



HOLOS

ISSN: 1518-1634

holos@ifrn.edu.br

Instituto Federal de Educação, Ciência e
Tecnologia do Rio Grande do Norte
Brasil

TORRES, V. P.; AROCA, R. V.; BURLAMAQUI, A. F.
AMBIENTE DE PROGRAMAÇÃO BASEADO NA WEB PARA ROBÓTICA
EDUCACIONAL DE BAIXO CUSTO
HOLOS, vol. 5, 2014, pp. 252-259
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
Natal, Brasil

Disponível em: <http://www.redalyc.org/articulo.oa?id=481547174023>

- Como citar este artigo
- Número completo
- Mais artigos
- Home da revista no Redalyc

redalyc.org

Sistema de Informação Científica
Rede de Revistas Científicas da América Latina, Caribe, Espanha e Portugal
Projeto acadêmico sem fins lucrativos desenvolvido no âmbito da iniciativa Acesso Aberto

AMBIENTE DE PROGRAMAÇÃO BASEADO NA WEB PARA ROBÓTICA EDUCACIONAL DE BAIXO CUSTO

V. P. TORRES^{1*}, R. V. AROCA¹ e A. F. BURLAMAQUI²

¹Instituto Metrópole Digital - UFRN

²Escola de Ciências e Tecnologia – UFRN

vpaivatorres@gmail.com*

Artigo submetido em janeiro/2014 e aceito em julho/2014

DOI: 10.15628/holos.2014.1902

RESUMO

Este artigo descreve um ambiente de programação multiplataforma baseado na web que permite que usuários escolham entre a programação escrita tradicional e o método didático de programação em blocos. O objetivo desse trabalho é facilitar a maneira como estudantes aprendem a programar sistemas

embarcados utilizando hardware aberto e de baixo custo, como o Arduino Uno, além de métodos modernos e diferentes abordagens de programação. Os resultados foram obtidos e validados através de testes práticos com estudantes universitários. Eles aprovaram o sistema como uma alternativa mais didática.

PALAVRAS-CHAVE: Robótica Educacional, Ambiente Web, Programação Visual.

WEB BASED PROGRAMMING ENVIRONMENT FOR LOW-COST EDUCATIONAL ROBOTICS

ABSTRACT

This paper describes a multi-platform web based programming environment that allows users to choose between classic textual programming and the didactic method of block programming. The objective of this work is to ease the way students learn how to program embedded systems using low cost open-source

hardware, such as Arduino Uno, modern methods and different programming approaches. The results were validated through practical tests with university students. They have approved the system as a more didactic alternative.

KEYWORDS: Educational Robotics, Web Environment, Visual Programming.

1 INTRODUÇÃO

Existem diversas plataformas de desenvolvimento para sistemas embarcados e robótica educacional disponíveis na Internet. É comum encontrar editores de código com destaque de sintaxe para linguagens tradicionais, como C/C++ e Java. Entretanto, ainda não existe uma ferramenta baseada na web direcionada para robótica educacional que contemple envio de código para os dispositivos e programação visual em blocos quando se considera plataformas *open-source* e *open-hardware*. Algumas soluções se aproximam em um certo quesito, mas em outros deixam a desejar [9], como é o caso do CodeBender [1], que não possui edição em blocos, e do MIT App Inventor [2], que possui edição em blocos, mas é direcionado a produção de aplicações Android e requer a compilação, transferência e instalação do arquivo a cada modificação do programa.

Este artigo descreve um ambiente de programação multiplataforma baseado na web, o qual permite que usuários escolham entre a programação tradicional (textual) e o método didático de programação gráfica/visual em blocos. O objetivo é facilitar a maneira como estudantes aprendem a programar robôs e sistemas embarcados em geral através da adoção de hardware aberto e de baixo custo, como Arduino Uno, N-bot [3], entre outros; e a utilização de uma plataforma de desenvolvimento fácil de usar e dotada de recursos didáticos importantes para a introdução de alunos iniciantes.

O sistema pode ser utilizado como ferramenta de suporte à robótica educacional em diversos projetos, incluindo o projeto Um Robô por Aluno (URA) [4], o qual já está em fase de execução com aulas em algumas escolas públicas da região metropolitana de Natal. Existe também uma versão similar do ambiente, mantido pela empresa Roboeduc, também na área de robótica educacional, mas voltada ao setor privado. Porém, os projetos não permanecem isolados. Acontece uma troca de tecnologias entre as iniciativas, resultando em um retorno social de grande interesse aos dois. Poderemos então avaliar o comportamento da ferramenta em casos reais com diferentes turmas de faixa etária e conhecimento variados.

2 TRABALHOS RELACIONADOS

O App Inventor, mantido pelo Massachusetts Institute of Technology (MIT) [1], foi disponibilizado em 2010 pela Google e abriu caminho na educação computacional pesquisando sobre maneiras didáticas de se construir aplicativos para o sistema operacional móvel Android. O sistema, baseado na web, permite que qualquer usuário com experiência básica em computação crie uma aplicação capaz de rodar no sistema operacional por meio de programação visual em blocos. Agora disponível em sua segunda versão, o sistema funciona inteiramente via web, sem mais a necessidade de executar uma aplicação Java para o editor de blocos.

Desenvolvido por uma equipe da Universidade de Patras, Grécia [2], o CodeBender é uma aplicação web tornada pública em meados de 2012 com o objetivo de ajudar o usuário a escrever e enviar seu programa para uma plataforma Arduino, totalmente via navegador web. Conta com editor de código tradicional com destaque de sintaxe, indentação e contagem de linhas. Uma inovação do sistema é a possibilidade de envio do programa compilado via web diretamente para um Arduino, sem a necessidade de instalação de nenhum software adicional.

Não se deve esquecer do Ambiente de Desenvolvimento Web do N-Bot (ADWN) [3], desenvolvido para atuar juntamente com o robô de baixo custo N-bot – ganhador do desafio internacional *10-dollars robot design challenge* na African Robots Network (AFRON) e exposto na Berkeley University, nos Estados Unidos. Ele funciona da seguinte maneira: o usuário descreve um programa através de blocos na tela do navegador web e tons DTMF são gerados e depois são decodificados por um circuito na placa do próprio robô. A conexão é feita através de cabos de áudio (mesmo conectores de fone-de-ouvido, possibilitando inclusive a utilização de celulares legados para o controle do aparelho, por meio de ligações telefônicas ou gravações de áudio de comandos DTMF gerados anteriormente) entre o N-bot e o computador do usuário. Um ponto positivo é que a plataforma de programação pode ser usada em qualquer computador, sem a necessidade de privilégios administrativos ou hardware adicional. Só é preciso um *browser* que ofereça suporte ao JavaScript.

Aparelhos Android também possuem mais uma opção para programação, o Android Web Based Integrated Development Environment (anwide) [5]. Ele pode ser muito útil na programação de *cellbots* – robôs que utilizam celulares e/ou smartphones como controlador. É um aplicativo que possui um servidor web embarcado e provê ao usuário uma plataforma de programação acessível via navegador web, sem a necessidade de internet, apenas de uma rede local. O anwide faz uso da camada de abstração Scripting Layer for Android (SL4A), uma camada disponível em sistemas Android que permite que *scripts* sejam executados no aparelho e dessa forma, o usuário é capaz de programar um celular ou robô em Python diretamente do seu navegador web.

3 SOLUÇÃO PROPOSTA

Visto a inexistência de uma solução a qual contemplasse todos os melhores pontos disponíveis no atual estado da arte para o ensino didático de sistemas embarcados e robótica educacional, optou-se por mesclar as melhores características dos sistemas já existentes em um só, o qual chamamos de CodeRhino.

Ele possui um editor de código com destaque de sintaxe, indentação e contagem de linhas; editor de programação visual em blocos; e módulo de envio de código. O módulo de envio é uma Java Applet executada no navegador do usuário, a qual baixa o código escrito pelo usuário, compilado em um servidor Linux, e envia, com o auxílio do AVR-DUDE, o programa gerado para o Arduino conectado na máquina. Todo o processo é transparente ao usuário e acontece no navegador web de sua preferência, desde que a Máquina Virtual Java (JVM) e seus respectivos plug-ins estejam instalados e configurados corretamente, tanto no sistema operacional como no browser. Segundo a Oracle, mais de 2 bilhões de dispositivos rodam a plataforma, desde smartphones até servidores dedicados [6]. Bancos e outros sistemas também utilizam a tecnologia como ponte para realizar suas funções de segurança no navegador do cliente. Pode ser necessário a instalação de drivers específicos em alguns computadores e acesso administrativo num primeiro momento em diferentes versões de sistema operacional. A Figura 1 descreve, de maneira simplificada, o funcionamento do sistema na forma de um diagrama de alto nível.

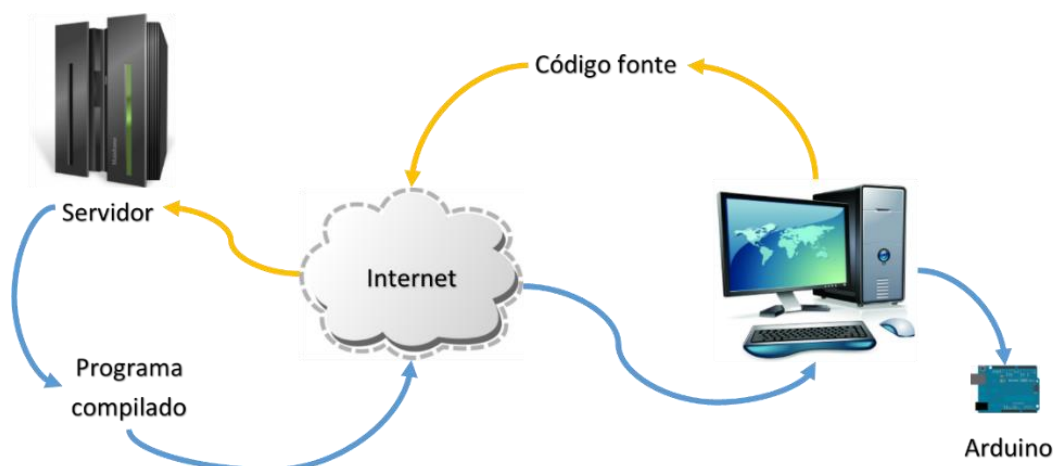


Figura 1 - Diagrama de alto nível do sistema proposto (do amarelo para o azul).

3.1 EDITOR COM DESTAQUE DE SINTAXE

Como o sistema é voltado para web, precisa-se de um editor de código escrito em JavaScript, uma linguagem interpretada e que tem suporte oferecido por todos os navegadores web modernos, como o Google Chrome e o Mozilla Firefox. Apesar de existirem várias opções disponíveis, o editor escolhido foi o Ace, que se equipara em funcionalidade e desempenho a editores nativos famosos como o Sublime (multiplataforma), o Vim (unix-like) e o TextMate (Mac OS X). Além de ser totalmente *open-source*, o Ace também é utilizado na Cloud9 IDE (www.c9.io), um ambiente de desenvolvimento integrado que funciona online, direto do navegador, e oferece suporte a um variado número de linguagens. O site possui um expressivo número de usuários e alguns famosos como o Sound Cloud, Mozilla e Heroku. Ou seja, é uma tecnologia consolidada e validada.

3.2 EDITOR DE BLOCOS

Optou-se pela utilização do Blockly [7] como editor de programação visual em blocos. Um ponto positivo dessa tecnologia é ser utilizada com sucesso na plataforma App Inventor, do MIT, desde 2010 é um ponto positivo para a tecnologia, também escrita em JavaScript. No começo ela era implantada em conjunto com uma aplicação Java, mas agora passou a ser totalmente via *browser* no sistema online de criação de *apps* para Android. Outro fator importante é a existência prévia de um módulo que adiciona suporte a conversão da lógica criada através dos blocos (programação gráfica/visual) para código Arduino (um dialeto da linguagem C/C++), conhecido como BlocklyDuino [8].

3.3 COMPILADOR

A compilação é realizada em um servidor, que roda o sistema operacional livre Linux, através do AVR-GCC, que é o compilador GNU GCC acrescido de algumas bibliotecas relacionadas à plataforma de micro controladores AVR AtMega. Foi feito um *shell script* para que se pudesse incluir no código fonte do programa escrito pelo usuário e/ou gerado pelos blocos lógicos todas as bibliotecas padrões do Arduino, simulando o ambiente de desenvolvimento integrado (IDE) oficial, o qual também faz essa tarefa em *background*, quando clicamos em “compilar” no menu principal.

3.4 ENVIO DO CÓDIGO

O AVR-DUDE é o software responsável pelo envio do programa compilado, em formato hexadecimal, gerado durante a compilação do código fonte, disponível nas plataformas Microsoft Windows, Apple Mac OS e GNU Linux. Uma Java Applet foi desenvolvida para efetuar o *download* do programa gerado pelo compilador e o software de envio (AVR-DUDE) correspondente à plataforma do usuário (o sistema operacional pelo qual o usuário acessa o ambiente de programação). Sendo assim, torna-se necessário que o usuário tenha instalado em seu sistema a plataforma Java, que inclui a máquina virtual (JVM) e os respectivos plug-ins dos navegadores. Grande parte dos sistemas operacionais já possui o Java instalado, seja de fábrica ou por qualquer outra necessidade como Internet Banking, por exemplo. Em alguns computadores, pode ser necessário instalar os drivers da placa Arduino. Na Figura 2, a execução da etapa de envio do código pode ser melhor visualizada:



Figura 2 - Etapas na execução do sistema.

O procedimento final é executado da seguinte forma, conforme a Figura 3:

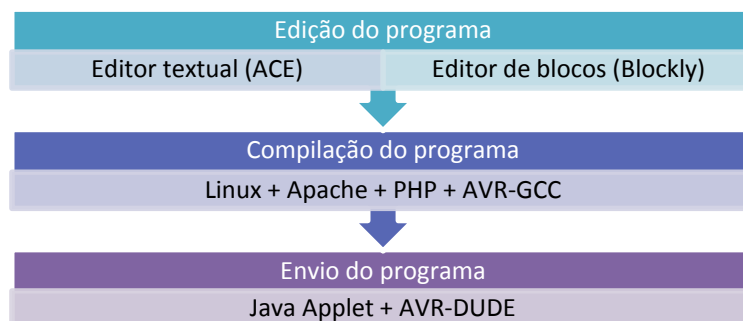


Figura 3 - Etapas na execução do sistema.

3.5 INTERFACE COM O USUÁRIO

Utilizamos o Twitter Bootstrap, um simples, porém poderoso, *front-end framework* para desenvolvimento web, e JavaScript para compor a interface já que o design e usabilidade de uma página é um quesito importante na experiência final do usuário. O *frame-work* em questão foi de grande ajuda nessa parte, permitindo implementar uma interface bonita e usável. Tudo foi agregado em um portal que direciona a dois tipos de programação (Figura 4): avançada, que tem editor de código tradicional com destaque de sintaxe; básica, que permite edição visual didática em blocos lógicos.

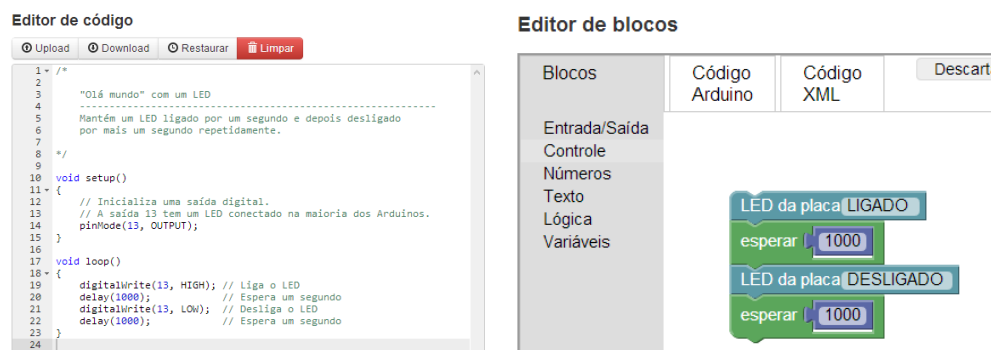


Figura 4 - Editor de código tradicional textual e visual em blocos.

4 RESULTADOS OBTIDOS

Uma turma de 15 alunos do curso de Bacharelado em Tecnologia da Informação, da Universidade Federal do Rio Grande do Norte (UFRN) foi voluntária a participar da validação do sistema. Todos assistiram a uma aula teórica básica de introdução a programação para Arduino, baseada no que pode ser considerado o *hello world* da computação embarcada: o piscar de um LED. Depois, eles receberam a tarefa de implementar o exercício proposto utilizando a programação textual tradicional e depois, utilizando a programação visual em blocos. Ao final das atividades, cada um respondeu anonimamente a um formulário eletrônico para gerar os dados apresentados a seguir, nas Figura 5 e Figura 6, acompanhadas de suas respectivas perguntas.

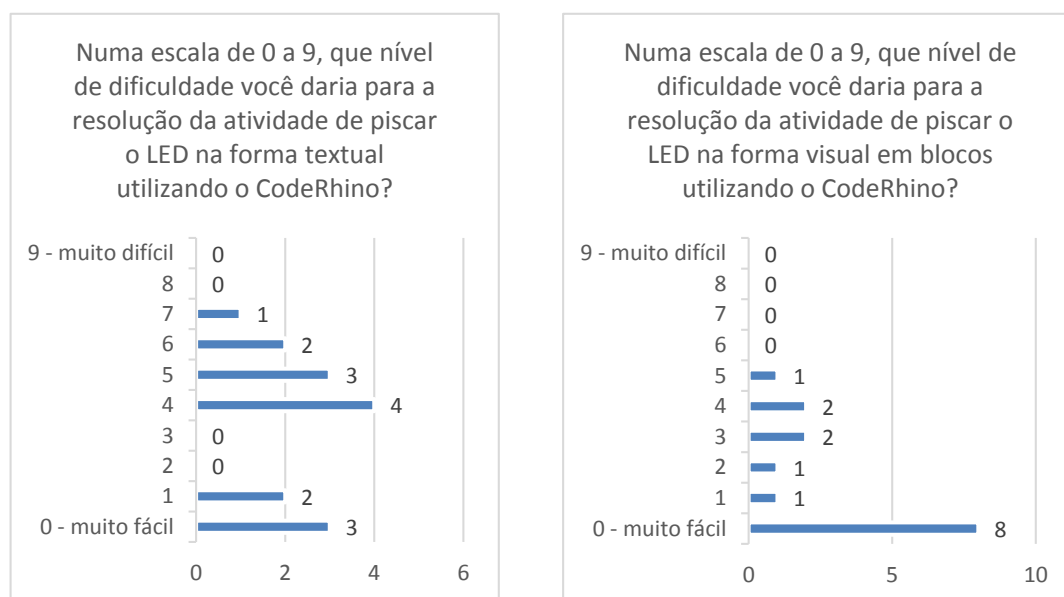


Figura 5 - Respostas quanto ao nível de dificuldade.

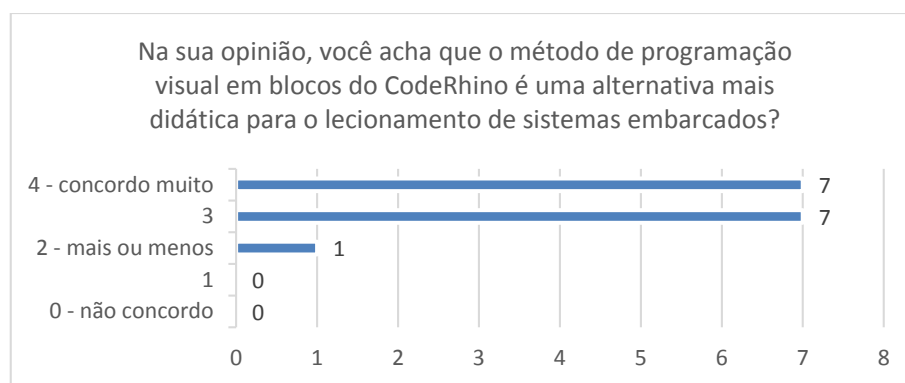


Figura 6 - Opinião geral sobre o sistema.

Outro aspecto observado ao longo do desenvolvimento desse trabalho foi a tendência na utilização de celulares – mais especificamente smartphones – para a realização de tarefas computacionais de variados tipos, incluindo controle de robôs, programação de sistemas embarcados, quando antes se utilizavam computadores comuns ou hardwares dedicados, específicos.

5 CONCLUSÃO

A robótica educacional vem crescendo ao longo do tempo e ganhando importância e destaque notáveis em países de todo o mundo. Apesar de vir ganhando força no Brasil, a realidade daqui ainda é um tanto defasada em relação aos grandes polos tecnológicos europeus e norte-americanos. Um dos fatores que dificulta a adesão da robótica educacional em escolas públicas e para pessoas de baixa renda no país é a baixa inclusão digital. O sistema proposto pode ajudar na solução dessa problemática, pois permite que qualquer computador seja usado para programar e gravar um Arduino sem a necessidade de instalação de nenhum software adicional, todo o processo é feito via web, além de ser uma tecnologia nacional para ambientes educacionais que se propõe a facilitar a compreensão dos alunos iniciantes. Através da edição em blocos, a programação gráfica/visual, podemos trabalhar com vários exercícios e possibilidades, sempre de forma simplificada e intuitiva, sem comprometer o conteúdo original abordado pelo tema.

6 TRABALHOS FUTUROS

Com a solução colocada em prática, esperamos realizar alguns aprimoramentos na mesma. Estamos planejando o desenvolvimento de um sistema de acompanhamento de alunos onde será possível o cadastro e execução de atividades, acompanhamento de notas, correção dos códigos fonte escritos pelos alunos, dicionário de funções de uma determinada linguagem, apostila online para consulta rápida, dentre outros recursos.

O desenvolvimento de programas em assembly (linguagem de *OP codes* mais próxima da linguagem de máquina dos dispositivos) está para ser implementado também na plataforma. Isso facilitaria a programação de baixo nível dos dispositivos para um usuário mais avançado e ajudaria prontamente na disciplina de Organização e Arquitetura de Computadores e suas semelhantes.

Tudo isso deve ser armazenado em um banco de dados para que sejam gerados estatísticas e dados para mais pesquisas relacionadas ao tema.

7 AGRADECIMENTOS

Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

8 REFERÊNCIAS BIBLIOGRÁFICAS

1. CodeBender. Codebender. On-Line. Disponível em: <http://codebender.cc/>. Acesso em Ago/2013.
2. MIT. Explore MIT app inventor. On-Line. Disponível em: <http://appinventor.mit.edu/explore/>. Acesso em Ago/2013.
3. R. V. Aroca, R. Pitta, A. Burlamaqui, and L. M. G. Gonçalves, "N-bot: A low cost educational robot". On-Line. Available at: <http://www.natalnet.br/~aroca/afon/>, 2012. Access date: Jun/2013.
4. R. V. Aroca, R. P. Barros, A. Burlamaqui, and L. M. G. Gonçalves, "Um robô por aluno: uma realidade possível" in Anais do Workshop de Robótica Educacional (Latin American Robotics Symposium). Fortaleza. ISSN 978-85-7669-261-4, 2012.
5. anwide. Android Web Based Integrated Development Environment. On-Line. Disponível em: <https://code.google.com/p/anwide/>. Acesso em Ago/2013.
6. Oracle. Java: learn about Java Technology. On-Line. Disponível em: <http://www.java.com/en/about/>. Acesso em Ago/2013.
7. blockly. A visual programming editor. On-Line. Disponível em: <https://code.google.com/p/blockly/>. Acesso em Ago/2013.
8. BlocklyDuino. A web-based visual programming editor for Arduino. On-Line. Disponível em: <https://github.com/gasolin/BlocklyDuino>. Acesso em Ago/2013.
9. Alves, R. M.; Sampaio, F. F.; Elia, M. F. DuinoBlocks: Um Ambiente de Programação Visual para Robótica Educacional. In: XL Seminário Integrado de Software e Hardware (SEMISH) / XXXIII Congresso da Sociedade Brasileira de Computação (CSBC), 2013, Maceió, AL. Anais do XXXIII Congresso da Sociedade Brasileira de Computação. Porto Alegre, RS: SBC, 2013.