

## Aprendizagem de Algoritmos e Programação por meio da ferramenta visual HelpBlock

Eduardo Rodrigues Gomes<sup>1</sup> Fabrício Pretto<sup>1</sup> Maria Claudete Schorr<sup>2</sup>

<sup>1</sup>Universidade do Vale do Taquari – Univates – Lajeado – RS – Brasil

<sup>2</sup>Universidade Federal do Rio Grande do Sul – UFRGS – Porto Alegre – Brasil

eduardo.r.g.d@gmail.br, fabricio.pretto@univates.br,  
mclaudetesw@univates.br

**Abstract.** *The rates of disapproval and evasion on computing courses are mainly related to learning difficulties in algorithm and programming. The visual tools have been an alternative that allows students create a structure of solution in a macro way, dismissing technical details irrelevant in it first approach. In order to support the learning process of algorithms was developed a visual programming tool called HelpBlock. The tool was validated with 23 students of an Algorithm and Programming class in a Higher Education Institution of Rio Grande do Sul. The outcomes showed a significant improvement during the construction of algorithmic solutions and greater students motivation.*

**Resumo.** *Os índices de reprovação e evasão nos cursos de computação estão relacionados principalmente às dificuldades de aprendizagem de algoritmos e programação. As ferramentas visuais têm sido uma alternativa, pois permitem ao estudante estruturar a solução de maneira macro, dispensando detalhes técnicos irrelevantes em sua primeira abordagem. Com o intuito de apoiar o processo de aprendizagem de algoritmos, desenvolveu-se a ferramenta HelpBlock para programação visual. A ferramenta foi testada com 23 estudantes de uma turma de Algoritmos e Programação de uma IES do Rio Grande do Sul. Os resultados demonstraram melhora significativa na construção das soluções algorítmicas e maior motivação dos estudantes.*

### 1. Introdução

O ensino de programação nos cursos da computação ou mesmo em outros cursos que não são da área, apresentam um índice de evasão e reprovação preocupante (Giraffa; Mora, 2015). Dados do Ministério da Educação (MEC, 2016) apontam que todos os cursos devem ter em sua formação o desenvolvimento do raciocínio lógico e a resolução de problemas. Neste sentido as disciplinas de Algoritmos e Programação atendem esta necessidade, porém percebe-se que os estudantes possuem grande dificuldade na resolução dos problemas propostos nesta disciplina.

Dados estatísticos apontam que a média de reprovação é de 30%, demonstrando que praticamente um terço dos estudantes não conseguem atingir a média mínima (WATSON; LI, 2014). Segundo Ramos et. al (2015), o Brasil aparece em terceiro lugar

com o maior índice de reprovação. Para Amaral (2015) e Iepsen (2013), usar ambientes visuais pode auxiliar na aprendizagem de Algoritmos e Programação, bem como propiciar um aumento significativo na motivação do estudante, tendo em vista a facilidade de interação.

Sendo assim, desenvolveu-se uma ferramenta visual que possa auxiliar na resolução dos problemas, no desenvolvimento do raciocínio lógico, bem como nas habilidades de abstração, sequenciamento, reconhecimento de padrões, entre outras que segundo Henrique e Tedesco (2017) são necessárias para a aprendizagem de Algoritmos e Programação, conseqüentemente facilitando a interação do estudante e promovendo uma maior motivação para permanecer na disciplina. A ferramenta visual, denominada HelpBlock, possui plataforma web e utiliza a biblioteca Blockly (Blockly API, texto digital), facilitando a interação do estudante e assim, promovendo uma maior motivação na resolução de problemas computacionais. Esta ferramenta foi testada com 23 estudantes de Algoritmos e Programação de uma IES do interior do Rio Grande do Sul.

## 2. Embasamento Teórico

Raciocínio lógico, lógica, lógica de programação e linguagem de programação são alguns conceitos básicos necessários para o ensino de Algoritmos e Programação. Lógica é a arte de pensar corretamente e que ensina a colocar em ordem o pensamento (FORBELLONE, 2003). Lógica de programação é a técnica de ordenar pensamentos para alcançar determinado objetivo. Pode ser representada por inúmeras linguagens de programação, como por exemplo, C++, Java, Javascript, entre outras. Porém essas linguagens envolvem muitos detalhes computacionais, então para melhor representar o raciocínio da lógica de programação são utilizados algoritmos.

Noschang et. al (2014) cita que o ensino da programação inicial dos cursos deve focar no desenvolvimento da lógica de programação dos alunos e a capacidade de resolver problemas, deixando a aprendizagem de uma linguagem de programação para um segundo momento. Também argumenta que IDEs (Integrated Development Environment) profissionais disponíveis (Netbeans, Eclipse, entre outras) não são a melhor opção, pois na sua maioria apresentam suas opções e mensagens no idioma inglês, o que acaba sendo um obstáculo a mais na jornada de aprendizado dos estudantes.

Uma das maneiras encontradas para melhorar o desempenho dos alunos nas disciplinas de programação foi o desenvolvimento de ferramentas para auxiliar na programação. Essas ferramentas têm como objetivo apresentar ambientes visuais tornando o desenvolvimento das soluções mais atrativas e motivadoras (AMARAL, 2015); BOUCINHA, 2017).

O Blockly utilizado para o desenvolvimento da ferramenta HelpBlock é uma biblioteca de programação visual que funciona na web e tem código aberto. Este utiliza blocos gráficos conectáveis para representar os conceitos de código como repetição, condição, loops entre outros. Permitindo assim que os usuários usem de princípios de programação sem ter de se preocupar com erros de sintaxe (Blockly API, 2017).

## 2.1 Trabalhos relacionados

Alguns projetos já criados com o objetivo de auxiliar e melhorar o aprendizado de programação para estudantes que estejam iniciando na área de computação e que possuem alguma característica em comum com a ferramenta proposta, serão apresentados nesta seção.

### 2.1.1 RoboMind

RoboMind é um software que foi criado com o objetivo de auxiliar a educação em tecnologia. Possui sua própria linguagem de script e consiste em programar um robô simulado. Assim ao estarem programando os alunos aprendem lógica, informática e robótica. A linguagem desenvolvida recebeu o nome de ‘ROBO’. É uma linguagem resumida que possui um pequeno número de regras e não depende de um conhecimento prévio, possibilitando o início imediato dos alunos na ferramenta. Apesar de ser uma linguagem resumida, ‘ROBO’ permite que sejam desenvolvidos vários programas muito interessantes (Robo Mind, 2017).

### 2.1.2 Embrio

Embrio é um ambiente de programação Visual que auxilia na programação de plataformas Arduino. Ele trabalha com blocos conectáveis, porém também permite que o código seja escrito diretamente caso o usuário julgue a codificação seja uma melhor opção que os blocos em algum momento. Também vale ressaltar que ele permite a criação de blocos próprios (Souza, 2015).

### 2.1.3 Raspiblocos

Trata-se de um ambiente de programação visual capaz de traduzir instruções representadas por blocos encaixáveis, que serve para enviar comandos para um modelo robótico. A interface do ambiente foi implementada utilizando a biblioteca Blockly e o modelo robótico com um controlador Raspberry Pi. Os componentes do modelo, conectados ao minicomputador são: dois motores com rodas, um sensor de proximidade, um led e um botão (Heinen et. al, 2015). A arquitetura do Raspiblocos foi dividida em quatro camadas que são: interface, aplicação, controle e componentes (Figura 1).

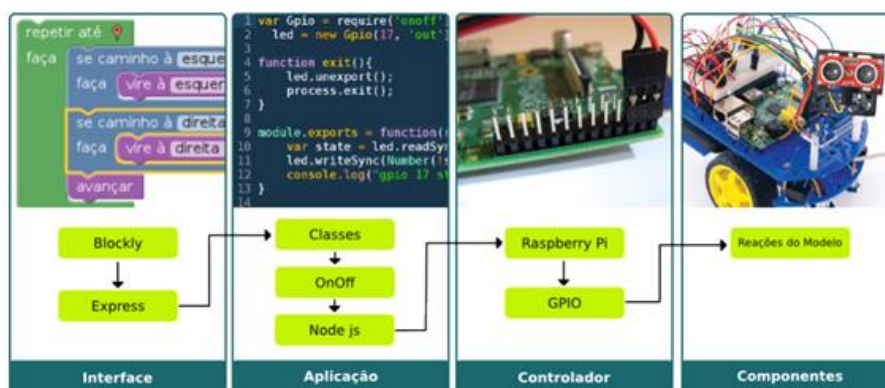


Figura 1: Arquitetura e interação entre os componentes da ferramenta (Heinen, 2015)

### 3. Metodologia

O projeto apresentado é de natureza quantitativa e qualitativa. Segundo D'Angelo (2016), a pesquisa quantitativa consiste em quantificar os dados para obter respostas para um questionamento de um problema de pesquisa. Essa quantificação pode ser realizada tanto através de dados levantados por questionários, como também através da análise de resultados. Este tipo de pesquisa geralmente é usado quando se quer confirmar algum tipo de hipótese. D'Angelo (2016) também fala sobre a pesquisa qualitativa, que diferente da quantitativa, se preocupa mais com as considerações particulares dos entrevistados e os dá uma maior liberdade, normalmente é utilizada para validar teste de produtos.

Também foi levada em consideração a aprendizagem significativa, que segundo Ausubel (1963, p. 58) é o aprendizado de uma nova informação de forma substantiva e não arbitrária. As características básicas da aprendizagem significativa são a não-arbitrariedade e a substantividade. Não-arbitrariedade significa que um material significativo se relacionado de maneira não arbitrária com um conhecimento já existente no indivíduo. O conhecimento prévio é importante para que haja a compreensão e fixação de novas informações que de certa forma “se ancoram” em conhecimentos relevantes já existentes na estrutura cognitiva da pessoa Ausubel (1963). Ainda segundo Ausubel (1963), a substantividade significa que o que o indivíduo aprende é a substância do novo conhecimento, das novas ideias, e não as palavras precisas usadas para expressá-las.

Então levando em conta a ideia de aprendizagem significativa, a realização do levantamento de resultados foi feito através da aplicação de um pré-teste contendo 2 exercícios de programação, seguido de uma intervenção pedagógica onde foi apresentada a ferramenta HelpBlock, desenvolvida nesta pesquisa. Após isso então foi aplicado um pós-teste, que assim como o pré-teste consistiu em 2 exercícios de programação. O pré-teste teve como objetivo medir o conhecimento prévio que os alunos tinham sobre o assunto, e o pós-teste serviu para medir se a ferramenta HelpBlock facilitou a resolução dos exercícios e a compreensão da forma como o código funciona. Por fim foi solicitado que os alunos respondessem um questionário de satisfação.

A turma na qual foi realizada a intervenção foi a de Algoritmos e Programação, e contava com 23 alunos. No dia da intervenção os alunos estavam na terceira aula sobre matrizes e assim já tinham conhecimento sobre laços de repetição, testes condicionais, entradas e saídas. A atividade teve duração de duas horas e meia.

### 4. Análise e Discussão dos Resultados

Com intuito de apoiar o processo de identificação do problema e estruturação da solução de maneira macro, as ferramentas visuais oferecem recursos que minimizam a carga cognitiva relativa a escrita de código, mantendo o foco no raciocínio lógico para resolver o problema. A programação por meio de blocos tem sido utilizada em diversos segmentos e por diversos aplicativos. Dessa forma, rapidamente o estudante consegue elaborar uma proposta de solução e validar sua proposição.

A ferramenta HelpBlock, fazendo uso da biblioteca Blockly, seguiu o princípio da programação via blocos, associado a um ambiente de gerenciamento de exercícios. Nessa proposta, o professor tem a possibilidade de cadastrar exercícios e liberar para sua turma. Os estudantes podem acessar a tarefa, resolver e enviar para avaliação. O professor, tem a possibilidade de avaliar e registrar um retorno sobre o exercício.

Os exercícios são cadastrados de acordo com seu tipo de classificação (Sequencial, Condição, Repetição) e nível de dificuldade (Fácil, Médio, Alto). Para os estudantes, no ambiente de prototipação, estão disponíveis as seguintes opções de blocos programáveis:

- Logic – Contém os blocos lógicos de condição como, por exemplo, o if else, número maior ou menor, verdadeiro ou falso.
- Loops – Contém os blocos de repetição como, por exemplo, o while e o for.
- Math – Contém os blocos de matemática, que são blocos que trazem o cálculo da raiz quadrada, por exemplo, ou até blocos que verificam que determinado número é par ou ímpar, restos de divisão entre outros.
- Text – Contém os blocos de texto como, por exemplo, o imprime, converte para maiúscula, bloco criador de texto, e também um bloco de entrada que solicita ao usuário alguma informação.
- Variables – Neste grupo é possível criar as variáveis, assim como utilizar blocos específicos para definir valores a cada variável criada.

Na mesma tela de resolução de problemas, o estudante possui opções adicionais como a visualização do código-fonte, entrega do exercício e execução, conforme ilustra a figura 2. As opções estão detalhadas na sequência.

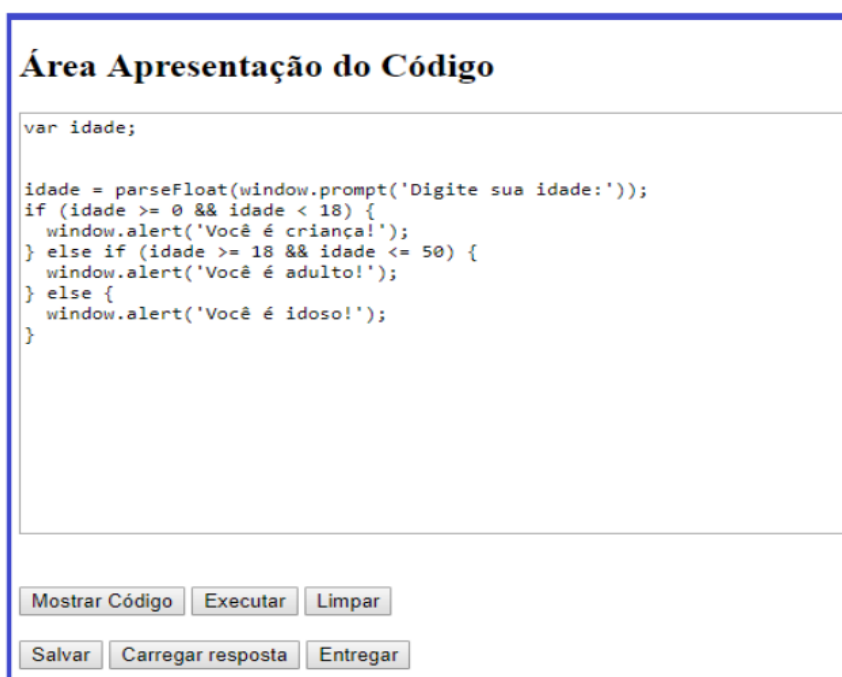
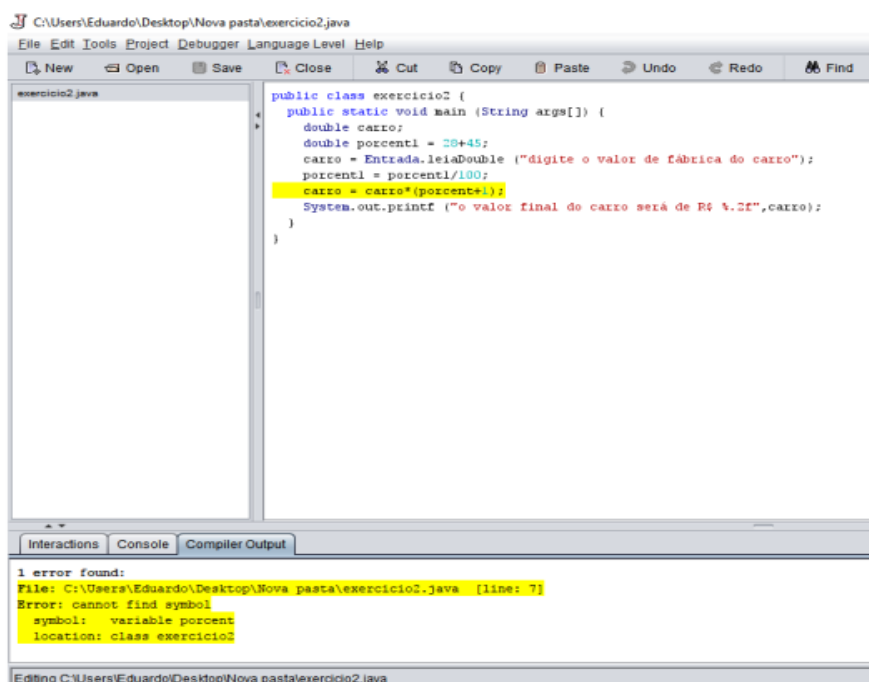


Figura 2: HelpBlock - codificação e entrega do exercício

- Mostrar código – Converte em javascript os blocos que estão na área de desenvolvimento e apresenta na área de apresentação de código.
- Limpar – Limpa a área de apresentação do código.
- Executar – Pega os blocos que estão na área de desenvolvimento e executa o algoritmo formado.
- Salvar – Pega todos os blocos que estão na área de desenvolvimento, converte em texto e salva no banco, porém é importante salientar que cada aluno pode salvar apenas uma resposta por exercício, caso exista uma resposta já salva de determinado usuário para determinado exercício a próxima vez que o usuário clicar em salvar essas respostas irão sobrepor-se mantendo apenas o último registro salvo.
- Carregar resposta – Busca o que está salvo no banco e monta na área de desenvolvimento.
- Entregar – Envia para o professor os blocos que estão na área de desenvolvimento, juntamente com um comentário falando sobre como foi a resolução do exercício e se teve alguma dificuldade durante. Cada aluno pode entregar apenas uma resposta por exercício.

Para analisar o desempenho dos estudantes com a utilização da ferramenta, realizou-se três momentos, sendo eles o pré-teste, a intervenção pedagógica e o pós-teste. Durante o pré-teste foi solicitado que os alunos resolvessem 2 exercícios utilizando a ferramenta com a qual estavam trabalhando durante as aulas, o Dr. Java. Alguns problemas durante a resolução dos exercícios foram percebidos, um deles é a dificuldade em lidar com a sintaxe da ferramenta (figura 3), onde o estudante perde muito tempo para acertar a sintaxe ao invés da solução do problema em si.



```
C:\Users\Eduardo\Desktop\Nova pasta\exercicio2.java
File Edit Tools Project Debugger Language Level Help
New Open Save Close Cut Copy Paste Undo Redo Find
exercicio2.java
public class exercicio2 {
    public static void main (String args[]) {
        double carro;
        double percent1 = 28+45;
        carro = Entrada.leiaDouble ("digite o valor de fábrica do carro");
        percent1 = percent1/100;
        carro = carro*(percent+1);
        System.out.printf ("o valor final do carro será de R$ %.2f",carro);
    }
}
Interactions Console Compiler Output
1 error found:
File: C:\Users\Eduardo\Desktop\Nova pasta\exercicio2.java [line: 7]
Error: cannot find symbol
symbol:   variable percent
location: class exercicio2
Editing C:\Users\Eduardo\Desktop\Nova pasta\exercicio2.java
```

Figura 3: Erro clássico durante a solução do problema no Dr. Java

Após a realização do pré-teste, passou-se a utilizar a ferramenta HelpBlock. Apresentou-se inicialmente a ferramenta para os estudantes, explicando seu funcionamento e os blocos disponíveis. Em seguida os estudantes passaram a resolver exercícios com o HelpBlock, conforme apresentado na figura 4.



Figura 4: Utilização do HelpBlock

Percebeu-se os estudantes motivados ao utilizar a ferramenta e resultados significativos de soluções foram apresentadas por eles. Depoimentos de alunos como: 'Os recursos oferecidos (entradas, processamento e saídas) foram satisfatórios e de boa usabilidade para a resolução do problema'; 'Trata apenas lógica, deixando as formalidades por conta do próprio programa'; 'Ferramenta interessante, principalmente, ao iniciar uma cadeira de programação, pois os blocos lhe mostram como os comandos se interligam, a sequência de execução'. Estes depoimentos demonstram a satisfação do estudante ao usar a ferramenta.

Um pós-teste foi realizado após a intervenção com a utilização do HelpBlock. Como resultados, os alunos apresentaram melhores soluções, além de acertar uma quantidade maior de exercícios. Com isso, pode-se afirmar que a utilização da ferramenta visual HelpBlock auxiliou na aprendizagem de Algoritmos e Programação dos estudantes.

## 5. Conclusões

Utilizar ferramentas que possibilitam oferecer novas maneiras de ensino aos estudantes é fundamental, a programação visual ajuda a despertar o interesse e a motivação, auxiliando na resolução dos problemas e compreensão dos conteúdos propostos.

Mesmo utilizando blocos para programar, a ferramenta ainda possibilita que os alunos gerem o código formado por seus blocos, existindo um contato com as linguagens de programação, o que é importante para um segundo momento, onde se introduziria a programação com as linguagens de alto nível, como Java, Javascript, PHP entre outras.

Os resultados preliminares demonstram a importância de ferramentas que apoiem o processo de ensino-aprendizagem de maneira prática, porém, descomplicada. Isso não significa remover a complexidade mas sim trabalhá-la no momento adequado, fortalecendo o estudante para o aprendizado de novos conhecimentos. As ferramentas visuais contribuem para o processo de entendimento do problema, agindo como um facilitador na resolução de problemas. De posse da visão do todo, da ideia central, o estudante se apropria da capacidade de acessar níveis mais elevados de complexidade, percorrendo um caminho gradativo.

Sendo assim, esta ferramenta demonstrou ser um apoio no ensino de Algoritmos e Programação. Novos testes com mais estudantes pretendem ser realizados.

## Referências

- AMARAL, Érico Marcelo Hoff do. Processo de Ensino e Aprendizagem de Algoritmos integrando ambientes imersivos e o paradigma de blocos de programação visual. Tese UFRGS – PPGIE – Porto Alegre, 2015.
- AUSUBEL, D.P. (1963). The psychology of meaningful verbal learning. New York, Grune and Stratton.
- Blockly API. Disponível em: <<https://developers.google.com/blockly/guides/overview>> Acesso em: 25 mai 2017.
- BOUCINHA, Rafael Marimon. Aprendizagem do Pensamento Computacional e Desenvolvimento do Raciocínio Lógico. Porto Alegre, 2017.
- D'ANGELO, Pedro. Pesquisa quantitativa e pesquisa qualitativa: qual a diferença?. 27 jan. 2016. Disponível em: <<http://blog.opinionbox.com/pesquisa-quantitativa-e-pesquisa-qualitativa-qual-a-diferenca/>> Acesso em: 26 mai 2017.
- FORBELLONE, Andre Luiz Villar. Lógica de programação: a construção de algoritmos e estrutura de dados. Makron Books. São Paulo, 2003
- GIRAFFA, Lucia Maria Martins; MORA, Michael da Costa. Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno. III Conferencia Latinoamericana sobre el Abandono en la Educación Superior (III CLABES), 2015.
- HEINEN, Eduarth. RASPIBLOCOS: Ambiente de Programação Didático Baseado em Raspberry Pi e Blockly, 2015. Disponível em: <[http://tcc.tsi.gp.utfpr.edu.br/attachments/approvals/26/GP\\_COINT\\_2015\\_2\\_09\\_PROPOSTA.pdf?1455717641](http://tcc.tsi.gp.utfpr.edu.br/attachments/approvals/26/GP_COINT_2015_2_09_PROPOSTA.pdf?1455717641)> Acesso em: 26 mai 2017.
- HENRIQUE, Mychelline Souto; TEDESCO, Patrícia C. de A. R.. Uma Revisão sistemática da Literatura sobre conhecimentos, habilidades, atitudes e competências



desejáveis para auxiliar a aprendizagem de programação. VI Congresso Brasileiro de Informática na Educação (CBIE 2017). Anais dos Workshops do VI Congresso Brasileiro de Informática na Educação (WCBIE 2017). Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/7505/5300>>. Acesso em: 20 dez 2017.

IEPSEN, Edécio Fernando. Ensino de algoritmos: detecção do estado afetivo de frustração para Apoio ao processo de ensino e aprendizagem. Porto Alegre, 2013.

MEC. Ministério da Educação. Disponível em:<<https://www.mec.gov.br/>>. Acesso em: 13 ago de 2016.

NOSCHANG, Luiz F. ; PELZ, Fillipi; JESUS, Elieser A. ; RAABE, André L. A. Portugol Studio: Uma IDE para Iniciantes em Programação. XXXIV Congresso da Sociedade Brasileira de Computação - CSBC 2014. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2014/001.pdf> > Acesso em 24 mai 2017.

RAMOS, Vinicius; FREITAS, Mateus; GALIMBERT, Maurício. A Comparação da Realidade Mundial do Ensino de Programação para Iniciantes com a Realidade Nacional: revisão sistemática da literatura em eventos brasileiros. Florianópolis, 2015. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/5178>>. Acesso em: 25 out. 2016.

RoboMind. Disponível em: <<https://www.robomind.com.br/>> Acesso em: 26 mai 2017.

SOUZA, Fábio. Embrio: Uma ferramenta de programação visual para Arduino. 09 abr. 2015. Disponível em: <<https://www.embarcados.com.br/embrio-uma-ferramenta-de-programacao-visual-para-arduino/>> Acesso em: 26 mai 2017.