DuinoBlocksII: Uma ferramenta de apoio ao ensino-aprendizagem de lógica de programação e robótica educacional para alunos da rede pública do Ensino Médio

José Augusto M. Vidal[1,3], Fábio Ferrentini Sampaio[1,2], Antoanne Pontes[1,3]

[1]Programa de Pós-Graduação em Informática (PPGI/UFRJ)
[2]Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais (NCE/UFRJ)
[3]Centro de Estudos e Sistemas Aplicados de Recife (CESAR) / Escola Estadual José Leite Lopes (CEJLL)

jamv@ufrj.br, ffs@nce.ufrj.br, antoanne@ufrj.br

Abstract. *This work introduces a visual programming language called DuinoBlocksII, based on Arduino and open-source (and free) technology for educational robotics, that aims to support the teaching-learning programming logic and the awakening of computational thinking. A preliminary study was implemented in a public school in Rio de Janeiro - Brazil during two months in 2017. Some partial results are also presented.*

Resumo. *Este trabalho apresenta o DuinoBlocksII, uma ferramenta de programação visual voltada para a robótica educacional baseado em placas Arduino e de tecnologias livres (e gratuitas). Tem como objetivo o apoio ao ensino-aprendizagem de lógica de programação e o despertar do pensamento computacional. Um estudo preliminar foi implementado numa escola pública na cidade do Rio de Janeiro durante um bimestre de 2017. Alguns resultados parciais também são apresentados.*

1. Introdução

Nos últimos anos temos assistido, em diferentes países, variadas iniciativas que procuraram incentivar o aprendizado de programação entre os estudantes do ensino fundamental e médio. Os argumentos utilizados para tais abordagens são, principalmente, de duas naturezas: aqueles que enxergam a necessidade do ensino de programação como uma importante qualificação para os que estarão entrando no mercado de trabalho nos próximos anos; aqueles que entendem que o ensino de programação implica no desenvolvimento de importantes habilidades e competências necessárias a várias disciplinas do currículo escolar e na solução de problemas do cotidiano. Nesse segundo caso, tais habilidades e competências podem ser englobadas pelo que hoje chamamos de Pensamento Computacional (*Computational Thinking*).

O termo Pensamento Computacional (PC) foi cunhado por Papert (1980) e aperfeiçoado por Wing em artigo seminal publicado na *Communications of ACM* em (2006). No seu trabalho, a autora afirma que pensar de maneira computacional será uma habilidade

fundamental que deveria ser praticado por todos que queiram resolver problemas de maneira sistemática. Barr e Stephenson (2011) listam algumas características sobre o processo, tais como: organizar e analisar dados logicamente, representar dados através de abstrações como modelos ou simulações e automatizar soluções de maneira eficiente tentando generalizá-las para outras variedades de problemas.

No entanto, o ensino de programação não é tarefa fácil, pois como observado por Pontes (2013) e Júnior et al. (2005), o aluno, geralmente novato, depara-se com uma série de problemas durante seu aprendizado (falta de organização do raciocínio, entendimento de questões de lógica relativamente complexos, dificuldade na compreensão do idioma Inglês e a falta de motivação gerada pelo medo prévio da disciplina). Tais problemas estão atrelados a metodologias que enfocam muito mais a memorização de conceitos, em detrimento da prática de programação e resolução de tarefas que fazem parte do cotidiano do aluno, gerando uma grande evasão e repetência nos cursos/disciplinas iniciais de Computação [Ramos e Espadeiro, 2015; Resnick et al., 2009].

Assim, a fim de tentar contribuir com a melhoria do ensino de computação, o Laboratório LIvRE em parceria com o NAVELabs, vêm desenvolvendo ferramentas de software, hardware e práticas pedagógicas que facilitem e ampliem o trabalho de professores e alunos em novos contextos de ensino com tecnologias. O trabalho aqui apresentado trata de uma dessas iniciativas - unindo a Robótica Educacional e o uso da linguagem de programação visual DuinoBlocksII (DBKII) - na formação inicial de programadores participantes de um curso técnico integrado (ensino médio) de desenvolvimento de jogos digitais.

Este trabalho está dividido da seguinte forma: na seção 2 apresentamos, de forma sucinta, o arcabouço teórico no qual está baseada esta presente pesquisa. A seção 3 expõe o ambiente de programação visual baseada em blocos - DuinoBlocksII. A seção 4, por sua vez, descreve o experimento realizado com alunos do 1º ano do Ensino Médio, bem como os resultados parciais obtidos. O trabalho finaliza com a seção 5 onde são apresentadas as conclusões e trabalhos futuros.

2. Arcabouço Teórico

2.1. Robótica Educacional

Um dos aspectos da teoria Construtivista de Jean Piaget (1973) considera a manipulação dos objetos a chave para as crianças construírem seu próprio conhecimento. Por sua vez, o Construcionismo de Seymour Papert (1980) adiciona à teoria de Piaget a ideia de que esta construção se dá de forma mais efetiva quando o aprendiz se engaja de maneira consciente na construção de algo tangível.

Por ser fortemente baseada em atividades práticas e fundamentada na manipulação de objetos físicos que fazem parte da realidade dos estudantes, alguns autores como Benitti (2012), Eguchi (2010) e Campos (2011) defendem o emprego da Robótica Educacional como “porta de entrada” para o ensino de programação.

2.2. Linguagens de Programação Visuais

Diferentes autores como Gomes e outros (2008) e Resnick e outros (2009) argumentam que a utilização de linguagens de programação textuais acrescentam barreiras desnecessárias no ensino inicial de programação. Tais autores defendem ainda alternativas como o uso de Linguagens de Programação Visuais, capazes de encapsular as dificuldades existentes nas sintaxes das linguagens textuais, além de ampliar o alcance do ensinamento de lógica de programação para diferentes idades.

Na seção a seguir apresentamos o DuinoBlocksII, uma Linguagem de Programação Visual baseada em blocos desenvolvida especificamente para a programação de placas de robótica da família Arduino[1].

3. DuinoBlocksII

O DuinoBlocksII foi desenvolvido com o objetivo de facilitar e motivar o ensino de lógica de programação para usuários inexperientes na área de computação. O ambiente permite a implementação de diferentes projetos na modalidade da Robótica Educacional, sendo capaz de controlar a maioria das placas de hardware Arduino existentes no mercado. A versão atual é multiplataforma, podendo ser executada nos principais navegadores web hospedados na máquina do usuário, sem a necessidade de conexão com a internet[2].

3.1. Interface

O ambiente DBKII apresenta ao usuário uma paleta de blocos de comandos organizados em sete categorias (Figura 1), sendo capaz de monitorar diferentes tipos de sensores e atuadores que agirão de alguma forma no mundo físico. São elas:

- *Controle* - possui blocos que controlam o fluxo do programa utilizando-se de estruturas de controle, repetição e *delay*. Desta forma, o projeto pode se comportar de maneira diferente, dependendo do ambiente no qual ele estiver inserido.
- *Operadores* - contém diferentes opções de operadores matemáticos, booleanos e condicionais de comparação, com o intuito de manipular dados captados por sensores e aplicados em atuadores.
- *Entrada/Saída* - inclui os blocos que fazem a interface com o ambiente externo, lendo as informações captadas por sensores e/ou enviando informações para os atuadores através das portas digitais/analógicas que o Arduino oferece.
- *Variáveis* - engloba blocos que manipulam variáveis numéricas e textuais no sistema.
- *Comunicação* - contém blocos capazes de ler e escrever valores na janela de comunicação serial, fazendo um interfaceamento em tempo real entre o experimento e o usuário (p.ex. auxiliando a depuração de programas).

[1] https://www.arduino.cc/en/Guide/Introduction
[2] O ambiente está disponível para download no site do Laboratório LIvRE

- *Matemáticos* - possui as principais funções trigonométricas, além de radiciação, arredondamento, aleatoriedade, etc.
- *Motores* - permite o controle de motores DC e Servo.

O ambiente apresenta ainda, no lado direito da interface, uma área opcional capaz de apresentar o código *Wiring*[3] gerado pelo programa em blocos criado pelo usuário. Desta forma, o educador pode ir inserindo, com cautela, como funciona a programação textual para que o educando, de maneira gradual, vá se familiarizando com a realidade da programação de computadores.

O DBKII disponibiliza outras funcionalidades como verificar código (compilação), salvar e recuperar programas.

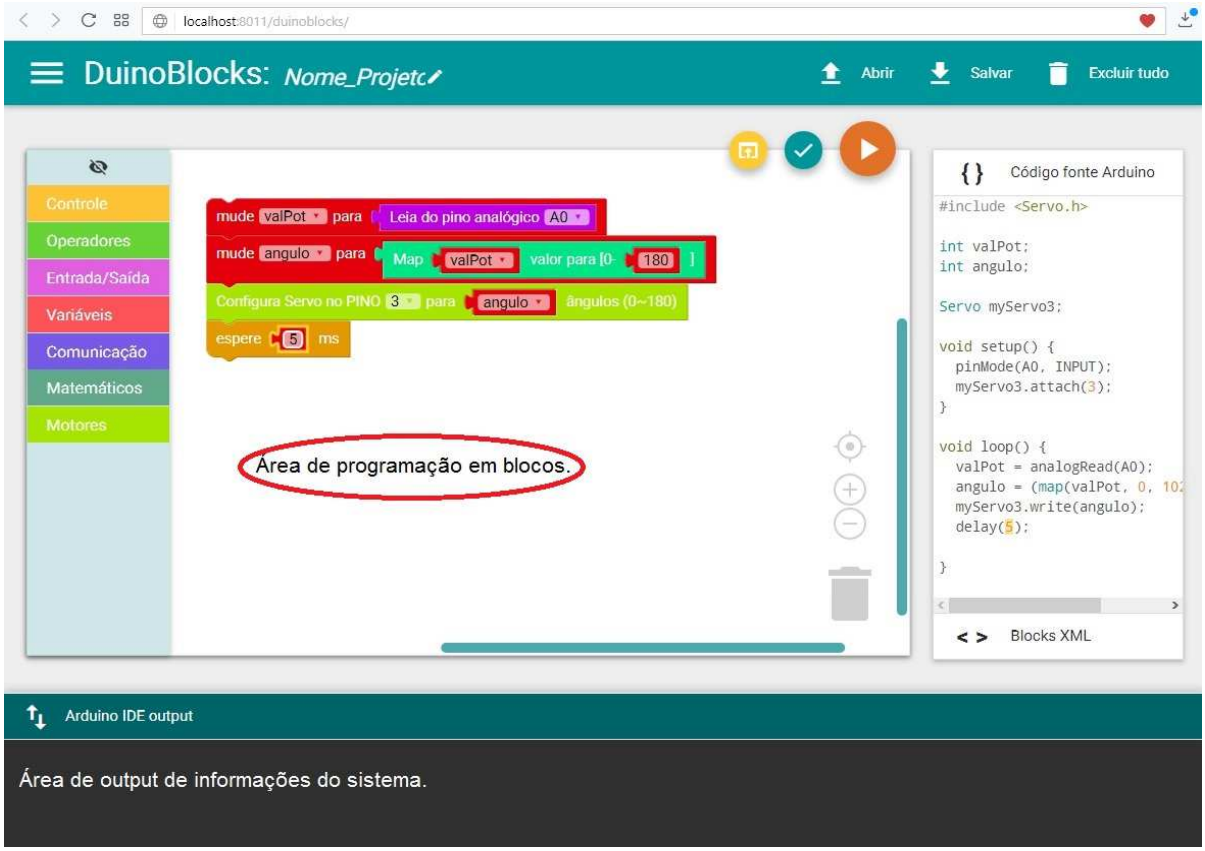

Figura 1. DBKII - Interface do sistema

3.2. Tecnologia Utilizada

O desenvolvimento do DBKII baseou-se na biblioteca de código livre Blockly (*https://developers.google.com/blockly)* desenvolvida pela Google e no ambiente Ardublockly implementado por Carlos Perate[4].

[3] Linguagem textual utilizada pela IDE do Arduino.
[4] Ver *https://github.com/carlosperate/ardublockly.*

A utilização da biblioteca da Google, juntamente com o Ardublocky, permitiu-nos desenvolver de forma relativamente rápida os comandos que compõem a linguagem DBKII, bem como o seu ambiente integrado visualizado pelo usuário final.

O DBKII é executado nos principais navegadores web utilizando a IDE original do Arduino - executada em 2o. plano - para compilar e fazer o upload do código. A comunicação entre a interface do DBKII e a IDE Arduino é feita por um sistema baseado na linguagem Python[5], versão 2.7, que também precisa estar instalada na máquina do usuário.

4. Estudo Realizado e Resultados Preliminares

O estudo preliminar realizado com o DBKII aconteceu no 2º bimestre de 2017 com uma turma de 42 alunos do Ensino Médio (14 a 18 anos), em uma escola pública integral no Estado do Rio de Janeiro.

Uma vez que a escola não dispunha de kits de robótica suficientes para toda a turma, optou-se por utilizar o Tinkercad[6] - uma plataforma web de simulação de circuitos Arduino (Figura 2). Desta forma, os projetos desenvolvidos pelas duplas de alunos seguiam, na maioria das vezes, os seguintes passos: (1) desenho do circuito elétrico do projeto no Tinkercad; (2) programação no ambiente DuinoBlocksII; (3) cópia do código programado para o ambiente Tinkercad e posterior execução; (4) discussão na dupla sobre os resultados gerados/observados na placa Arduino seguido de modificação do código (quando o projeto não funciona como esperado) ou aprimoramento do projeto (aumento da complexidade).

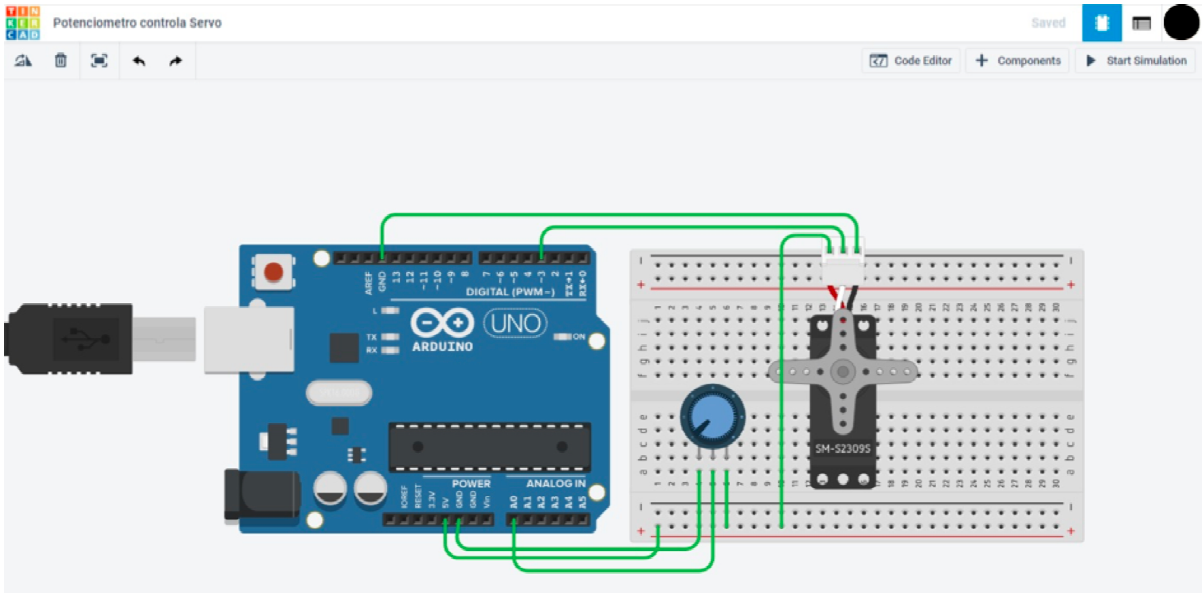

Figura 2. Tinkercad: Simulador web de circuitos eletrônicos

[5] https://www.python.org
[6] https://www.tinkercad.com

Cada aula tinha a duração aproximada de 100 minutos e as atividades propostas eram inspiradas na apostila disponibilizada na compra do kit iniciante de robótica da Robocore[7]. A Tabela 1 apresenta as 20 atividades trabalhadas pelos estudantes.

Aula	Conteúdo	Objetivo
1	Introdução Arduino	Apresentar o Arduino, alguns sensores e atuadores
2	Introdução DuinoBlocksII	Apresentar o DuinoBlocksII e suas características.
3	Introdução Tinkercad	Apresentar o ambiente de simulação de circuitos
4	LED piscando	Ideia de programação sequencial, pinagem (*output*) e *delay* em milissegundos
5	LEDs piscando	Ideia de circuitos em série e em paralelo
6	Exercício	Implementar um circuito que se comporte como um semáforo tradicional
7	Correção + variáveis	*Feedback* do exercício + introdução à variáveis
8	Exercício	Implementar um circuito de pisca-pisca com comportamento Iô-Iô (vai e volta)
9	Correção + exercício	Feedback do exercício + implementar um circuito de pisca-pisca com comportamento aleatório
10	Correção + *inputs*	*Feedback* do exercício + *push button* (*inputs*)
11	Exercício	Implementar um botão acendendo LEDs
12	Estrutura de controle	Apresentar a estrutura de controle *if/else*
13	Operadores	Apresentar operadores lógicos condicionais
14	Exercício	Implementar um botão que controle uma sequência de LEDs
15	Correção + exercício	*Feedback* do exercício + implementar um circuito que possui o comportamento inverso do exercício anterior
16	Correção + portas analógicas	*Feedback* do exercício + apresentar a ideia de intervalo de valores utilizando um potenciômetro
17	Arduino real	Apresentar um kit básico de robótica demonstrando que a simulação é muito parecida com a realidade
18	Comunicação serial	Apresentar o conceito de depuração + *buzzer*
19	Exercício	Potenciômetro controlando servo motor
20	Correção	*Feedback* do exercício

Tabela 1. Tópicos em ordem cronológica dos conteúdos ministrados

[7] https://www.robocore.net/loja/produtos/arduino-kit-iniciante.html

Alguns aspectos relevantes observados no estudo:

- As avaliações foram feitas durante os momentos dos exercícios (vide Tabela 1), sendo propostas como uma atividade prática a ser resolvida pelos alunos em duplas ou triplas.
- A média da turma foi de 6,4 (valores de 0,0 a 10,0) e o desvio padrão foi de 2,17, sendo que 6 alunos foram bastante ausentes durante o bimestre. Retirando estes 6 alunos do cálculo, a média da turma sobe para 6,9 pontos e desvio padrão passa a ser de 1,52 com apenas 4 alunos com nota inferior a 5 pontos. No ano de 2016, turmas equivalentes de programação, ministradas pelo mesmo professor tiveram as seguintes médias:
    - Turma 1004: 41 alunos – média: 6,0 – desvio padrão: 1,93;
    - Turma 1003: 41 alunos – média: 5,6 – desvio padrão: 2,74;
    - Turma 1002: 41 alunos – média: 5,3 – desvio padrão: 2,31;
    - Turma 1001: 40 alunos – média: 5,3 – desvio padrão: 2,08;
- Alguns alunos dominaram rapidamente o processo de programação com blocos e passaram a construir seus programas utilizando a programação textual (*Wiring*), economizando tempo na depuração do código e se familiarizando cada vez mais com os tipos de códigos que eles vão se deparar no futuro em novas disciplinas que irão participar.
- Com relação à interface, todos os educandos souberam utilizar bem suas funcionalidades. No entanto, notou-se alguma dificuldade em visualizar algumas mensagens emitidas pelo DBKII na área de saída.
- Após a finalização do bimestre, muitos alunos que participaram do estudo procuraram a equipe de professores de programação da Escola manifestando o desejo de continuar o desenvolvimento de projetos e estudos com robótica.

5. Conclusões

O DuinoBlocksII é um ambiente de programação visual baseado em blocos que tem por objetivo facilitar, de forma mais prazerosa e engajadora, o aprendizado de lógica de programação e o desenvolvimento do pensamento computacional por alunos do ensino médio e primeiros anos do ciclo básico de graduação.

Os resultados preliminares obtidos através dos trabalhos desenvolvidos, as notas finais dos alunos e o despertar do interesse de alguns deles por programação e robótica, nos permite pensar que o DBKII, juntamente com estratégias de aprendizagem baseada em projetos, têm sido promissores, mas carecendo ainda de mais investigação.

5.1. Trabalhos futuros

É necessário que sejam feitos novos estudos com um controle maior sobre as variáveis observáveis (projeto e sequência pedagógica) a fim de investigar o verdadeiro potencial do DuinoBlocksII e da Robótica Educacional, auxiliando na implementação de diferentes estratégias pedagógicas.

Uma vez que boa parte dos alunos não possui kits de robótica, não foi possível propor atividades extraclasse que motivassem os alunos a avançarem em alguns projetos desenvolvidos. Nesse sentido, a integração do DBKII com o LabVAD (Laboratório Virtual de Atividades Didáticas em Ciência e Robótica[8]) pode ser uma solução para problemas dessa natureza.

Por último, verificamos ser necessária a adição de novos blocos de comandos na ambiente DBKII a fim de torná-lo mais completo, possibilitando a utilização de novos componentes que ajudem no desenvolvimento do Pensamento Computacional dos educandos.

Referências

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?. *Acm Inroads*, *2*(1), 48-54.

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. Computers & Education, 58(3), 978-988.

Campos, F. R. (2011). *Currículo, tecnologias e robótica na educação básica*(Doctoral dissertation, Tese (Doutorado em Educação), PUC-SP).

Eguchi, A. (2010). What is educational robotics? Theories behind it and practical implementation. In D. Gibson & B. Dodge (eds.), Proceedings of Society for Information Technology & Teacher Education International Conference 2010 (pp. 4006-4014). Chesapeake, VA: AACE.

Gomes, A., Henriques, J., & Mendes, A. (2008). Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. Educação, Formação & Tecnologias-ISSN 1646-933X, 1(1), 93-103.

Júnior, J., Rapkiewicz, C. E., Delgado, C., and Xexeo, J. A. M. (2005). Ensino de algoritmos e programação: uma experiência no nível médio. In XIII Workshop de Educação em Computação (WEI'2005). São Leopoldo, RS, Brasil.

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

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

Pontes, H. P. (2013). Desenvolvimento de jogos no processo de aprendizado em algoritmos e programação de computadores. Proceedings of the XII Simpósio Brasileiro de Games e Entretenimento Digital (SBGames). São Paulo.

[8] Plataforma de acesso (via web) a laboratórios de robótica remotos, baseado em tecnologias livres. Os usuários se cadastram e podem programar placas robóticas baseadas em Arduino as quais controlam diferentes sensores. (http://labvad.nce.ufrj.br/)

Ramos, J. L., & Espadeiro, R. G. (2015). Pensamento computacional na escola e práticas de avaliação das aprendizagens. Uma revisão sistemática da literatura. http://dspace.uevora.pt/rdpc/bitstream/10174/14227/1/challenges%202015b r.pdf

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all. Communications of the ACM, 52(11), 60-67.

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3):33–35.