



COBENGE

2018

XLVI Congresso Brasileiro de Educação em Engenharia e 1º Simpósio Internacional de Educação em Engenharia

03 a 06 de setembro de 2018
SALVADOR / BA

"Educação inovadora para uma Engenharia sustentável"



BR.INO: UMA FERRAMENTA PARA ENSINO DE PROGRAMAÇÃO EM ARDUINO PARA APLICAÇÕES EM ROBÓTICA USANDO LINGUAGEM NATIVA

Carlos Humberto Llanos Quitero - llanos@unb.br

Gabriel Rodrigues Pacheco - gabrielrodriguespacheco2@gmail.com

Mateus Berardo de Souza Terra - mateus.b.s.terra@gmail.com

Rafael Mascarenhas Dal Moro - rafaelmdalmoro@gmail.com

Víctor Rodrigues Pacheco - victorrpacheco98@gmail.com

Universidade de Brasília - Asa norte, Campus Darcy Ribeiro

70910-900 - Brasília - Distrito Federal

Resumo: Este artigo propõe a implantação da Robótica Educacional, utilizando linguagem de programação para Arduino estruturada em português, chamada Br.ino, como uma ferramenta alternativa para o aprimoramento do ensino de computação básica. O objetivo principal deste estudo é apresentar a inovadora plataforma citada, contextualizando sua importância para disciplinas introdutórias de programação, bem como os resultados positivos que a aplicação prática, em sistemas reais, dos conteúdos estudados em ditas matérias. Ademais, são introduzidos os conceitos da plataforma Arduino e compiladores para essa plataforma. Por fim, propõe-se uma metodologia de implantação do modelo de ensino enunciado, avaliando seus benefícios e os benefícios deste são avaliados.

Palavras-chave: Arduino. Robótica. Programação. IDE. Ensino.

1 INTRODUÇÃO

A educação brasileira apresenta, em cursos superiores relacionados à computação, sérios índices de reprovação em disciplinas de introdução à programação. De acordo com (CANEDO; SANTOS; LEITE, 2018), nos anos de 2009 a 2012, 49% dos estudantes

Organização:



Realização:



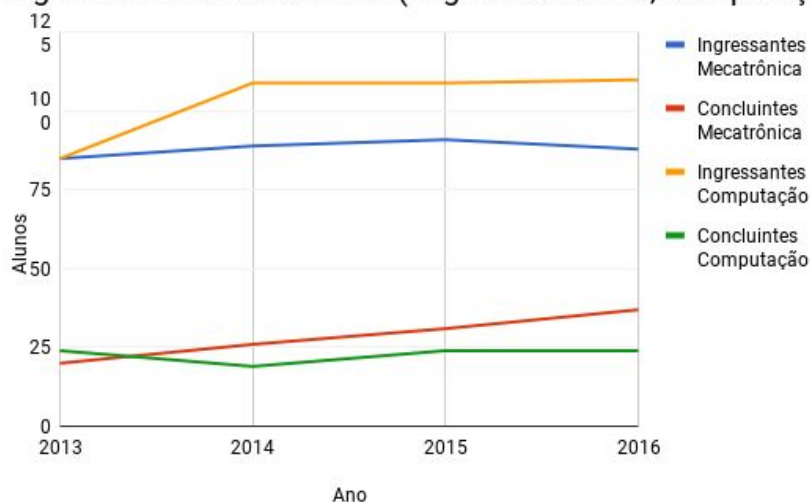
matriculados nessas disciplinas não obtiveram êxito. Contudo, esta habilidade é uma tarefa primordial a diversos cursos de tecnologia.

A Tabela 1 exibe o número de alunos da Universidade de Brasília (UnB) inscritos, por semestre, nas matérias de introdução à programação de acordo com levantamento realizado junto à Secretaria de Administração Acadêmica (SAA) incluindo as matérias de Algoritmos e Programação de Computadores (APC), Introdução à Ciência da Computação (ICC) e Computação Básica (CB). Os dados apresentados reforçam a ideia de que a taxa de reprovação tem assumido níveis preocupantes.

Além disso, as taxas de evasão são problemáticas. Segundo levantamento do Decanato de Planejamento, Orçamento e Avaliação Institucional da UnB, ao analisar o número de estudantes ingressados e de estudantes concluintes nos cursos de Ciência da Computação, Licenciatura em Computação e de Engenharia Mecatrônica entre 2013 e 2016, vê-se que a média de não conclusão do curso (desse intervalo) representa 69,3% dos estudantes de Engenharia Mecatrônica e 77,9% dos estudantes de Computação, como mostrado no Gráfico 1.

Gráfico 1 - Relação de ingressantes/Concluintes em Eng. Mecatrônica e Computação

Ingressantes X Concluintes (Eng. Mecatrônica, Computação L.)



Fonte: Gerado a partir de dados do relatório do Decanato de Planejamento, Orçamento e Avaliação Institucional (2018)

Tabela 1 - Índice de aproveitamento em matérias de introdução à programação (APC, ICC, CB)

Turmas de introdução à programação da UnB	Alunos matriculados	Percentual de não concluintes	Número de alunos não concluintes
2015	1788	40%	709
2016	2000	38%	755
2017	2263	48%	1088

Total:	6051	42%	2484
--------	------	-----	------

Fonte: Gerado a partir de levantamento da SAA

Um ponto importante no ensino de programação dentro da engenharia é que, em cursos com foco em controle e automação, é importantíssimo que alunos novatos possam ter um contato efetivo com plataformas para sistemas embarcados, como o Arduino. Neste contexto, estes podem ser definidos como sistemas computacionais de uso específico, que devem atender sérias restrições tais como: (a) baixo consumo de potência/energia (que restringem taxas de *clock* à ordem de MHz), (b) desempenho apropriado para o(s) algoritmo(s) implementados, (c) tamanho pequeno e (d) baixo custo.

Considerando o cenário apresentado, vê-se que a introdução de sistemas embarcados para alunos do primeiro semestre de cursos de automação e controle por meio de sistemas robóticos é um modelo que estimula o estudante a aprender, que lhe permite absorver conhecimentos com atividades práticas e que, possivelmente, amplia a taxa de permanência da graduação. Ademais, o ensino de programação para alunos calouros pode se beneficiar com a introdução precoce de plataformas como o Arduino, principalmente, se oferecidas de forma amigável e intuitiva.

Com base nessa perspectiva, foi criado o Br.ino, como uma nova ferramenta que pode facilitar a aquisição do domínio da lógica de programação e introduzir o uso de sistemas embarcados. Esta ferramenta baseia-se em uma linguagem de programação em português, que se destaca de outras linguagens como o Portugol por ser aplicável à robótica, uma grande ferramenta para o ensino. Portanto, pode ser usada em projetos práticos e materiais, corroborando para a motivação nos alunos.

Um exemplo de aplicação dessa ferramenta é a disciplina de Introdução à Engenharia Mecatrônica, ofertada na UnB a alunos de primeiro semestre do curso. Neste contexto, foi proposto o uso da ferramenta de programação citada a fim de auxiliar na elaboração de projetos envolvendo temas de robótica, automação e controle de pequenos processos. Na nossa experiência, essa plataforma tem se mostrado eficiente, ampliando o interesse pela área de atuação, assim como aumentando a dedicação dos alunos.

No contexto de ensino para a programação o Prof. Reiner Hartenstein (HARTENSTEIN, 2008) compara as dificuldades específicas do paradigma de von Neumann (tais como *memory wall* e *power wall*) com a barreira educacional no ensino de computação (denominada pelo autor de *education wall*). Neste sentido, este trabalho visa aliviar as dificuldades do ensino da computação mediante a prática simultânea da lógica de programação com aplicações imediatas, altamente motivadores, como a área de robótica, controle e automação, usando uma ferramenta que faz ênfase na intuição.

2 FUNDAMENTAÇÃO TEÓRICA

No contexto da abordagem do ensino de programação para alunos de engenharia sabe-se que o ensino de Lógica de Programação depende fortemente da capacidade de abstração. Neste sentido, Cambruzzi e Souza (2015) indicam que a habilidade de programar depende do desenvolvimento de vários níveis de abstração por parte do alunos. Os resultados

Organização:



Realização:





obtidos por eles indicam que a utilização de uma metodologia mais lúdica e interativa (como a Robótica Educativa) colabora neste sentido. A introdução destes métodos elevou o desempenho em relação ao grupo de referência, em especial para problemas complexos. Além disso, as turmas que participaram das atividades com a robótica afirmaram ter maior motivação e maior compreensão dos conceitos trabalhados na disciplina (CAMBRUZZI; SOUZA, 2015).

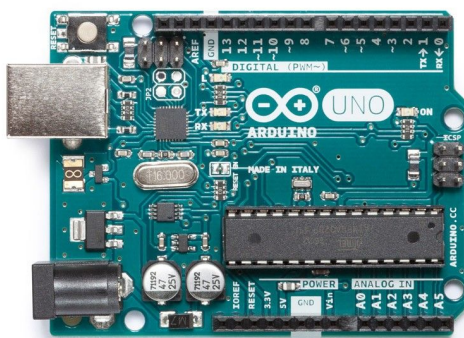
A partir deste estudo, pode se afirmar que a Robótica Educativa contribui significativamente em duas frentes: (a) no aumento da aprendizagem dos alunos e sua percepção dos conhecimentos adquiridos e (b) no interesse de alunos nos cursos de computação e redução das taxas de evasão destes (CAMBRUZZI; SOUZA, 2015). Esta afirmação fica ainda mais potencializada no contexto do ensino de programação para alunos de cursos de automação e controle, mecatrônica e áreas correlatas.

3 ARDUINO E TRABALHOS CORRELATOS

3.1 A placa Arduino

O Arduino é uma placa de prototipação eletrônica para sistemas embarcados com um microcontrolador Atmel AVR, capaz de interagir com o ambiente que o cerca por meio de entradas e saídas analógicas e digitais. Criada para projetos escolares ou de design, é uma plataforma de *hardware livre*, de baixo custo e versátil, ideal para pessoas com pouca ou nenhuma experiência. Como exemplo, é mostrada a placa Arduino Uno (modelo mais popular entre as variantes) na Figura 1.

Figura 1 - Placa Arduino UNO



Fonte: <https://store.arduino.cc/usa/arduino-uno-rev3>

3.2 Trabalhos correlatos

Atualmente, existem diversas plataformas para programar o Arduino, sendo a maioria delas caracterizadas pelo uso da programação na língua inglesa. Algumas delas contam com a possibilidade de usar uma interface de desenvolvimento com as opções traduzidas (embora o código-fonte seja em inglês), ou de programar por meio de blocos visuais (alguns destes são traduzidos). Ao se observar a relação de interfaces disponibilizada na Tabela 2 fica evidente



que a filosofia *open-source* é dominante na comunidade do Arduino. A seguir são brevemente discutidos as características destas plataformas de apoio.

Tabela 2 - Relação com diversas IDEs e suas principais características.

Nome da IDE	Linguagem em linhas ou blocos?	Interface em Português?	Língua	Pago/Gratuito	Open Source
Ardublock	Blocos	Sim	Inglês/Português(limitado)	Gratuito	Sim
Arduino	Linhas	Sim	Inglês	Gratuito	Sim
Brasilino	Linhas	Não tem interface	Português	Gratuito	Sim
Br.ino	Linhas	Sim	Português/Inglês	Gratuito	Sim
Embrio	Blocos	Não	Inglês	Pago	Não
S4A (<i>Scratch for Arduino</i>)	Blocos	Não	Inglês/Português(limitado)	Gratuito	Sim
Programino IDE	Linhas	Não	Inglês	Pago	Não

Fonte: Gerado a partir de dados obtidos de <https://playground.arduino.cc/Main/DevelopmentTools>

- **Ardublock:** É um plugin para a IDE do Arduino para desenvolvimento em blocos virtuais. A ferramenta gera o código textual baseado na montagem criada pelo programador, auxiliando também no aprendizado da linguagem textual. Também está disponível, com menos blocos, em português.
- **Arduino IDE:** É um software desenvolvido em Java. Esquematizado para introduzir a programação a pessoas leigas. Inclui um editor de código capaz de compilar e carregar programas para a placa com um único clique. O Software é *open-source* e disponibiliza diversos plugins para integrar mais funcionalidade a ele.
- **Brasilino:** É uma biblioteca para o Arduino que pode ser instalada sem dificuldade pelo próprio gerenciador de bibliotecas da IDE padrão. Permite programar em linguagem *Wiring* utilizando comandos facilitados em português.
- **Embrio:** É uma IDE para desenvolvimento misto(blocos e linhas) que busca repartir o processo de desenvolvimento e aproveitar as vantagens de cada abordagem para desenvolvimento de softwares mais complexos e/ou aprendizado. A ferramenta não é *open-source* e é paga, porém possui uma versão gratuita com limite de operações.
- **S4A (*Scratch for Arduino*):** É um plugin para o programa Scratch, usado principalmente para desenvolvimento de jogos utilizando blocos, que adiciona componentes para o controle do Arduino, quando este está conectado ao computador.
- **Programino IDE:** É uma IDE para desenvolvimento na linguagem Arduino com mais recursos. Ela traz ferramentas para melhor visualização de dados, como matrizes, e para personalização da IDE.

No contexto das plataformas supracitadas pode ser observado uma dificuldade na interface, especificamente para estudantes novatos falantes de língua portuguesa, que permitam o ensino e aplicação da programação de maneira intuitiva.

4 A PROPOSTA DO BR.INO

Tendo em vista o contexto enunciado neste artigo e as ferramentas correlatas descritas, que auxiliam de diversas formas o desenvolvimento de algoritmos para a plataforma Arduino, propõe-se uma ferramenta alternativa que facilitará ainda mais o ensino introdutório e possibilitará o uso da robótica de forma precoce com os calouros de cursos voltados à tecnologia. Tal ferramenta consiste em uma linguagem de programação estruturada em português, com sintaxe semelhante ao C chamada *Br.ino*.

Figura 2 - Logo da empresa Br.ino



Fonte: <http://brino.cc>

A Tabela 3 traz uma comparação entre um código escrito nesta linguagem e na linguagem do Arduino. É possível perceber como a sintaxe de ambas é muito próxima e que os termos escolhidos foram o mais próximo possível de traduções literais, salvo onde fosse possível buscar outras que agregassem mais sentido para o código.

Tabela 3 - Comparação entre a linguagem Br.ino e Arduino (Código piscar/blink)

Br.ino	Arduino (<i>Wiring</i>)
<pre>Numero LED = 13; Configuracao() { Pino.definirModo(LED, Saida); } Principal() { Pino.ligar(LED); esperar(1000); Pino.desligar(LED); esperar(1000); }</pre>	<pre>int LED = 13; void setup() { pinMode(LED, OUTPUT); } void loop() { digitalWrite(LED, HIGH); delay(1000); digitalWrite(LED, LOW); delay(1000); }</pre>



Fonte: Br.ino IDE - disponível em <http://brino.cc>

O Br.ino não se propõe a utilizar a programação em blocos, pois embora seja uma ferramenta valiosa para o desenvolvimento do pensamento computacional, plataformas de blocos visuais afastam o programador de linguagens utilizadas profissionalmente tais como o C, Python e Java, restringindo o desenvolvedor a casos de usos disponibilizados pelo projetista dos blocos. Esta nova linguagem busca aproximar sua sintaxe ao C/C++, visto que são utilizadas pelo Arduino. Além disso, tais linguagens são amplamente utilizadas em cursos introdutórios à programação assim como no mercado de trabalho.

Com o uso dessa ferramenta, propõe-se uma metodologia para disciplinas de introdução à programação visando maximizar o interesse do aluno, bem como facilitar sua compreensão da lógica, potencializando o desenvolvimento do pensamento lógico. Essa cronologia segue etapas definidas abaixo, a fim de serem realizadas no período de um semestre:

- (a) **Entendimento Inicial:** Neste estágio o aluno é introduzido ao funcionamento do computador, possivelmente com algumas aulas sobre a história da computação. Após estas primeiras aulas, é introduzido o conceito de fluxogramas; contudo, não de maneira teórica, mas sim em pequenos problemas práticos. Dessa forma, o aluno poderá se familiarizar com o processamento de instruções e com o desenvolvimento de algoritmos bem definidos.
- (b) **Introdução da Programação e Robótica:** Nesse estágio, são apresentados ao aluno o Arduino e o Br.ino por meio de problemas do cotidiano que podem ser solucionados, em escala, por estas ferramentas, como por exemplo um semáforo de trânsito ou um elevador de carga. Tais problemas inserem conceitos como *loops* de repetição, laços condicionais e variáveis, que compõem a base da programação. Estes devem ser apresentados de forma prática, podendo ser paralelamente abordados de forma teórica.
- (c) **Introdução de bibliotecas:** Neste estágio, o aluno está familiarizado com os conceitos básicos e, portanto, são necessários desafios além dos iniciais. Com isso, novos componentes com métodos de controle mais elaborados são introduzidos. Para facilitar a integração destes, são apresentadas bibliotecas de código da comunidade do Arduino, desenvolvidas em inglês. Dessa forma, o aluno começará a lidar com ambas as línguas e fará uma transição gradual para a linguagem nativa do Arduino por meio da inserção de funções externas e da comparação do código desenvolvido por ele e da tradução disponibilizada pela IDE do Br.ino.
- (d) **Desenvolvimento:** Nesta etapa é esperado que o aluno domine a linguagem C, adquirida por meio da transição do Br.ino para o Arduino, e que tenha desenvolvido o pensamento computacional. A partir desse momento, a robótica será deixada de lado como foco principal para se focar no desenvolvimento de outros conceitos aplicados comumente ao desenvolvimento de sistemas, tais como ponteiros, manipulação de arquivos, entrada e saída padrão, entre outros.

Organização:



Realização:



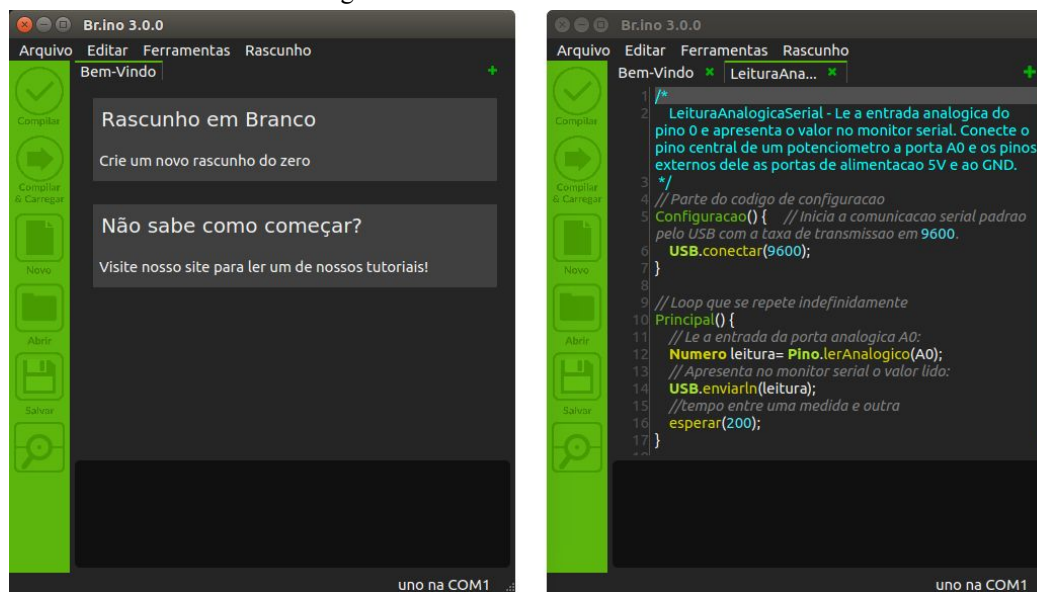
5 RESULTADOS

O Ambiente de Desenvolvimento Integrado (*IDE*) do Br.ino foi desenvolvido em Python utilizando o *port* do Qt *PyQt* na versão 5.10. A interface do programa (criada por um designer) visa tornar o ambiente mais intuitivo (ao posicionar os principais comando próximos à área de texto com legendas claras) e mais confortável para aqueles que programam durante períodos prolongados (por meio de um tema escuro, menos agressivo aos olhos).

A linguagem de programação Br.ino possui termos próximos a traduções literais dos comando em *Wiring*, de forma a buscar uma aproximação desta, porém sem perder o sentido para os leitores nativos. Como exemplo, podemos citar a tradução do *setup*, que equivale ao método *Configuracao*. A fim de adotar boas práticas de programação os acentos e caracteres especiais (como o cedilha) foram removidos.

Para compilar o código, primeiro é realizada uma tradução do código-fonte desenvolvido, para que, em seguida, este seja compilado pela ferramenta *arduino-builder* também utilizada pela IDE original. Devido a esta arquitetura, é possível mesclar código em *Wiring* com código em Br.ino e utilizar bibliotecas previamente desenvolvidas pela comunidade do Arduino. Dessa forma, tal linguagem não está em desvantagem quando comparada com sua contrapartida, permitindo o aprendizado de forma mais simples por traduzir termos essenciais e aproximar o algoritmo do entendimento do programador.

Figura 3 - Interface de Usuário da IDE do Br.ino



Fonte: Acervo Próprio

Além disso, o código traduzido é disponibilizado ao desenvolvedor, possibilitando também que a ferramenta seja uma aliada à aprendizagem de inglês e facilitando a transição para outras linguagens quando assim for desejável ao curso.



6 CONCLUSÕES

Tendo em vista os dados apresentados é evidente a necessidade de mudanças na forma como as matérias de programação são introduzidas aos alunos do primeiro semestre de engenharias e computação, a fim de refrear problemas sociais vinculados à evasão e, conseqüentemente, reprimir o desperdício de recursos governamentais.

É necessário trazer para dentro de sala a ideia do “mundo maker” permitindo aos alunos aprenderem fazendo e, a partir deste, correlacionar com a teoria, pois desta forma a assimilação dos conceitos se torna muito mais clara, e o aluno passa a não ver o conteúdo como dados, mas como informações e ferramentas. Neste contexto, o Br.ino se apresenta como intermediário ideal, introduzindo no cenário nacional uma ferramenta a ser à sala de aula. Os resultados do uso do Br.ino na disciplina de Introdução à Engenharia Mecatrônica são muito estimulantes, em que pode ser percebido um aumento na taxa de aprendizagem da lógica de programação, sustentada pelo projeto rápido de pequenos sistemas embarcados, baseados no Arduino.

REFERÊNCIAS

Internet:

Arduino (2018). Disponível em: <http://www.arduino.cc/> . Acesso em 9 de Maio de 2018.

Arduino (2018). Disponível em: <https://pt.wikipedia.org/wiki/Arduino> . Acesso em 9 de Maio

Br.ino (2018). Disponível em: <http://brino.cc/> . Acesso em 9 de Maio de 2018.

Decanato de Planejamento, Orçamento e Avaliação Institucional (2018). Disponível em: http://www.dpo.unb.br/index.php?option=com_phocadownload&view=category&id=57:folderr-unb-em-numeros&Itemid=676 . Acesso em 10 de Maio de 2018.

HARTENSTEIN, R. The von Neumann Syndrome (2008), Disponível em: <https://pdfs.semanticscholar.org/a5ab/9efba44aea53c1abc56bb79f53b3c630aea0.pdf>.

Artigos de periódicos:

CANEDO, E. Dias; SANTOS, G. Almeida e LEITE, L. Lopes. An Assessment of the Teaching-Learning Methodologies Used in the Introductory Programming Courses at a Brazilian University. **Informatics in Education - An International Journal**, Vilnius University Institute of Data Science and Digital Technologies, v.1, n.1, pp. 45-59, 2018. Acesso em 10 de Maio de 2018.



COBENGE

2018

XLVI Congresso Brasileiro
de Educação em Engenharia
e 1º Simpósio Internacional
de Educação em Engenharia

03 a 06 de setembro de 2018
SALVADOR / BA

"Educação inovadora
para uma Engenharia sustentável"



Trabalhos em Eventos

CAMBRUZZI, Eduardo; SOUZA, Rosemberg M. Robótica Educativa na aprendizagem de Lógica de Programação: Aplicação e análise. In: Congresso Brasileiro de Informática na Educação/Conferência Latino-americana de Objetos e Tecnologias de Aprendizagem, 2015, Maceió. **Anais**. Bahia, 2015.

BR.INO: A TOOL FOR TEACHING ARDUINO PROGRAMMING FOR ROBOTICS APPLICATIONS USING NATIVE LANGUAGE

Abstract: *This article proposes the implementation of Educational Robotics, using an Arduino programming language structured in Portuguese, called Br.ino, as an alternative tool for the improvement of teaching of basic computing. The main objective of this study is to present the innovative platform mentioned, contextualizing its importance for introductory programming disciplines, as well as the positive results that the practical application, in physical systems, of the contents studied by these subjects has. In addition, the Arduino board concept and Arduino programming platforms are introduced. Finally, it is proposed a methodology for the implementation of the teaching model and the benefits of this are evaluated.*

Key-words: *Arduino. Robotics. Programming. IDE. Teaching.*

Organização:



Realização:

