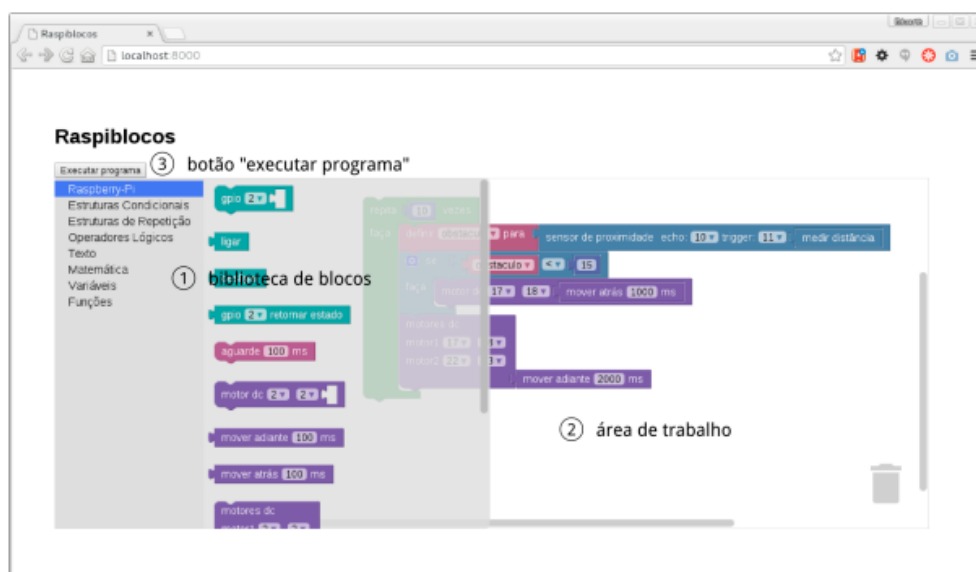


Fonte: Alves; Sampaio; Elia (2013)

Outra utilização de ambientes de programação pode ser observada no trabalho de Heinen (2015), "**Raspiblocos: ambiente de programação didático baseado em Raspberry Pi e Blockly**", no qual foi criado um ambiente de programação baseado em Raspberry Pi.

Raspberry trata-se de um minicomputador, uma placa eletrônica semelhante ao Arduino, que pode ser conectada a componentes eletrônicos como motores e LEDs, a interação se dá através de uma aplicação Web e o minicomputador pode ser conectado por Wi-fi. Foi utilizada a biblioteca de programação visual chamada Blockly, que permite que os comandos possam ser representados por blocos encaixáveis. Procurou-se utilizar equipamentos com preços mais acessíveis para que sua construção e replicação fossem facilitadas. A Figura 3.2 mostra o layout da aplicação Raspiblocos.

Figura 3.2 – Layout da aplicação Raspiblocos



Fonte: Heinen (2015)

Em “**Ambiente de robótica para aplicações educacionais com superlogo**”, Chella (2002) desenvolveu um ambiente de programação usando a linguagem de programação do SuperLogo 3.0 (ambiente ilustrado na Figura 2.1). Neste caso, a interface utiliza-se de comandos de texto para a interação com os componentes eletrônicos. A aplicação é hospedada em um servidor Web, a qual pode controlar presencialmente ou remotamente os recursos do dispositivo. Apesar de a interação ser feita por comandos de texto, a linguagem escolhida permite ao aluno aprender com mais facilidade.

No trabalho “**Ensino de computação com scratch no ensino fundamental – um estudo de caso**”, Wangenheim, Nunes e Santos (2014) buscaram a realização de um estudo de caso com crianças do ensino fundamental utilizando o Scratch, para saber quais conteúdos foram aprendidos e se os objetivos da aprendizagem foram alcançados. Os autores acreditam que a utilização do Scratch facilite o aprendizado, motive e promova uma experiência agradável para os alunos. O estudo de caso foi realizado com crianças de 6 a 7 anos de uma

escola privada, durante seis semanas. Concluíram que, em poucas aulas, as crianças conseguiram programar uma história utilizando o Scratch, bem como foi possível observar que o conteúdo de programação pode ser integrado ao currículo escolar de forma interdisciplinar, já que as aulas ocorreram em conjunto com os conteúdos de Literatura e Artes.

O kit educativo LEGO® Mindstorms foi utilizado na pesquisa de Aguiar et al. (2015) intitulada “**Introdução à Robótica e Estímulo à Lógica de Programação no Ensino Básico Utilizando o Kit Educativo LEGO® Mindstorms**”. O trabalho propôs relatar o impacto do uso da robótica educacional para estimular o ensino de lógica de programação. O kit possui um software de interface simples, que utiliza a programação através de blocos para movimentar as peças de montagem dos robôs. Para evidenciar os benefícios do uso da robótica no ensino de programação, a temática foi implementada para alunos do ensino fundamental e médio, com idades entre 14 e 15 anos. Através de questionários observou-se que houve uma melhora na resolução das questões sobre lógica após as aulas utilizando o kit educativo.

3.1 Estudo Comparativo

O presente estudo propõe a criação de um ambiente educacional, composto por *hardware* e *software*. Como visto nos trabalhos relacionados, geralmente se trabalha de maneira semelhante, ora baseados no ensino através de blocos, ora por códigos. A proposta desta pesquisa está no desenvolvimento de um produto cujo *software* combine as duas formas de programação, na qual o aluno possa escolher e visualizar ambas as formas, como, por exemplo, usando sequências de comandos, apenas clicando e arrastando blocos ou optando pela programação direta por código sem o auxílio dos blocos.

Vale ressaltar que os comandos a serem providos pelo *software* do ambiente Educacional proposto serão baseados nas atividades dos componentes de *hardware*, permitindo que o aluno visualize seu resultado diretamente no *hardware*, explorando, conseqüentemente, conceitos de variáveis, entrada e saída, laços, condicionais e outras estruturas de programação.

De maneira geral, a introdução deste método, que une a programação visual com a programação de código, procura facilitar o entendimento do aluno, pois ele pode formar sequências de blocos de comandos e ao mesmo tempo ver a ação do código, possibilitando a compreensão de conceitos da lógica de programação de maneira lúdica, tendo em vista os blocos de montar de plástico interligados.

Com relação ao *hardware*, os componentes desenvolvidos a partir da combinação de blocos de montar de plástico interligados e componentes eletrônicos, ambos de baixo custo, a exemplo de LEDs, motores DC, servo motores, emissores e receptores de ultrassom, emissores e receptores de infravermelho, entre outros.

Para o desenvolvimento da interface do *software*, foi utilizada a linguagem de programação JavaScript e preceitos da Interface Homem-Máquina, explorando características de usabilidade.

No Quadro 3.1, comparam-se as principais características entre os trabalhos relacionados e o proposto. Percebe-se que o diferencial deste trabalho está na utilização de blocos montáveis para construção de um ambiente educacional, comumente encontrado em distribuições Lego¹, no qual o aluno tem a liberdade de construir estruturas com os Legos, usando a criatividade, porém essas possuem um elevado custo comercial.

Quadro 3.1 – Análise comparativa com trabalhos relacionados

Características	DuinoBlocks	RaspBlocos	Ambiente Superlogo	Trabalho Porposto
Blocos de Comandos	x	x		x
Chamada de Procedimentos	x	x	x	x
Integração com <i>Hardware</i>	x	x	x	x
Programação por Código			x	x
Utilização de Blocos Montáveis				x

Fonte: Elaborado pelo autor

Portanto, são diversas as tentativas de introduzir métodos e ferramentas para promover o ensino de lógica de programação. Neste capítulo, foram destacadas diversas abordagens que utilizam desde código até blocos de comando, mas poucos trabalhos presentes na literatura estudada apresenta a junção dos mesmos, de maneira a facilitar o aprendizado.

Ainda, com relação aos métodos de ensino, a integração de *hardware* comumente utiliza ambientes de alto nível, a exemplo de Arduino e Raspberry Pi. E ainda que este

¹ LEGO Group. Disponível em: <<https://www.legobrasil.com.br/>>

trabalho também proponha a utilização de Arduino, este servirá apenas como plataforma de comunicação, tornando-o imperceptível quanto a sua utilização, uma vez que a interação das crianças será dada com o *software* desenvolvido e os componentes eletrônicos introduzidos em blocos de montar de plástico interligados.

4 Ambiente Proposto

Neste capítulo, apresenta-se o método e as ferramentas utilizadas no desenvolvimento desta monografia. Na Seção 4.1 apresenta-se uma introdução sobre o projeto. Na Seção 4.2 é mostrado o software desenvolvido. Na Seção 4.3 apresenta-se o protótipo de hardware. Na Seção 4.4 é dada uma visão geral sobre a aplicação. E na Seção 4.5 são abordadas as dificuldades encontradas durante o desenvolvimento.

4.1 Introdução

Como discutido na Seção 2.1, com o avanço rápido da tecnologia, competências relacionadas com a mesma se tornam cada vez mais importantes, e aqueles que as possuem, saem um passo à frente dos demais, mesmo que não estejam diretamente relacionados com a área tecnológica.

A lógica de programação estimula o desenvolvimento de várias competências, porém aprendê-la não é uma tarefa fácil, e para isso ferramentas como SuperLogo e Scratch auxiliam professores e alunos no ensino e compreensão deste tema. Como visto também, a RE contribui significativamente na educação, e a união de ferramentas de auxílio ao ensino e a RE forma um conjunto poderoso para a aprendizagem dos alunos.

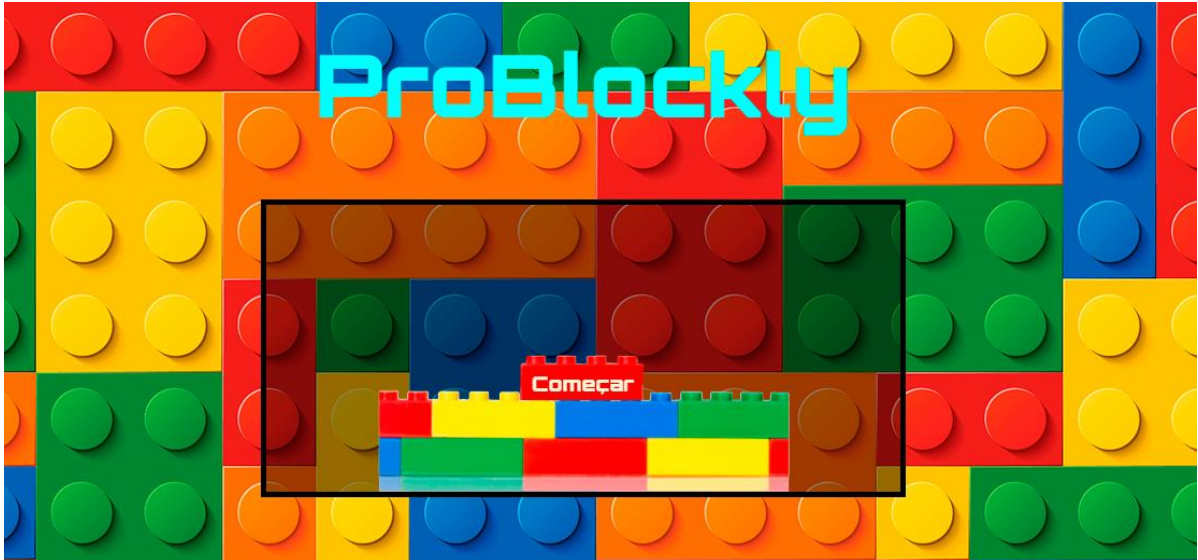
No ambiente proposto tem-se um *software* que auxilia no ensino de lógica de programação e um *hardware* composto por Arduino, componentes eletrônicos dispostos em blocos de montar de plástico interligados. O *software* deve produzir algoritmos que mostrarão os resultados através do *hardware* ou no próprio *software* através de uma área de simulação.

4.2 Ambiente ProBlockly

ProBlockly é a denominação dada ao ambiente, seu *software* foi desenvolvido com o intuito de ensinar programação com o uso de blocos de comandos, ele é dividido basicamente

em duas partes, um servidor e uma página aberta em um navegador Web. O *software* possui uma tela de inicialização como mostra a Figura 4.1.

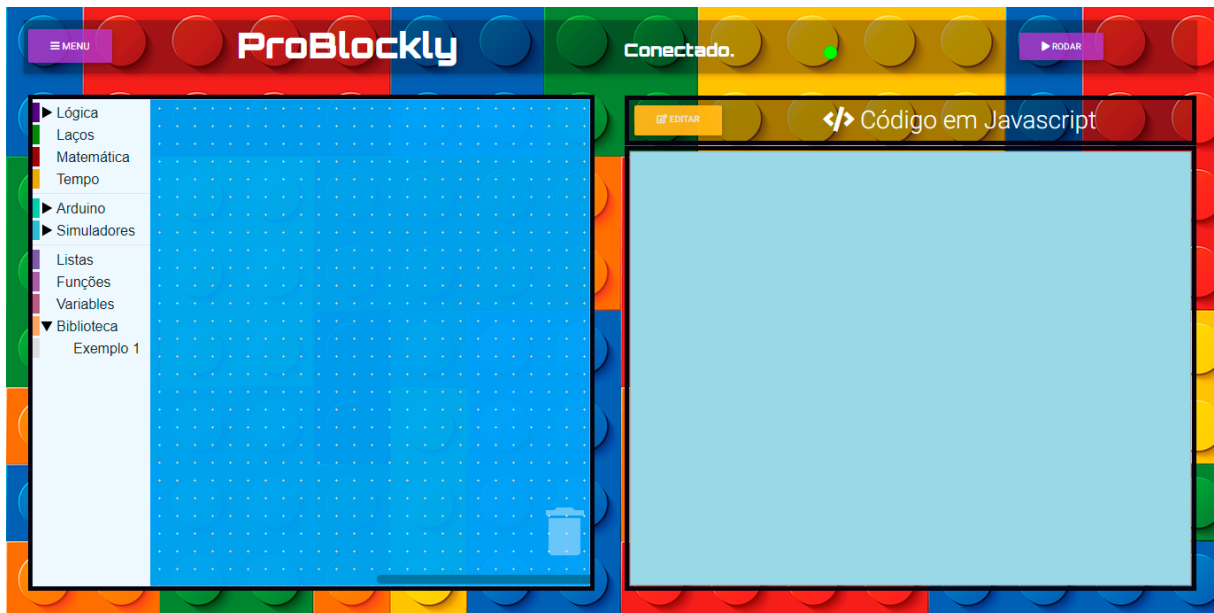
Figura 4.1 – Tela Inicial ProBlockly



Fonte: Elaborada pelo autor

A tela principal é composta por uma área de barra de título, menu lateral (ou caixa de ferramentas), visualizar a conexão com servidor e mudar para a página de simulação. A Figura 4.2 apresenta uma visão geral da tela principal.

Figura 4.2 – Tela principal do *software* ProBlockly



Fonte: Elaborada pelo autor

No Apêndice A apresentam-se outras telas do *software* com detalhamento do menu principal contendo opções para “salvar” um projeto contendo a programação em blocos construída; “abrir arquivo” para abrir um projeto salvo; “configurações” que permite

personalizar as cores do *background* do *software*, principalmente para melhor visualização do simulador; e “experimente” para alternar entre as diferentes visualizações descritas a seguir.

Ainda na Figura 4.2, observa-se a caixa de ferramenta onde estão os comandos que poderão ser utilizados na programação em blocos. Também se identificam duas áreas. Na primeira (Figura 4.3) é a área de trabalho, na qual o aluno pode escolher o tipo de bloco de comando que irá utilizar, esse bloco pode ser posicionado à direita apenas clicando, arrastando e soltando o mesmo.

Esse tipo de interação, *Drag and Drop*, facilita a programação, em que o usuário não precisa se preocupar com erros de sintaxe de código, apenas precisa focar na sequência lógica que deseja construir.

Figura 4.3 – Área de trabalho ProBlockly



Fonte: Elaborada pelo autor

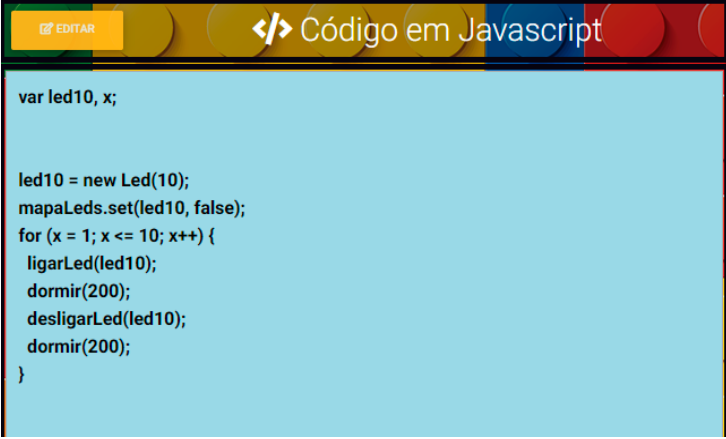
Ainda, sobre a Figura 4.3, tem-se um exemplo de sequência de comandos criados com os blocos para fazer um LED acender e apagar repetidamente com intervalos de tempo pré-definidos. Observa-se ainda que o LED denominado pela variável *led10* utilizará o pino 10 do Arduino.

No Apêndice B apresentam-se os comandos relacionados aos componentes eletrônicos disponibilizados nesta versão do *software*. Exemplos de programação em blocos são apresentados no Apêndice C.

A segunda área (Figura 4.4) pode ser usada para exibição de situações simuladas ou exibição e edição de código, na qual o aluno pode ver o código que é gerado a partir dos

blocos que estão na área de trabalho, a exibição é realizada automaticamente a cada alteração. Entretanto, também existe a opção de se programar diretamente por código, isto faz com que o aluno tenha uma experiência mais próxima da realidade das linguagens de programação comuns. Esse código é gerado em JavaScript, e possui alguns métodos criados especificamente para fazer a comunicação com o Arduino. A Figura 4.4 mostra o resultado da geração de código dos blocos apresentados no exemplo da Figura 4.3.

Figura 4.4 – Área de exibição e edição de código



```

var led10, x;

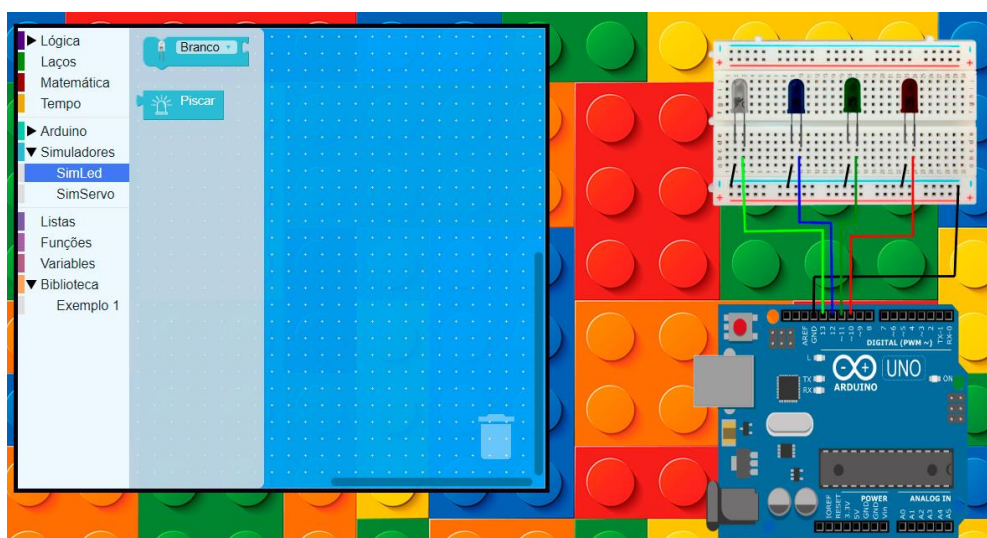
led10 = new Led(10);
mapaLeds.set(led10, false);
for (x = 1; x <= 10; x++) {
  ligarLed(led10);
  dormir(200);
  desligarLed(led10);
  dormir(200);
}

```

Fonte: Elaborada pelo autor

Na versão atual do *software*, existem dois cenários de simulação, denominados SimLed e SimServo. Em futuras atualizações, novos cenários serão incorporados para outros componentes eletrônicos. A ideia é que essa possa auxiliar o aluno a compreender melhor os conceitos de programação antes de utilizar o *hardware*. As Figura 4.5/4.6 (a) e (b) exemplificam uma simulação com LEDs e servo motor, respectivamente.

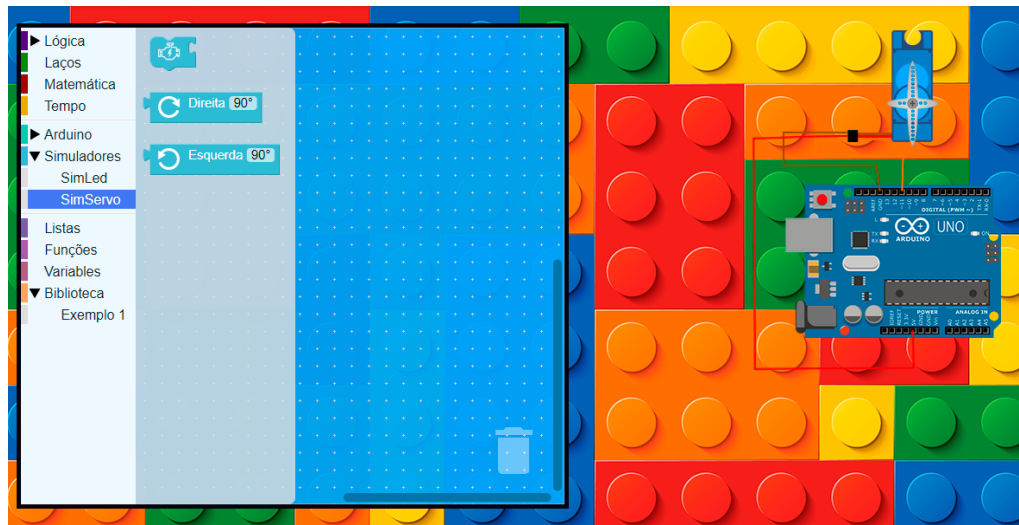
Figura 4.5 – Área de simulação para acionamento de: (a) LEDs (SimLed); e (b) servo motor (SimServo)



(a)

“continua”

Figura 4.5 – Área de simulação para acionamento de: (a) LEDs (SimLed); e (b) servo motor (SimServo)



(b)

Fonte: Elaborada pelo autor

Para fazer a interação com o *hardware*, o usuário clica no botão Rodar, localizado na parte superior direita da Figura 4.2, que em seguida irá extrair o código JavaScript equivalente à sequência criada, e enviará a *String* de código via Ajax, por método POST, para o servidor que tratará a requisição. Junto à requisição, estão partes de código que se utiliza de variáveis e métodos da biblioteca Johnny-Five². Essa biblioteca é responsável por fazer a conexão entre o servidor e o Arduino, fornecendo recursos para a manipulação da placa. Após receber o código, o servidor o executará.

O código, quando executado em JavaScript, fará a comunicação com o Arduino por meio do protocolo Firmata³, em seguida o servidor enviará uma resposta para a página, que avaliará se o procedimento teve êxito ou não, em caso de sucesso, o usuário terá um *feedback* visualizando diretamente o *hardware*, caso contrário, mensagens no *software* o alertarão de um possível erro.

É importante destacar que o servidor de comunicação é simples e fica na própria máquina do cliente, o servidor é iniciado pelo Node.js⁴, ou seja, toda sua implementação também é feita em JavaScript. Também é preciso salientar que em caso de necessidade toda a interação pode ser feita pela Internet, porém por motivos de segurança, o servidor e a página

² Johnny-Five. Disponível em: <<http://johnny-five.io/>>

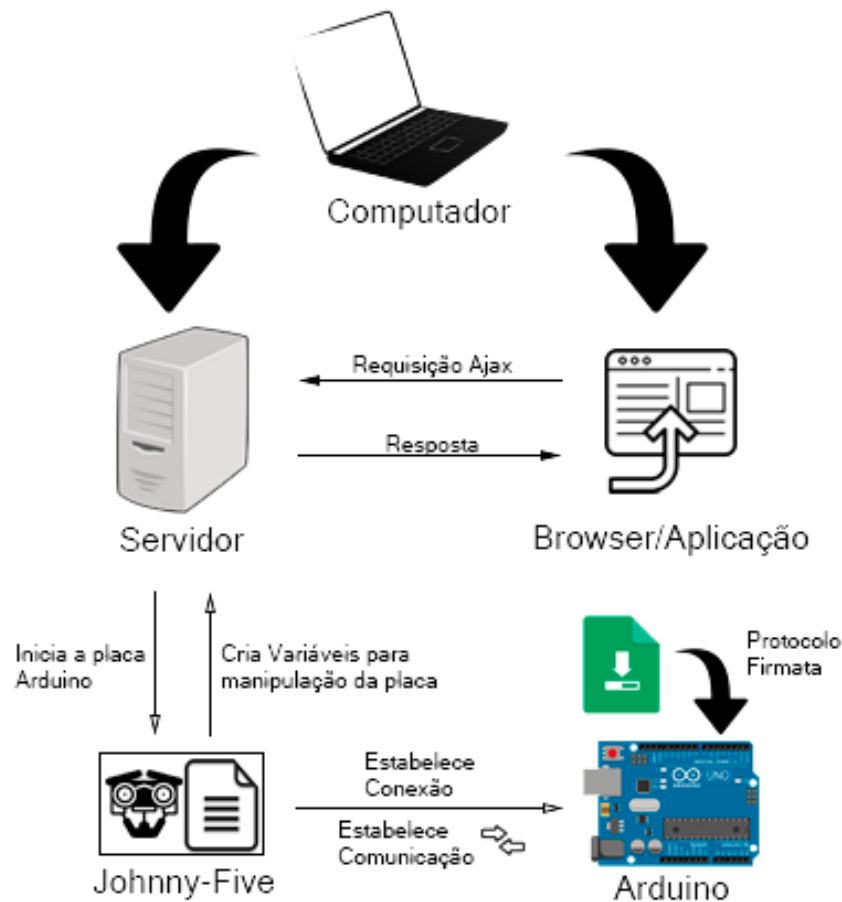
³ Firmata Library. Disponível em: <<https://www.arduino.cc/en/reference/firmata>>

⁴ Node.js. Disponível em: <<https://nodejs.org/en/>>

são locais, já que para a comunicação de alguns módulos, o servidor pode dar acesso ao cliente para executar *scripts* diretamente no computador (servidor), o qual o Arduino está conectado, que dependendo do indivíduo, pode injetar códigos maliciosos.

Como mostra a Figura 4.6, a aplicação faz requisições para o servidor, que tem acesso à porta de comunicação de *hardware*. Para interagir com o Arduino, é necessária a biblioteca Johnny-Five e o protocolo Firmata instalado na placa.

Figura 4.6 – Diagrama de comunicação do software ProBlockly



Fonte: Elaborada pelo autor

A biblioteca Johnny-Five é responsável por criar variáveis para a manipulação da placa, que são utilizadas no servidor. Essas variáveis são referências para objetos que possuem métodos específicos para cada componente. Esses métodos só podem ser devidamente interpretados pelo Arduino se esse possuir o protocolo de comunicação Firmata instalado, esse protocolo pode ser encontrado nos exemplos que a própria IDE oficial do Arduino oferece.

O protocolo Firmata foi desenvolvido para possibilitar a integração entre o microcontrolador (Arduino) e o computador, ou outro dispositivo móvel havendo, desta forma, uma melhor comunicação (VEIGA, 2018). Este protocolo pode ser implementado em

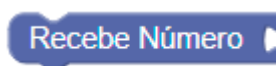
vários tipos de arquitetura de microcontroladores, sendo o Arduino o mais comum e com mais funcionalidades, como o Firmata é implementado em *firmware*, ele facilita a manipulação do microcontrolador e torna mais simples seu uso em ambientes de programação. Além disso, a biblioteca de referência já vem no pacote de software da IDE do próprio Arduino (STEINER, 2009). Desta forma, a utilização do protocolo Firmata, contribui significativamente com a manipulação de microcontroladores, agilizando e simplificando todo o processo da programação.

4.2.1 Blocos Lógicos

O desenvolvimento do ambiente baseado em programação em blocos é tido pela biblioteca Blockly⁵ e apesar de possuir vários blocos pré-definidos, houve a necessidade de desenvolver blocos específicos que representasse comandos relacionados ao *hardware*.

A biblioteca Blockly dá suporte para criação de novos blocos personalizados, e para isto, segue-se uma sintaxe simples, a Figura 4.7 ilustra um exemplo de bloco.

Figura 4.7 – Exemplo de bloco



Fonte: Elaborada pelo autor

Para a criação de um bloco simples como o representado na Figura 4.7, é preciso de uma estrutura, como:

- O nome do bloco, vamos chamá-lo de “exemplo”; e
- Uma função construtora.

O nome do bloco é seu identificador, é com ele que se referencia ao adicionar um bloco na caixa de ferramentas (Menu Lateral). A função é responsável por inicializar parâmetros como: cor do bloco e tipo de conexões, por padrão a função se chama “init”.

Ao se pensar em um bloco simples como este, podemos perceber que ele possui um texto e uma cor, além de 3 conexões. Neste caso os parâmetros utilizados foram:

⁵ Blockly. Disponível em: <<https://developers.google.com/blockly>>

1. Um valor que será recebido em uma variável chamada “numero” na conexão da direita, ela é restrita a somente números;
2. Um texto, que neste caso é “Recebe Número”; e
3. Uma cor, neste caso o Azul.

Este exemplo em código é representado pela Figura 4.8.

Figura 4.8 – Código para criação de um bloco

```

573 Blockly.Blocks['exemplo'] = {
574   init: function() {
575     this.appendValueInput("numero")
576       .setCheck("Number")
577       .appendField("Recebe Número");
578     this.setPreviousStatement(true, null);
579     this.setNextStatement(true, null);
580     this.setColour(230);
581     this.setTooltip("");
582     this.setHelpUrl("");
583   }
584 };

```

Fonte: Elaborada pelo autor

Essa é a primeira parte para criação de um bloco, como esses blocos são utilizados para programação, eles precisam de alguma forma serem relacionados a um código em JavaScript.

Figura 4.9 – Função que retorna o código do bloco

```

570
571 Blockly.JavaScript['exemplo'] = function(block) {
572   var numero = Blockly.JavaScript.valueToCode(block, 'numero', Blockly.JavaScript.ORDER_ATOMIC);
573   // Apenas recebe um número
574   var code = ''+numero+';\n';
575   return code;
576 };

```

Fonte: Elaborada pelo autor

A Figura 4.9 mostra a função de retorno de código, esta função pode ser chamada pela aplicação ao iniciar o processo de execução do código feito pelo aluno e é chamada automaticamente quando o bloco é adicionado à área de trabalho. Neste exemplo, a função apenas retorna o valor que é recebido na conexão à direita, seguido de ponto e vírgula e uma quebra de linha ao final.

4.2.2 Menu Lateral ou Caixa de Ferramentas

A caixa de ferramentas é o local no qual as opções de blocos aparecem para o aluno. Em forma de menu lateral, pode-se escolher blocos, que estão divididos em categorias, de acordo com suas funções.

A implementação do menu é feita através de um código *Extensible Markup Language* (XML), a Figura 4.10 exemplifica esta implementação.

Figura 4.10 – Código de exemplo para criação da categoria Lógica

```

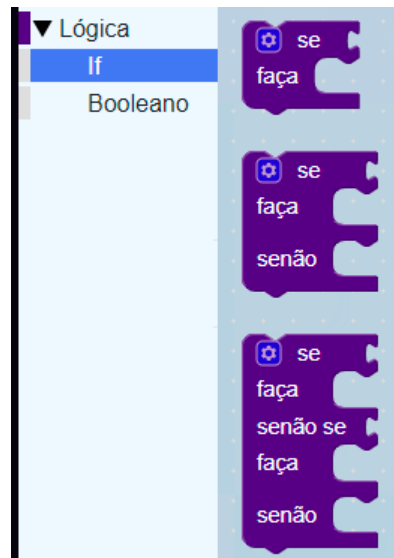
1  <xml id="toolbox" style="display: none">
2    <category name="Lógica" colour="#4B0082">
3      <category name="If">
4        <block type="controls_if"></block>
5        <block type="controls_if">
6          <mutation else="1"></mutation>
7        </block>
8        <block type="controls_if">
9          <mutation elseif="1" else="1"></mutation>
10       </block>
11      </category>
12      <category name="Booleano">
13        <block type="logic_compare"></block>
14        <block type="logic_operation"></block>
15        <block type="logic_negate"></block>
16        <block type="logic_boolean"></block>
17        <block type="logic_null"></block>
18        <block type="logic_ternary"></block>
19      </category>
20    </category>
21  </xml>

```

Fonte: Elaborada pelo autor

O resultado do código XML da Figura 4.10 pode ser visto na Figura 4.11. Este código não é visível, para que o menu lateral seja adicionado, é preciso injetar o XML na área de trabalho, por meio do método *inject* da biblioteca Blockly. Esse procedimento é feito na criação da área de trabalho, junto à caixa de ferramentas são adicionados outros componentes como, por exemplo, o ícone de lixeira e a grade de plano de fundo, que serve para demarcar visualmente os espaços onde os blocos se encaixam.

Figura 4.11 – Exemplo de caixa de ferramentas



Fonte: Elaborada pelo autor

4.2.3 O Hardware

Para ilustrar a ideia da construção da parte física do ambiente, foram utilizados os seguintes componentes eletrônicos:

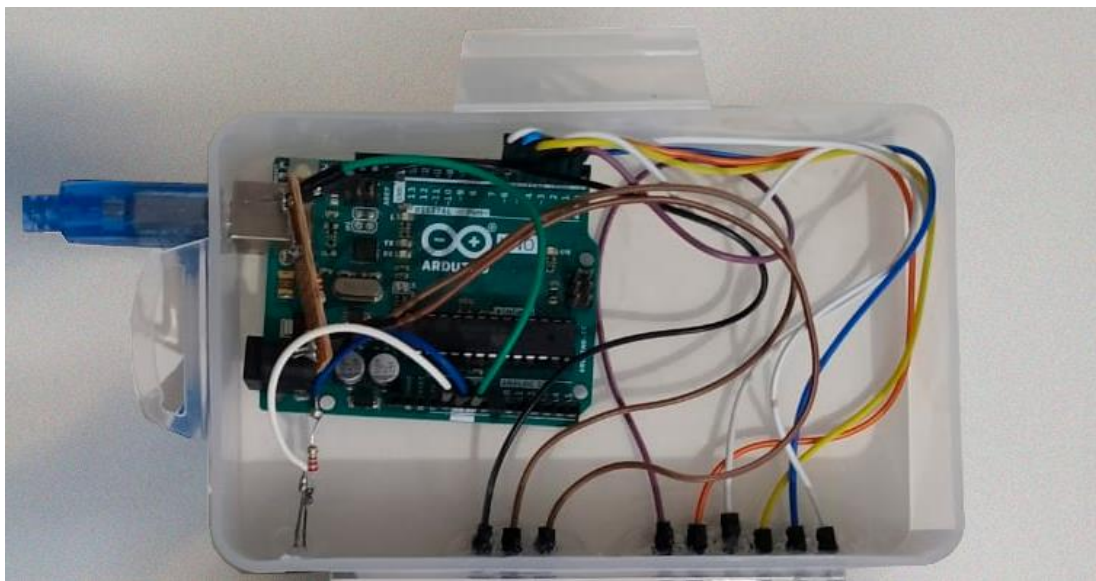
- Microcontrolador Arduino Uno encapsulado;
- Módulo sensor de distância ultrassônico;
- LEDs Infravermelho (Emissor e Receptor);
- LEDs Coloridos; e
- Servo Motor.

O *hardware* consiste na união desses componentes com blocos de montar de plástico interligados de baixo custo, similares ao Lego, fabricando peças separadas, que podem ser montadas e utilizadas conforme a imaginação do aluno, bem como do microcontrolador Arduino (Figura 4.12) encapsulado em uma interface que facilite a interação com este. Ainda que tenham sido integrados os componentes eletrônicos destacados, novos componentes poderão ser utilizados.

Figura 4.12 – Microcontrolador Arduino encapsulado: (a) aparência externa; (b) aparência interna



(a)



(b)

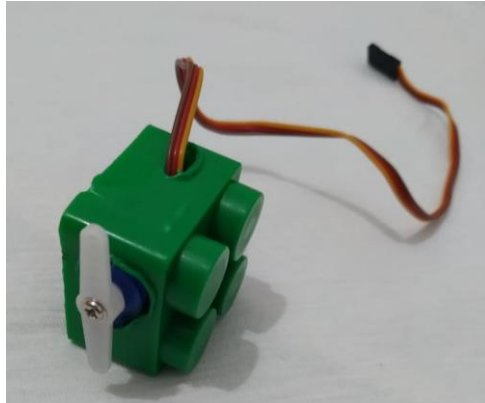
Fonte: Elaborada pelo autor

Os blocos unidos com os componentes eletrônicos são protótipos⁶ que servem para interação com o *software*. O Intuito não é fabricar blocos prontos para comercialização, mas demonstrar o que é possível de se desenvolver.

A Figura 4.13 ilustra um bloco de montar integrado com um servo motor, possui duas saídas, sendo uma para os fios e outra para a hélice.

⁶ Tutorial de fabricação dos protótipos, Disponível em: <<https://www.youtube.com/watch?v=7vig5b-sEC8>>

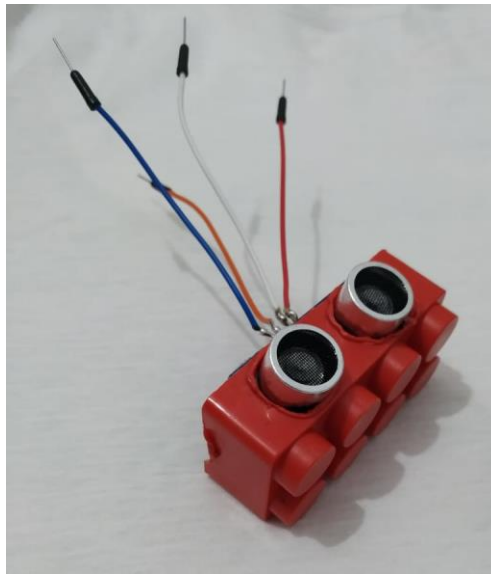
Figura 4.13 – Servo Motor integrado a um bloco de montar de plástico interligável



Fonte: Elaborada pelo autor

Outro exemplo pode ser observado na Figura 4.14. Nessa pode-se identificar o módulo sensor de distância ultrassônico integrado a um bloco de montar.

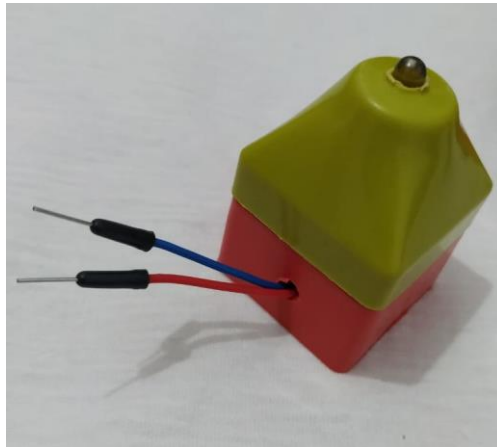
Figura 4.14 – Sensor ultrassônico integrado a um bloco de montar de plástico interligável



Fonte: Elaborada pelo autor

Outro protótipo é ilustrado na Figura 4.15. Neste observa-se um LED na extremidade do bloco e as saídas de ligação na lateral. Este ilustra tanto um LED luminoso quanto um infravermelho (receptor e emissor), pois a aparência desses LED's são iguais.

Figura 4.15 – LED integrado a um bloco de montar de plástico interligável

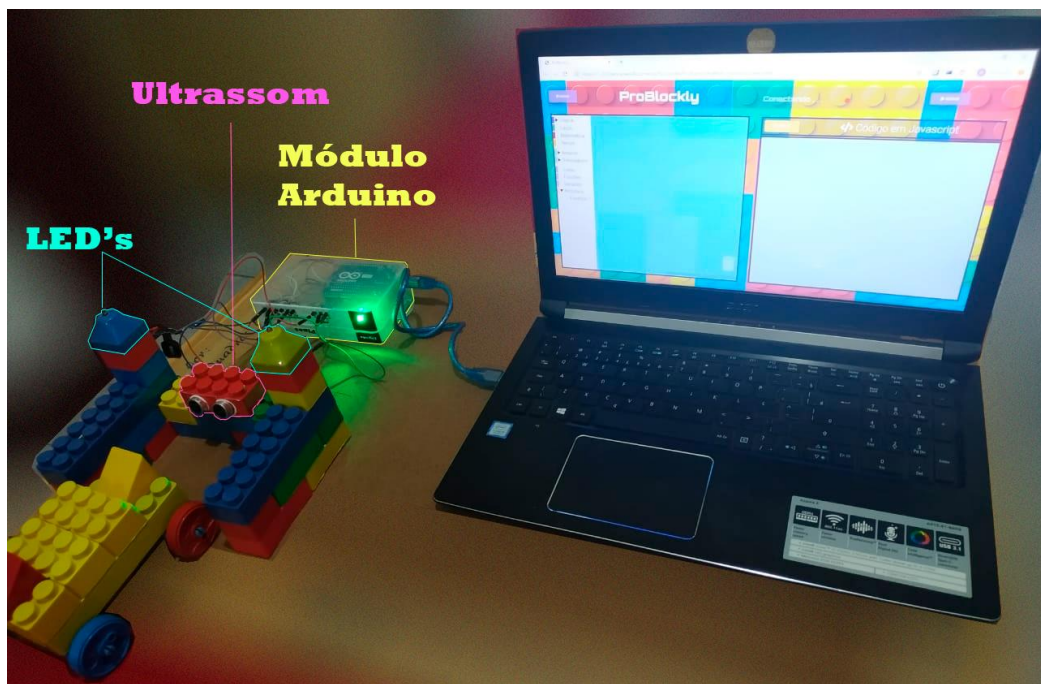


Fonte: Elaborada pelo autor

Esses são apenas alguns exemplos de protótipos que ilustram a integração de componentes eletrônicas a bloco de montar de plástico interligável de baixo custo e fácil replicação.

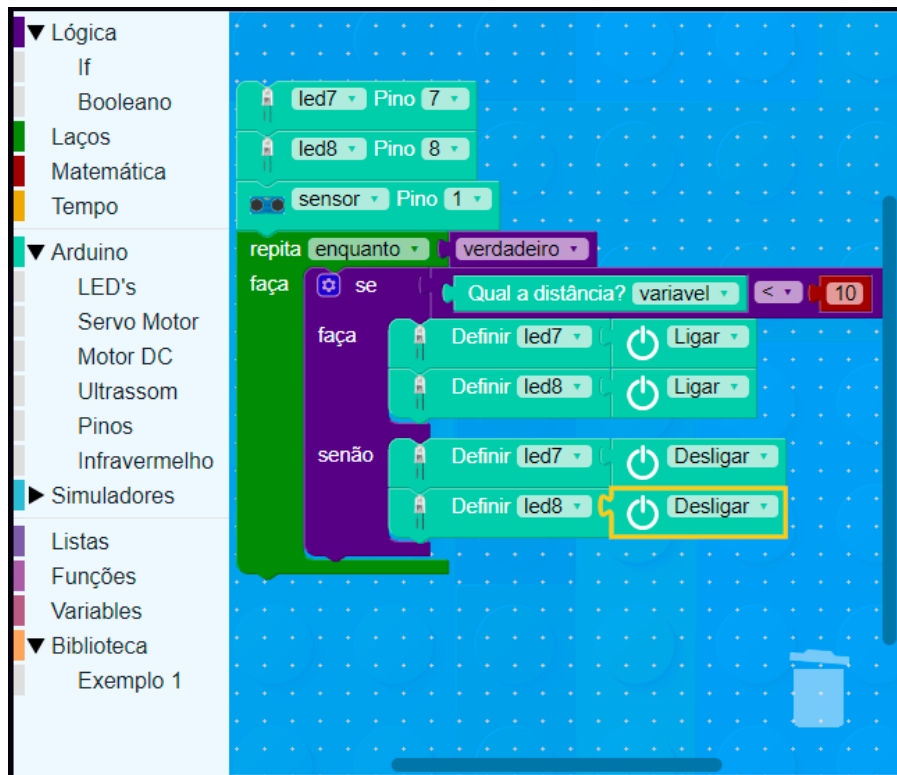
Para ilustrar a utilização desses componentes, ou seja, uma exemplificação do ambiente ProBlockly, apresenta-se um protótipo que simula uma garagem construída com blocos de montar de plástico interligados contendo um sensor de distância (Figura 4.14) e LED's (Figura 4.15). A Figura 4.16 ilustra a situação na qual o aluno precisa programar o *hardware* para acender os LED's quando um carrinho entrar na garagem. A programação em blocos é apresentada na Figura 4.17 e o JavaScript na Figura 4.18.

Figura 4.16 – Simulação de uma garagem construída no ambiente proposto



Fonte: Elaborada pelo autor

Figura 4.17 – Programação em blocos da simulação de uma garagem construída no ambiente proposto



Fonte: Elaborada pelo autor

Figura 4.18 – JavaScript da simulação de uma garagem construída no ambiente proposto

```

EDITAR </> Código em Javascript

var led7, led8, sensor, variavel;

led7 = new Led(7);
mapaLeds.set(led7, false);
led8 = new Led(8);
mapaLeds.set(led8, false);
criarUltrassom(sensor,1);
while (true) {
  if ((verificaDistancia(variavel)) < 10) {
    ligarLed(led7);
    ligarLed(led8);
  } else {
    desligarLed(led7);
    desligarLed(led8);
  }
}

```

Fonte: Elaborada pelo autor

Acredita-se que a proposta do *hardware* é de baixo custo. A Tabela 4.1 apresenta o investimento para construção dos componentes apresentados.

Tabela 4.1 – Investimento financeiro para construção do *hardware* do ambiente ProBlockly

Material	R\$
Kit de blocos de montar de plástico interligados*	
Microcontrolador Arduino encapsulado	Microcontrolador Arduino Uno Invólucro Led Conectores Fios
Sensor de distância ultrassônico integrado a um BMPI*	Módulo de Sensor de distância ultrassônico
LED Infravermelho integrado a um BMPI*	LED Infravermelho emissor LED Infravermelho receptor Conectores Fios
LED Colorido integrado a um BMPI*	LED Colorido Conectores Fios
Servo Motor integrado a um integrado a um BMPI*	Servo Motor Conectores Fios
Total R\$ 110,00	

Legenda: BMPI → Bloco de montar de plástico interligado

* → O bloco de plástico de cada solução já é considerado no valor do *kit*

Fonte: Elaborada pelo autor

Para o cálculo considerou-se uma unidade de cada componente eletrônico, bem como não se considerou o custo de ferramentas para construção dos blocos, como, por exemplo, furadeira, brocas e cola.

4.3 Considerações

O ambiente ProBlockly tem como finalidade auxiliar professores no ensino de lógica de programação para alunos, do Ensino Fundamental II e do Ensino Médio. Por ser voltada para crianças, o *software* possui um visual colorido que chame a atenção dessas, possui também um *layout* minimalista para facilitar a interação, isso faz com que a criança preocupe-se apenas com a programação.

O desenvolvimento do software foi feito através do editor de código Visual Studio Code⁷, o qual possibilita a edição, refatoração, depuração e versionamento de código. A linguagem de programação escolhida foi o JavaScript, que foi utilizada tanto para parte *client-side* quanto para parte *server-side*.

A parte do cliente foi programada com JQuery⁸, uma biblioteca popular do JavaScript, já a parte do servidor, necessitou da utilização do Node.js, uma plataforma de aplicação feita sobre o interpretador JavaScript do Google Chrome. O Node.js possibilitou a utilização da biblioteca Johnny-Five que é uma aplicação JavaScript para robótica e Internet das coisas (IoT, do inglês *Internet of Things*).

Na placa Arduino utilizou-se o protocolo Firmata, protocolo de comunicação com microcontroladores, que foi implementado em *firmware* por meio do Arduino IDE, ele foi utilizado para criar um canal de comunicação entre o software e o Arduino, assim é possível ter o controle total sobre o microcontrolador. O Arduino IDE utiliza o C++ como linguagem de programação.

A Linguagem de Marcação de Hipertexto (HTML, do inglês *Hypertext Markup Language*) e a Folhas de Estilo em Cascata (CSS, do inglês *Cascading Style Sheets*) foram utilizadas para criação da página Web e sua apresentação visual, respectivamente.

Para finalizar, alguns *software* para edição e criação de imagens também foram utilizados, como o Gimp⁹ para edição de imagens e geração de arquivos com formato .svg que foram fundamentais para animação dos simuladores.

4.4 Dificuldades Encontradas

A proposta iniciou-se com o intuito de criar uma biblioteca de blocos própria em linguagem Java, contudo percebeu-se que na busca do desenvolvimento da parte gráfica para a lógica de conexão de blocos e interação necessitava de um tempo excessivo. Foi utilizada a classe JPanel, que possui um método *paint*, durante esta etapa observou-se que ao adicionar componentes como JTextField, o JPanel disparava o método *repaint* infinitas vezes, isto

⁷ Visual Studio Code. Disponível em: <<https://code.visualstudio.com/>>

⁸ JQuery. Disponível em: <<https://jquery.com/>>

tornava inviável a interação com o campo de texto. Este *bug* dificultou a continuação do projeto. A correção da interação da classe `JPane` ao adicionar componentes fugia totalmente do que este trabalho propunha.

Como a proposta principal é auxiliar do ensino de programação e não a criação de uma nova biblioteca gráfica de blocos, então se optou por pesquisar soluções mais efetivas, para que o foco do desenvolvimento estivesse voltado para a interação do *software* com o *hardware*. Assim, mudou-se a escolha da linguagem de programação de Java para JavaScript, pois essa já possui bibliotecas que ajudam a agilizar o processo de desenvolvimento.

É preciso salientar que, para solução de alguns problemas, foram utilizadas bibliotecas que tornaram um pouco restrita a forma de programação de alguns módulos, principalmente em questão de tratamento de erros. A biblioteca Johnny-Five, foi utilizada para fazer a comunicação entre *software* e *hardware*, porém em alguns casos essa não possibilita o tratamento de erros, como a desconexão da placa durante o uso do *software*, este fato dificultou tratar situações atípicas.

Outra dificuldade encontrada está na execução de alguns métodos de forma síncrona, já que o cliente e o servidor utilizam tecnologias assíncronas (JavaScript e Node.js), porém foi solucionado com a utilização do método JavaScript `setInterval` e recursividade.

⁹ Gimp. GNU Image Manipulation Program. Disponível em: <<https://www.gimp.org/>>

5 Conclusão

Neste capítulo apresenta-se a conclusão sobre o trabalho desenvolvido nesta monografia. Na Seção 5.1 apresentam-se as considerações finais. Na Seção 5.2 descrevem-se as contribuições desta monografia e na Seção 5.3 algumas propostas para trabalhos futuros.

5.1 Considerações Finais

Observando o cenário atual do ensino de lógica de programação nas escolas, percebe-se que diversos países já possuem em seu currículo básico o ensino desse conteúdo. O ensino de programação desenvolve várias competências, como o raciocínio lógico, além de ser um conteúdo interdisciplinar. Porém introduzir conceitos de programação para crianças não é uma tarefa fácil, sabendo-se também da dificuldade das escolas em passar o conhecimento para os alunos, independente de qual seja, devido aos métodos tradicionais de apresentação de conteúdo, torna ainda mais complicado o ensino de lógica de programação.

A solução então é utilizar-se de estratégias que facilitem a aquisição da informação. O uso de TICs em sala de aula melhora a qualidade do ensino e, junto a RE, faz com que os alunos aprendam de forma mais interativa, e essa quebra de padrão no ensino, torna aulas tradicionais em aulas inovadoras, o aluno então passa a ter mais interesse em aprender.

A pesquisa de novas ferramentas que auxiliem professores e alunos vem aumentando, porém o uso da RE ainda é pouco explorado. Parte desse problema se deve ao assunto ainda ser relativamente novo e, um pequeno percentual, está relacionado ao fato da dificuldade de acesso a equipamentos.

A RE então pode ser fundamental para auxiliar no ensino de um conteúdo que muitas vezes é complexo, como é o caso da programação, e aprender lógica de programação junto à robótica pode ser a solução ideal para os alunos.

Destarte, ferramentas inovadoras que integrem várias tecnologias, podem trazer formas diferentes de passar e receber conhecimento, esta é uma necessidade das escolas na era da informação.

5.2 Contribuições deste Trabalho

Este trabalho teve como objetivo a criação de um ambiente educacional para auxiliar o ensino de lógica de programação para alunos, com foco nos Ensinos Fundamental II e Médio, porém acredita-se que pode ser utilizado inclusive em disciplinas introdutórias do ensino superior que tem a Computação como área meio ou fim.

Como contribuição principal, este trabalho traz o desenvolvimento de uma solução para auxiliar professores no ensino de programação, somado a criação de um protótipo de *hardware* para interação com o *software*, principalmente pelo fato de tratar-se de uma solução de baixo custo e que pode ser utilizada em cenários domésticos e/ou em ambiente escolar.

5.3 Proposta para Trabalhos Futuros

Para trabalhos futuros propõem-se as seguintes melhorias:

- O desenvolvimento de simuladores mais completos com opções de montagem de cenários dinâmicos, em que o aluno possa encaixar os componentes que quiser e fazer suas ligações com uma placa;
- Aperfeiçoamento da integração de componentes com os blocos de montar de plástico interligados, junto a construção de mais variações de opções de componentes eletrônicos;
- A implementação de mais tipos de blocos lógicos para novos componentes eletrônicos;
- Aperfeiçoamento audiovisual da aplicação, com mais efeitos sonoros que facilitem a interação do usuário; e
- A validação do ambiente em um delineamento experimental que analise questões relacionadas à interação e ergonomia, além da verificação da efetividade de seu uso em salas de aula.

REFERÊNCIAS

- AGUIAR, Y. Q.; MACIEL, B. K.; MATTOS, S. D. G.; SOARES, L. B.; OLIVEIRA, V. M. Introdução à Robótica e Estímulo à Lógica de Programação no Ensino Básico Utilizando o Kit Educativo LEGO® Mindstorms. In: **Anais dos Workshops do CBIE**. 2015. p. 1418-1424.
- ALVES, R. M.; SAMPAIO, F.; ELIA, M. F. Duinoblocks: Desenho e implementação de um ambiente de programação visual para robótica educacional. *Revista Brasileira de Informática na Educação*, v. 22, n. 3, 2013.
- BEZERRA, F. L.; DIAS, K. Programação de computadores no ensino fundamental: experiências com logo e scratch em escola pública. *Revista Traços*, 2012.
- BRASIL. Secretaria de educação fundamental. parâmetros curriculares nacionais: introdução aos parâmetros curriculares nacionais / secretaria de educação fundamental. Brasília: MEC/SEF, 1997, p. 126.
- _____. Ministério da educação. Secretaria de Educação Básica. **Diretrizes Curriculares Nacionais Gerais da Educação**. Brasília: MEC, SEB, DICEI, 2013. p. 562.
- CHELLA, M. T. **Ambiente de robótica para aplicações educacionais com superlogo**. 2002.
- GARLET, D.; BIGOLIN, N.; SILVEIRA, S. **Uma proposta para o ensino de programação de computadores na educação básica**. Departamento de Tecnologia da Informação, Universidade Federal de Santa Maria, RS, 2016.
- HEINEN, E. **Raspiblocos: ambiente de programação didático baseado em Raspberry Pi e Blockly**. Dissertação (B.S. thesis) – Universidade Tecnológica Federal do Paraná, 2015.
- INÁCIO, F. F.; OLIVEIRA, K. L.; MARIANO, M. L. S. Estilos intelectuais e estratégias de aprendizagem: percepção de professores do ensino fundamental. *Psicologia Escolar e Educacional, Associação Brasileira de Psicologia Escolar e Educacional*, v. 21, n. 3, 2017, p. 447–455.
- JÚNIOR, J. C. R. P.; RAPKIEWICZ, C. E.; DELGADO, C.; XEXEO, J. A. M. Ensino de algoritmos e programação: uma experiência no nível médio. **Anais...** In: XIII Workshop de Educação em Computação (WEI'2005). São Leopoldo, RS, Brasil. [S.l.: s.n.], 2005.
- LÉVY, P. **As tecnologias da inteligência: o futuro do pensamento na era da informática**. Rio de Janeiro: Editora 34, 1993. Coleção Trans, 1999.
- MIRANDA, L. C. D.; SAMPAIO, F. F.; BORGES, J. A. S. Robofácil: Especificação e implementação de um kit de robótica para a realidade educacional brasileira. *Brazilian Journal of Computers in Education*, v. 18, n. 03, 2011, p. 46–58.

MIRANDA, P. V.; PEREIRA, A. R.; RISSETTI, G. **A linguagem logo como alternativa lúdica de ensino**. Seminário Nacional de Pesquisa em Educação, 2016.

MORAN, J. M. **Novas tecnologias e mediação pedagógica**. [S.l.]: Papyrus Editora, 2000.

NOVA ESCOLA. **Por que ensinar Programação na escola?** 2016. Disponível em: <<https://novaescola.org.br/conteudo/113/por-que-ensinar-programacao-na-escola>>. Acesso em: 01 set. 2018.

OLIVEIRA, G. P.; MARCELINO, S. B. Estratégias didáticas com o software superlogo: Adquirir fluência e pensar com tecnologias em educação matemática acquire fluency and think with technologies in mathematics education: A proposal using superlogo software. *Educação Matemática Pesquisa: Revista do Programa de Estudos Pós-Graduados em Educação Matemática*, v. 17, n. 4, 2015, p. 816–842.

OLIVEIRA, J. F.; LIBÂNEO, J. C.; TOSCHI, M. S. **Educação escolar: políticas, estrutura e organização**. [S.l.]: Cortez Editora, 2017.

OLIVEIRA, M. L. S.; SOUZA, A. A.; BARBOSA, A. F.; BARREIROS, E. F. S. Ensino de lógica de programação no ensino fundamental utilizando o scratch: um relato de experiência. **Anais...** In: XXXIV Congresso da SBC-XXII Workshop de Ensino de Computação, Brasília. [S.l.: s.n.], 2014.

PAPERT, S. **Mindstorms: Children, computers, and powerful ideas**. [S.l.]: Basic Books, Inc., 1980.

PEREIRA, J.; JOSÉ, C.; RAPKIEWICZ, C. E. O processo de ensino-aprendizagem de fundamentos de programação: uma visão crítica da pesquisa no Brasil. **Anais...** In: Workshop sobre Educação em Computação (WEI). [S.l.: s.n.], 2004.

PUGA, S.; RISSETTI, G. **Lógica de programação e estruturas de dados, com aplicações em Java**. [S.l.]: Pearson Educación, 2009.

RICHTEL, M. **Programação vira disciplina em escolas infantis nos EUA. 2014**. Disponível em: <<https://www1.folha.uol.com.br/tec/2014/05/1456608-programacao-vira-disciplina-em-escolas-infantis-nos-eua.shtml>>. Acesso em: 17 ago. 2018.

SANTOS, F. L.; NASCIMENTO, F. M. S.; BEZERRA, R. M. Reduc: A robótica educacional como abordagem de baixo custo para o ensino de computação em cursos técnicos e tecnológicos. **Anais...** In: Workshop de Informática na Escola. [S.l.: s.n.], v. 1, n. 1, 2010, p. 1304–1313.

SASAHARA, L. R.; CRUZ, S. M. S. Hajime—uma nova abordagem em robótica educacional. **Anais...** In: Workshop de Informática na Escola. [S.l.: s.n.], v. 1, n. 1, 2007.

SILVA, N. C.; LIMA, A. C.; SOUZA, N.; SOUSA, D. F. Raciocínio lógico nas escolas: Uma introdução ao ensino de algoritmos de programação. **Anais...** In: Workshops do Congresso Brasileiro de Informática na Educação. [S.l.: s.n.], v. 6, n. 1, 2017, p. 1011.

STEINER, H. C. Firmata: Towards making microcontrollers act like extensions of the computer. Disponível em: <http://users.notam02.no/arkiv/proceedings/NIME2009/nime2009/pdf/author/nm090182.pdf>. Acesso em: 06 Mar. 2020.

TEZANI, T. C. R. A educação escolar no contexto das tecnologias da informação e da comunicação (TIC): desafios e possibilidades para a prática pedagógica curricular. *Revista Faac*, 2011, p. 36–45.

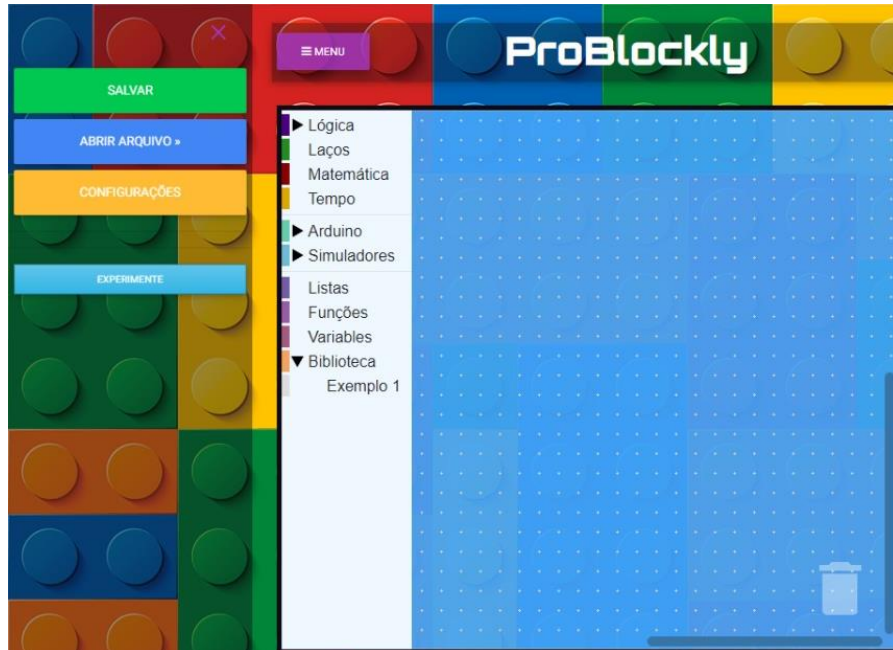
VEIGA, F. Entendendo o protocolo firmata. Disponível em: <https://medium.com/tht-things-hackers-team/entendendo-o-protocolo-firmata-512ae93f0b53>. Acesso em: 06 Mar. 2020.

WANGENHEIM, C. G. V.; NUNES, V. R.; SANTOS, G. D. Ensino de computação com scratch no ensino fundamental – um estudo de caso. *Revista Brasileira de Informática na Educação*, v. 22, n. 3, 2014, p. 115–125.

ZILLI, S. R. **A robótica educacional no ensino fundamental: perspectivas e prática.** Florianópolis, SC, 2004.

APÊNDICE A – Telas do ProBlockly

Figura A.1 – Telas do *software* do ProBlockly: (a) menu principal expandido; (b) opção experimente expandida; (c) opção salvar; (d) opção abrir arquivo; e (e) opção configuração.



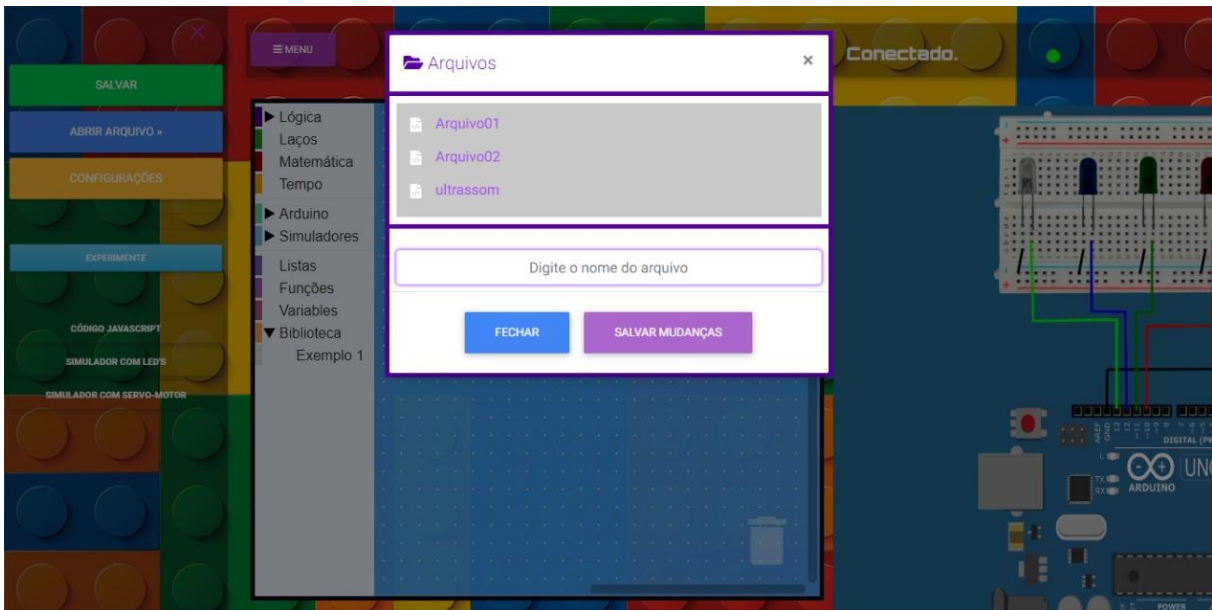
(a)



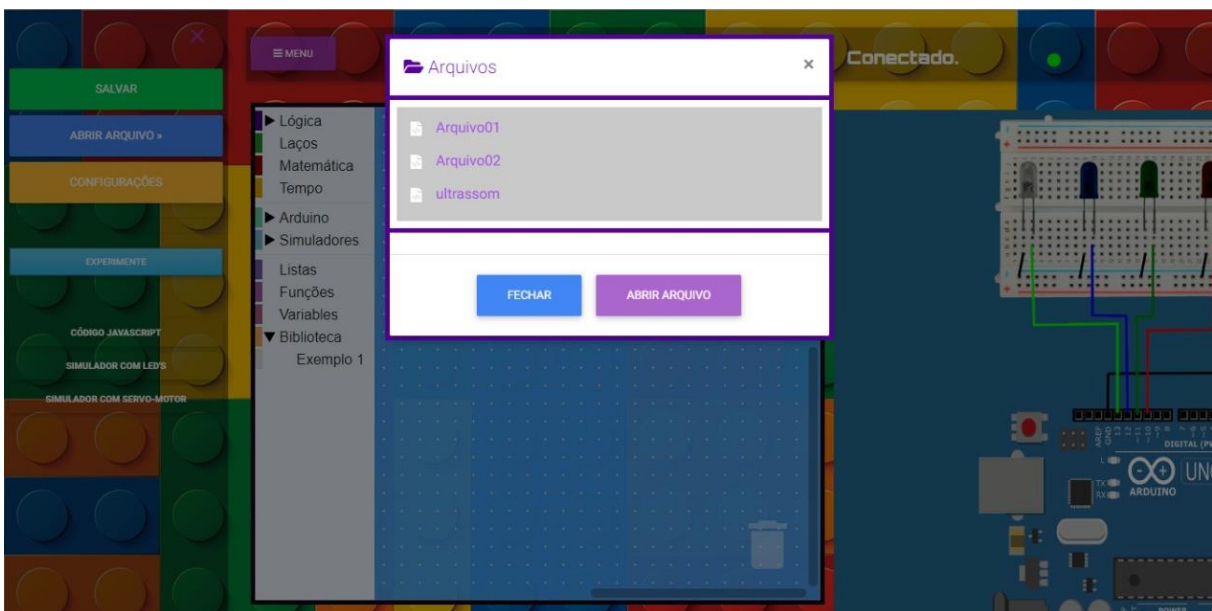
(b)

“continua”

Figura A.1 – Telas do software do ProBlocky: (a) menu principal expandido; (b) opção experimente expandida; (c) opção salvar; (d) opção abrir arquivo; e (e) opção configuração.



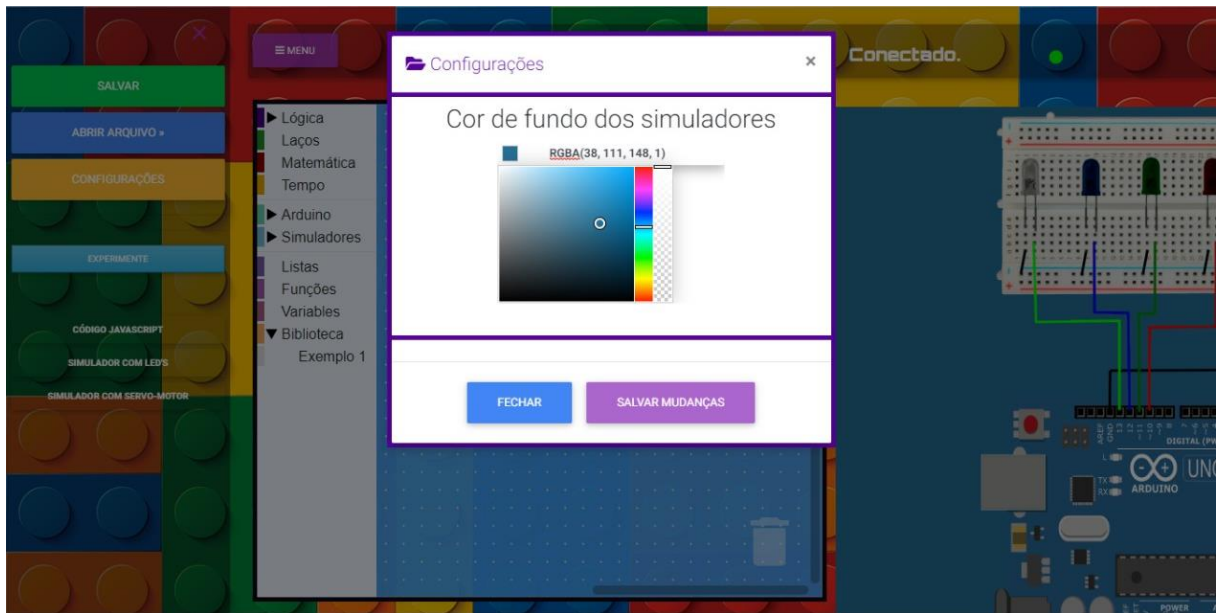
(c)



(d)

“continua”

Figura A.1 – Telas do *software* do ProBlocky: (a) menu principal expandido; (b) opção experimete expandida; (c) opção salvar; (d) opção abrir arquivo; e (e) opção configuração.



(e)

Fonte: Elaborada pelo autor

APÊNDICE B – Comandos relacionados aos componentes eletrônicos disponibilizados na versão 1.0 do ProBlockly

Figura B.1 – Opções do menu de componentes eletrônicos: (a) LED; (b) Servo Motor; (c) Motor DC; (d) Ultrassom; (e) Pinos; e (f) Infravermelho



(a)



(b)

“continua”

Figura B.1 – Opções do menu de componentes eletrônicos: (a) LED; (b) Servo Motor; (c) Motor DC; (d) Ultrassom; (e) Pinos; e (f) Infravermelho



(c)



(d)

“continua”

Figura B.1 – Opções do menu de componentes eletrônicos: (a) LED; (b) Servo Motor; (c) Motor DC; (d) Ultrassom; (e) Pinos; e (f) Infravermelho



(e)



(f)

Fonte: Elaborada pelo autor

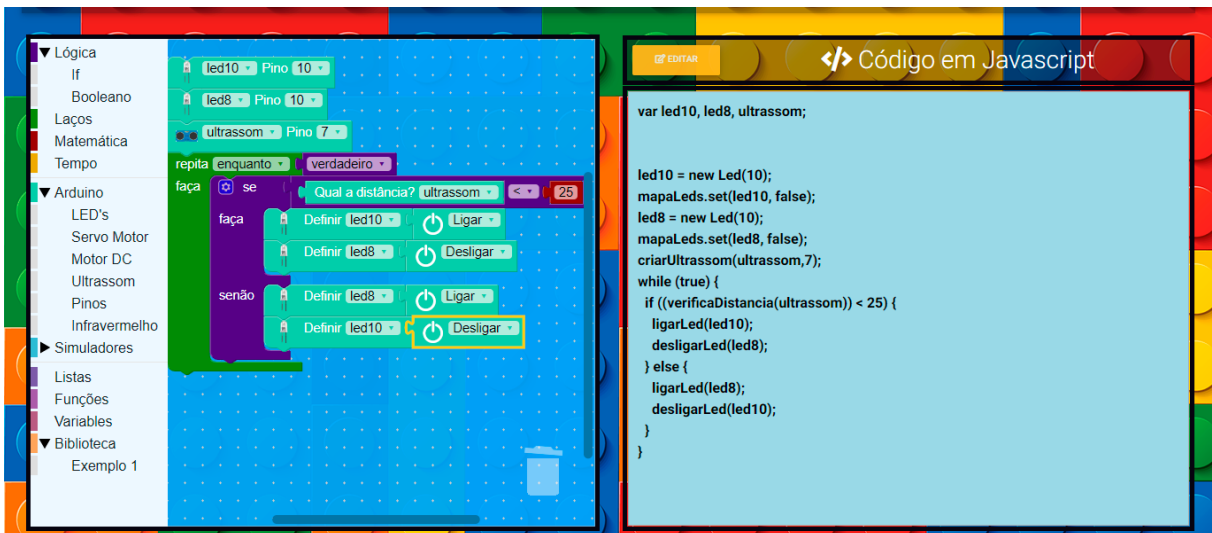
APÊNDICE C – Exemplos de Programação em Blocos no ProBlockly

Figura C.1 – Programação em blocos e JavaScript para acionamento de um LED conectado ao Arduino



Fonte: Elaborada pelo autor

Figura C.2 – Programação em blocos e JavaScript para acionamento de LEDs a partir da leitura de um ultrassom conectado ao Arduino



Fonte: Elaborada pelo autor

Figura C.3 – Programação em blocos e JavaScript para acionamento de um servo motor conectado ao Arduino

The image shows the Arduino IDE interface. On the left, a library manager sidebar is open, showing categories like 'Lógica', 'Laços', 'Matemática', 'Tempo', and 'Arduino'. Under 'Arduino', 'Servo Motor' is selected. The main workspace contains a block-based program: a 'variavel' block set to 'Pino 1', followed by a 'repita' block set to '4 vezes'. Inside the loop, there is a 'faça' block containing 'variavel' set to 'Direita 90°', an 'Esperar' block set to '200', 'variavel' set to 'Esquerda 90°', and another 'Esperar' block set to '200'. On the right, the 'Código em Javascript' tab is active, showing the following code:

```

var variavel;

variavel = new Servo(1);
for (var count = 0; count < 4; count++) {
  variavel.to((-90));
  dormir(200);
  variavel.to((90));
  dormir(200);
}

```

Fonte: Elaborada pelo autor

Figura C.4 – Programação em blocos e JavaScript para acionamento dos LEDs do Simulador SimLed

The image shows the Arduino IDE interface, identical to Figure C.3. The library manager sidebar is open, and 'LED's' is selected under the 'Arduino' category. The main workspace contains the same block-based program as in Figure C.3. On the right, the 'Código em Javascript' tab is active, showing the same JavaScript code as in Figure C.3:

```

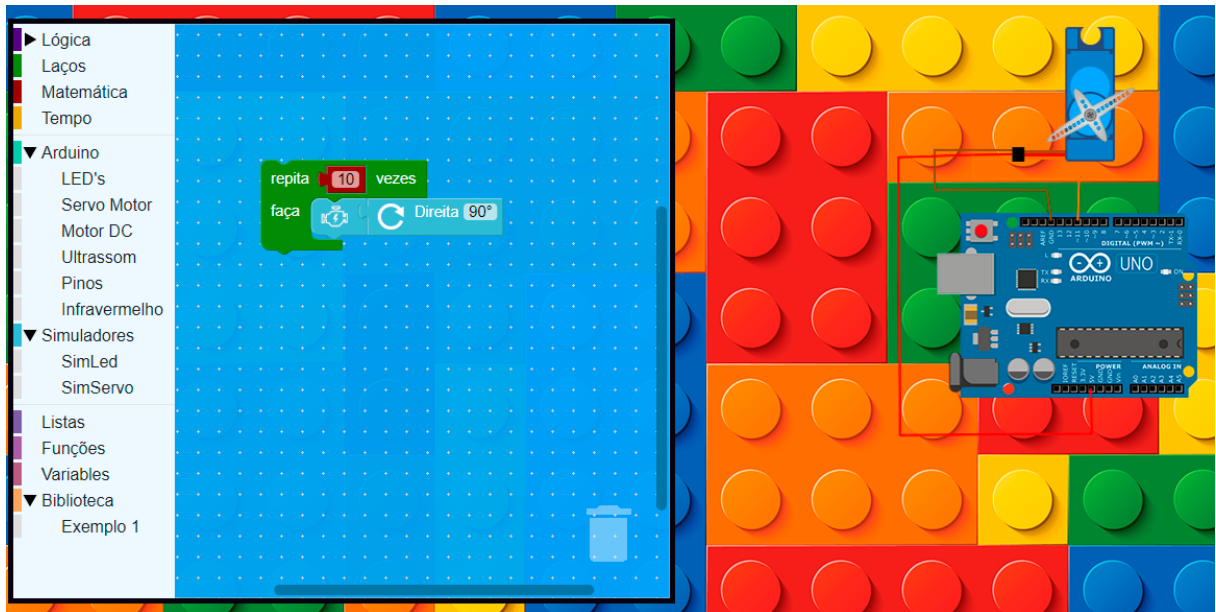
var variavel;

variavel = new Servo(1);
for (var count = 0; count < 4; count++) {
  variavel.to((-90));
  dormir(200);
  variavel.to((90));
  dormir(200);
}

```

Fonte: Elaborada pelo autor

Figura C.5 – Programação em blocos para acionamento do servo motor do Simulador SimServo



Fonte: Elaborada pelo autor