

DuinoBlocks for Kids: um ambiente de programação em blocos para o ensino de conceitos básicos de programação a crianças do Ensino Fundamental I por meio da Robótica Educacional

Rubens Lacerda Queiroz¹, Fábio Ferrentini Sampaio^{1,2}

¹Programa de Pós-graduação em Informática (PPGI/UFRJ)

²Instituto Tércio Pacitti de Pesquisas e Aplicações Computacionais.– Universidade Federal do Rio de Janeiro (UFRJ)
Cidade Universitária - Ilha do Fundão - CEP 20001-970, Rio de Janeiro, RJ, Brasil

rubensqueiroz@outlook.com, ffs@nce.ufrj.br

***Abstract.** Today there are different relevant initiatives to promote code learning by young children. One of them is based on the association of two successful strategies: Educational Robotics and Visual Programming Environments. This paper presents “DuinoBlocks for Kids”, a Visual Programming Language based on free technology focusing on teaching some basic concepts of computer programming for elementary school children, allowing them to perform robotic experiments both in a remote laboratory - through a web application – and locally, using an Arduino board connected to the user’s computer.*

***Resumo.** Existem hoje algumas iniciativas bastante relevantes no sentido de viabilizar o ensino de programação de computadores já nos primeiros anos do ensino fundamental, sendo uma delas baseada na associação de duas estratégias muito bem sucedidas: a Robótica Educacional e a utilização de Ambientes de Programação Visual baseados em blocos de encaixar. Este trabalho apresenta o DuinoBlocks for Kids, um Ambiente de Programação em Blocos baseado em tecnologia livre e voltado ao ensino de conceitos básicos de programação a crianças do Ensino Fundamental I, permitindo a manipulação de experimentos em Robótica tanto em um laboratório remoto - via aplicação web - quanto localmente.*

1. Introdução

Na sociedade informatizada em que vivemos as crianças já nascem imersas no mundo digital, sendo conhecidas, devido à sua aparente fluência com as novas tecnologias, como “nativos digitais” [Prensky 2001].

Entretanto, Mitchel Resnick (2014) argumenta que, para se adquirir fluência em relação às novas tecnologias, não é suficiente saber usar com naturalidade as Tecnologias de Informação e Comunicação (TIC), é necessário que se saiba “projetar, criar e se expressar através dessas tecnologias” e, para tanto, é imprescindível saber programar.

Um maior domínio das novas tecnologias que é, como mencionado anteriormente, potencializado por meio da aquisição da habilidade de saber programar

computadores, permite aos alunos utilizarem com mais propriedade e eficiência o computador como ferramenta de apoio ao aprendizado das disciplinas do currículo escolar.

Além disso, a aprendizagem de programação trabalha habilidades e competências ligadas ao raciocínio lógico e matemático, úteis a muitos outros aprendizados [Resnick et al. 2009; Buechley, Eisenberg & Elumeze 2007; Barbero & Demo 2011] e ao desenvolvimento do Pensamento Computacional - uma forma de pensamento característica dos cientistas da computação, mas, universalmente aplicável, que envolve um conjunto de atitudes e habilidades tais como o uso da recursividade, abstração e decomposição na solução de problemas tanto técnico-científicos quanto da vida cotidiana [Wing 2006].

Tais características tornam bastante nítida a relevância da inclusão do ensino de programação de computadores como parte integrante do currículo básico desde as primeiras séries do ensino fundamental. E, para que esse aprendizado possa ocorrer de forma eficiente, eficaz e prazerosa, faz-se necessário o desenvolvimento de ferramentas que deem o suporte adequado a este processo. O DuinoBlocks for Kids vem exatamente ao encontro dessa demanda.

2. Fundamentação Teórica

2.1. Robótica Educacional

A ideia do uso da Robótica em educação se baseia fortemente na Teoria Construcionista de Seymour Papert, que une a Teoria Construtivista de Jean Piaget - com quem trabalhou entre 1958 e 1963 na Universidade de Geneva, Suíça - ao uso do computador na educação [Pinto 2011].

Papert (1980) adicionou à teoria de Piaget (1974) a ideia de que a construção do conhecimento acontece de maneira mais efetiva quando o aprendiz se engaja, de forma consciente, na construção de objetos, não importando se físicos ou virtuais, desde que possam ser vistos e analisados e façam parte do universo de interesses da criança [Alimisis 2013], como é o caso dos robôs que, naturalmente, instigam a sua curiosidade [Perez et al. 2013]

A Robótica Educacional permite à criança manipular e controlar objetos concretos e, através destes objetos, observar a materialização dos comandos dados por ela ao computador, processo a partir do qual se estabelece a construção do seu conhecimento.

Por tratar-se de uma disciplina de caráter multidisciplinar, a Robótica permite que os alunos trabalhem uma grande diversidade de competências e habilidades à medida que engloba, em um único objeto de estudo, diversas áreas do conhecimento, como matemática, eletrônica, mecânica, inteligência artificial, artes e programação, sendo esta o foco mais específico deste trabalho.

Além disso, a aprendizagem por meio da Robótica estimula a colaboração, a habilidade de trabalhar em grupo, a elaboração de estratégias para a solução de problemas e o desenvolvimento do pensamento formal [Souza & Duarte 2015, López-Rodríguez & Cuesta 2015; Perez et al. 2013; Rus 2006; Denis & Hubert 2001].

2.2. Linguagens de Programação Visual para Crianças

Junto com os primeiros computadores pessoais (PCs), no fim dos anos 70, surgiu o interesse de se utilizar esses equipamentos nas escolas para o aprendizado de programação.

No entanto, a dificuldade de compreensão, por parte das crianças, da sintaxe das linguagens de programação existentes na época, bem como a não conexão dos programas desenvolvidos com os interesses delas [Resnick et al. 2009] acabaram contribuindo para o insucesso de muitas dessas iniciativas.

Desde então, novas iniciativas surgiram no sentido de se tentar vencer estas dificuldades. No que diz respeito ao entendimento da sintaxe das linguagens de programação textuais, uma alternativa encontrada foi o uso de Linguagens de Programação Visuais (Visual Programming Language - VLP), ou seja, linguagens nas quais “a sintaxe (semanticamente significativa) inclui expressões visuais” [Burnett 1999, p.1].

Com base nesse paradigma, o MIT Media Lab (<https://llk.media.mit.edu/>), iniciou, em 2003, o desenvolvimento do Scratch (<https://scratch.mit.edu/>), uma Linguagem de Programação Visual baseada em “blocos de encaixar” que tinha por objetivo permitir que qualquer pessoa, de qualquer idade, pudesse programar. O sucesso do projeto foi tal que, em 2009, apenas dois anos depois do lançamento do site, usuários de todas as partes do mundo, na sua maioria crianças e jovens de 8 a 16 anos, já faziam upload de mais de 1500 projetos por dia.

Além do Scratch, outros projetos de grande vulto, voltados ao ensino de programação para crianças, também adotam o conceito da Programação Visual por meio de blocos de encaixar. Dentre eles podemos citar o Code.org (<https://code.org/>), que possui suas próprias ferramentas de ensino de programação por blocos e o Programaê (<http://programae.org.br/>) e Code Club Brasil (<http://codeclubbrasil.org/>), que utilizam o Scratch em seus programas de aprendizado de programação.

A adoção, por esses projetos, de Ambientes de Programação Visual baseada em blocos, cujas combinações são bastante intuitivas (uma vez que as crianças comumente possuem brinquedos que trazem esta ideia) [Jr. & Guedes 2015], sugerem ser esta uma abordagem bastante acertada no que diz respeito ao ensino de programação para crianças, o que veio a motivar o uso deste mesmo paradigma no desenvolvimento do DuinoBlocks for Kids.

2.3. Software e Hardware Livres

O Movimento Software Livre é "baseado no princípio do compartilhamento do conhecimento e na solidariedade praticada pela inteligência coletiva" [da Silveira 2003, p.438].

Existem atualmente disponíveis em repositórios de Software Livre como o GitHub (<https://github.com/>), um número incalculável de softwares, cujos códigos podem ser livremente baixados, utilizados, modificados e redistribuídos, permitindo a qualquer pessoa auxiliar, em qualquer tempo, na melhoria dos softwares lá disponibilizados, bem como desenvolver, individual ou coletivamente, novos programas baseados nos códigos existentes.

Outro movimento relacionado à democratização do acesso à tecnologia, é o Movimento Hardware Livre. Hardware Livre é um hardware eletrônico projetado e oferecido da mesma maneira que um Software Livre. Pode-se citar como exemplo de projetos desse tipo o Arduino (<http://www.arduino.cc/>), utilizado neste trabalho, e o Raspberry Pi (<https://www.raspberrypi.org>).

Um aspecto bastante importante em relação ao desenvolvimento de aplicações a partir do uso de Software Livre e da associação destes ao uso de hardware de baixo custo, em contraponto ao uso de soluções proprietárias, é a democratização do acesso à tecnologia.

2.4. Arduino

Arduino (<http://www.arduino.cc/>) é um projeto de Hardware Livre desenvolvido por um grupo de 5 pesquisadores de diferentes nacionalidades (Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis) no Interaction Design Institute Ivrea (<https://interactionivrea.org/>), que tinha por objetivo a criação de uma plataforma de prototipagem eletrônica de código aberto, simples de ser utilizada por qualquer estudante, inclusive crianças, de maneira que elas pudessem facilmente criar “coisas” que funcionassem sem a necessidade de frequentar um curso completo de eletrônica.

O seu baixo custo, qualidade, flexibilidade e facilidade de uso, têm feito das placas de prototipagem eletrônica Arduino uma excelente opção para o desenvolvimento de projetos na área de Robótica Educacional.

3. Trabalhos Relacionados

Não foi encontrado na literatura um Ambiente de Programação Visual voltado à programação de placas de prototipagem eletrônica Arduino (tecnologia adotada neste trabalho), pensado e desenvolvido especificamente para o ensino de programação a crianças do Ensino Fundamental I. Ou seja, um Ambiente de Programação Visual composto por blocos graficamente mais atrativos e dotados de uma semântica menos abstrata que aquelas empregadas nos ambientes de programação que trabalham com o mesmo hardware.

Como exemplos de Ambientes de Programação Visual para placas Arduino atualmente existentes, podemos citar o DuinoBlocks [Alves 2013], que serviu de inspiração para o desenvolvimento do DB4K; O S4A - Scratch for Arduino (<http://s4a.cat/>), um fork do Scratch que permite a programação de placas Arduino; e o Ardublockly (<https://github.com/carlosperate/ardublockly>), ambiente sobre o qual foi desenvolvido o DuinoBlocks for Kids.

Algumas características do DuinoBlocks for Kids que o diferenciam dos ambientes acima listados e que buscam tornar o seu uso mais acessível e aprazível a crianças do Ensino Fundamental I são:

- Blocos de programação desenhados de modo a apresentarem uma semântica diretamente relacionada com os dispositivos sendo manipulados e com os efeitos por eles causados sobre estes dispositivos.
- Supressão de detalhes relacionados ao hardware, como pinagens e valores de níveis de tensão.

- Uso de linguagem icônica.
- Conjunto "enxuto" de blocos.
- Simplificação dos valores dos parâmetros utilizados nos blocos, como por exemplo: Temperatura: alta/baixa; Luz: muita/pouca; Velocidade: rápida/média/devagar.

4. DuinoBlocks for Kids (DB4K)

O Grupo de Informática Aplicada à Educação (GINAPE) da UFRJ vem, desde 2008, desenvolvendo o projeto LabVad - Laboratório Virtual de Atividades Didáticas em Ciências e Robótica (<http://labvad.nce.ufrj.br/>), cujo objetivo principal é o de, a partir de pesquisas acerca da utilização das TIC na educação, fomentar práticas pedagógicas que tirem proveito da ideia de "aprender fazendo", desenvolvendo para tanto atividades experimentais [Pinto 2011].

A estratégia utilizada pelo LabVad para auxiliar na viabilização da prática de atividades experimentais, tão importantes na Educação Básica, sobretudo nas disciplinas de ciências naturais (Física, Química e Biologia), foi a utilização de Laboratórios Remotos, ou seja, experimentos reais sendo executados a distância por meio de computadores ligados à internet [Pinto 2011].

A utilização dos laboratórios remotos é uma solução que possibilita a execução de experimentos reais sem que para isso seja necessária a aquisição de equipamentos especiais ou alocação de espaço para a sua realização.

Dentre outros experimentos, o ambiente LabVad possui um laboratório de Robótica Educacional denominado LabVad Robótica [Souza et al. 2014], que busca fornecer ao usuário uma experiência de programação para Robótica muito próxima daquela que ele teria se estivesse realizando um experimento localmente.

A figura 1 apresenta a imagem vista pelo usuário, por meio de um vídeo em tempo real apresentado na interface do LabVad Robótica, durante a realização dos experimentos.



Figura 1. Imagem vista pelo usuário por meio do vídeo apresentado pelo LabVad Robótica

O labVad Robótica foi desenvolvido inicialmente visando os públicos jovem e adulto, de maneira que não dispunha de ferramentas que viabilizassem o uso do mesmo por crianças. O desenvolvimento do DuinoBlok for Kids surge exatamente do desejo de expandir o alcance do referido projeto a crianças do Ensino Fundamental I, tendo o mesmo sido inspirado no Ambiente DuinoBlocks [Alves 2013], implementado pela mesma equipe.

4.1. O Ambiente

O DB4K foi projetado de modo a permitir a sua integração com o LabVad, tanto no que se refere às tecnologias utilizadas quanto aos experimentos a serem manipulados por meio do ambiente.

No entanto, além da utilização do DB4K para controle, via internet, dos experimentos disponibilizados pelo LabVad Robótica, buscou-se possibilitar a utilização do ambiente também em uma versão “client-side”, ou seja, que fosse capaz de ser executada no computador do usuário, sem a necessidade de conexão com a internet, e capaz de controlar, diretamente em uma placa Arduino conectada ao computador, outros experimentos além daqueles disponibilizados pelo LabVad.

A interface do DB4K presente no ambiente LabVad permite ao usuário, via programação, a manipulação dos seguintes dispositivos: Um conjunto de LEDs, um LED RGB, um Display de 7 Segmentos, um Display LCD um Servo Motor e um Motor DC.

Já o Ambiente DB4K “client-side”, permite também a utilização de um conjunto de sensores e a manipulação de um carrinho robô e de um braço robótico (Figura 2), o que vem a possibilitar o desenvolvimento de uma gama bastante variada de atividades relacionadas ao ensino de conceitos básicos de programação.

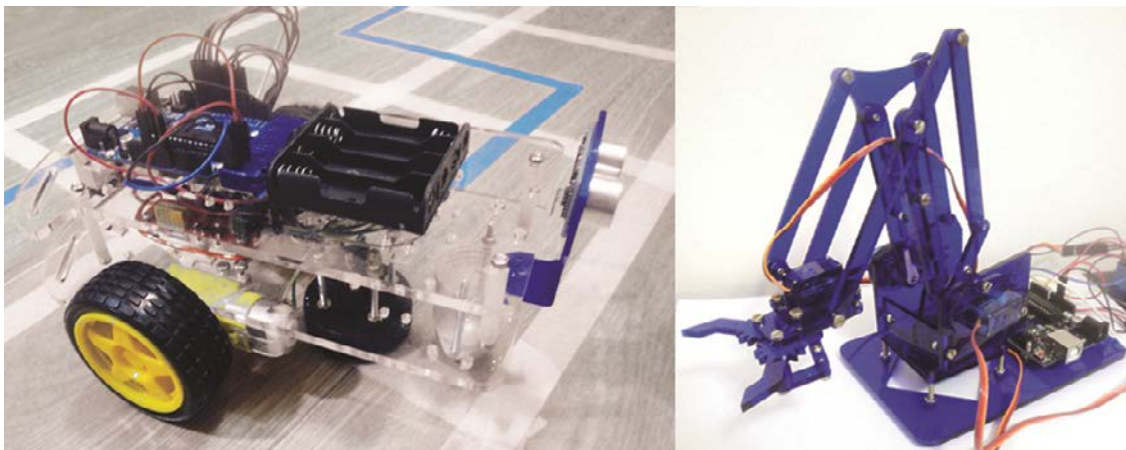


Figura 2. Carrinho e Braço Robóticos programáveis via DB4K client-side

Os blocos de programação presentes no ambiente foram desenhados de modo a apresentarem uma semântica diretamente relacionada com os dispositivos sendo manipulados e com os efeitos por eles causados sobre estes dispositivos.

Por exemplo, para acender um LED, desenhou-se o bloco “acender LED”; para escrever no display de LCD, desenhou-se o bloco “escrever ‘texto’ na linha ‘1’ do LCD”. Alguns desses blocos podem ser observados na Figura 3.

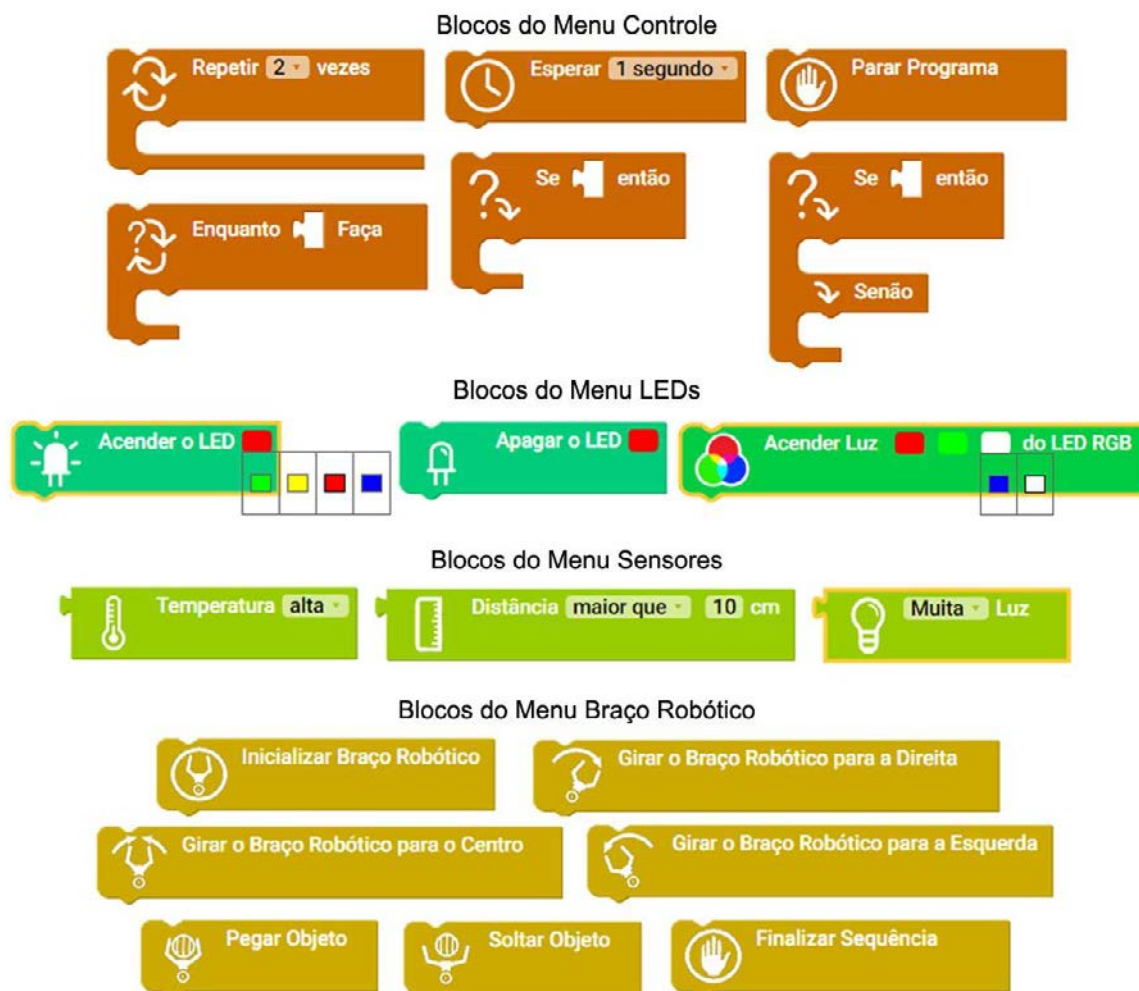


Figura 3. Amostra de blocos presentes no ambiente DB4K

Além dos blocos diretamente relacionados com os dispositivos a serem manipulados, foram desenhados também blocos para as estruturas de controle utilizadas em programação, como *repetição* e *decisão*.

A exemplo do que ocorre com o ScratchJr (<http://www.scratchjr.org/>) e com o Lego Mindstorms (<http://mindstorms.lego.com/>), optou-se por utilizar no DB4K, tanto uma representação textual quanto icônica para descrever as "funções" dos blocos. A ideia é a de facilitar, por parte dos usuários, o entendimento do fluxo de informação [Jr. & Guedes 2015].

Em relação às cores utilizadas no ambiente, decidiu-se por adotar cores vivas para os blocos e seus respectivos menus, direcionando a atenção do usuário exatamente para esses que são os elementos fundamentais do DB4K. Para o plano de fundo e barras de menus utilizou-se cores suaves de modo a não tornar visualmente cansativa a permanência possivelmente prolongada no ambiente.

Além da área de programação em blocos, para que o usuário possa, caso deseje, ter acesso à linguagem textual Wiring¹ associada a cada bloco do DuinoBlocks for Kids, decidiu-se por disponibilizar, na interface do DB4K, uma área para a exibição do programa textual correspondente ao programa criado por meio de seus blocos.

O DuinoBlocks for Kids (Figura 4) foi desenvolvido a partir do Ardublockly (<https://github.com/carlosperate/ardublockly>), um ambiente de programação em blocos para placas de prototipagem eletrônica Arduino direcionada ao público jovem e adulto e desenvolvido a partir da Biblioteca Blockly (<https://developers.google.com/blockly/>), uma biblioteca de código aberto criada pela Google Developers (<https://developers.google.com/>) e voltada para a construção de Ambientes de Programação Visual.



Figura 4. Visão Geral da Interface do DB4k client-side

O desenho da interface do DB4K aproveitou muitos dos elementos da interface do Ardublockly, já os blocos utilizados na programação foram totalmente redesenhados e programados. Alguns desses blocos estão presentes na Figura 3.

Desde o início do seu desenvolvimento, o ambiente vem sendo testado e avaliado pela equipe de especialistas do GINAPE/UFRJ, estando sua versão online, disponível para o público desde outubro de 2015, sendo atualmente utilizada em um curso de Robótica Educacional no Colégio Estadual José Leite Lopes, no município do Rio de Janeiro.

¹ Linguagem nativa para a programação das placas de prototipagem eletrônica Arduino, derivada da linguagem C/C++.

5. Considerações Finais

Os resultados aqui apresentados nos permitem concluir ser perfeitamente viável - lançando-se mão de bibliotecas e aplicações de código aberto disponíveis na Web, e da associação dessas a hardware de baixo custo - desenvolver ferramentas que facilitem aos docentes e discentes o uso das TICs para o aprendizado de programação por meio da Robótica.

Embora o DuinoBlocks for Kids tenha sido projetado para o ensino de conceitos básicos de programação a crianças do Ensino Fundamental I, acredita-se que o ambiente possa ser utilizado também para auxiliar na introdução ao aprendizado de programação por meio da Robótica Educacional a qualquer pessoa com pouco ou nenhum conhecimento nessas áreas.

Ressalta-se o fato de que, por se tratar de um ambiente construído com base em tecnologia livre, o DB4K pode servir como base para o desenvolvimento de outros ambientes adequados a determinadas necessidades específicas, assim como este utilizou o Ardublockly para o seu desenvolvimento, que por sua vez foi influenciado pelo ambiente BlocklyDuino (<https://github.com/BlocklyDuino/BlocklyDuino/>), tendo em todos eles sido utilizada a biblioteca Blockly.

5.1. Trabalhos Futuros

Para dar suporte ao uso adequado do DuinoBlocks for Kids e para que sejam feitas possíveis melhorias no ambiente de modo a torná-lo tão eficaz e eficiente quanto possível, é necessária ainda a realização de um conjunto de ações, entre elas:

- Criação de um site onde estarão presentes: uma descrição completa do ambiente, um "help" para o mesmo, tutoriais para a montagem dos experimentos controlados pelo DB4K, sugestões de atividades, a versão "client-side" do ambiente para download e um link para o LabVad.
- A realização de experimentos que busquem apontar os reais benefícios da utilização do DuinoBlocks for Kids no ensino de programação por meio da Robótica Educacional nas séries iniciais do Ensino Fundamental, bem como o aprimoramento do ambiente a partir dos resultados obtidos através desses experimentos.

Agradecimentos

Este trabalho foi desenvolvido com o financiamento parcial da RNP (Rede Nacional de Ensino e Pesquisa) através do Edital para os Grupos de Trabalho (GTs) da referida organização para o período 2014-2015.

Referências

- Alimisis, Dimitris. (2013) "Educational roboTIC: Open questions and new challenges". Themes in Science and Technology Education, v. 6, n. 1, p. pp-63.
- Alves, Raphael M. (2013) "Duinoblocks: Desenho e Implementação de um Ambiente de Programação Visual para Robótica Educacional". Dissertação (Mestrado em Informática) - Programa de Pós Graduação em Informática, Universidade Federal do Rio de Janeiro, Rio de Janeiro.

- Barbero, A.; Demo, G. B. (2011) “The art of programming in a Technical Institute after the Italian secondary school reform.” Proceedings ISSEP 2011.
- Buechley, L.; Eisenberg M.; Elumeze, N. (2007) “Towards a curriculum for electronic textiles in the high school classroom”. In: ACM SIGCSE Bulletin-2007, p. 28-32.
- Burnett, Margaret M. (1999) “Visual programming”. Wiley Encyclopedia of Electrical and Electronics Engineering.
- Da Silveira, Sérgio A. (2003) “Inclusão digital, software livre e globalização contra-hegemônica. Software Livre e Inclusão Digital”. Sao Paulo: Conrad, v. 7, p. 11.
- Denis, Brigitte; Hubert, Sylviane. (2001) “Collaborative learning in an educational roboTIC environment”. Computers in Human Behavior, v. 17, n. 5, p. 465–480.
- Jr., Almir de O.C.; Guedes, Ello´a B. (2015) " Uma Análise Comparativa de Kits para a Robótica Educacional". WEI - XXIII Workshop sobre Educação em Informática.
- López-Rodríguez, Francisco M.; Cuesta, Federico. (2015) “Andruino-A1: Low-Cost Educational Mobile Robot Based on Android and Arduino”. Journal of Intelligent & Robotic Systems, 3 maio.
- Papert, Seymour. (1980) “Mindstorms: Computers, Children and Powerful Ideas”. NY: Basic Books. 255p.
- Perez, Anderson L. F.et al. (2013) “Uso da Plataforma Arduino para o Ensino e o Aprendizado de Robótica”.[S.l: s.n.].
- Piaget, Jean. (1974)“To Understand is to Invent”. N.Y.: Basic Books.
- Pinto, Marcos de C. (2011) “Aplicação de Arquitetura Pedagógica em Curso de Robótica Educacional com Hardware Livre”. Dissertação (Mestrado em Informática) - Programa de Pós Graduação em Informática, Universidade Federal do Rio de Janeiro, Rio de Janeiro.
- Prensky, M. (2001) “Digital Natives, Digital Immigrants Part 1”, On the Horizon, Vol. 9 Iss: 5, p. 1–6.
- Resnick, Mitchel et al. (2009) “Scratch: programming for all”. Communications of the ACM, v. 52, n. 11, p. 60-67.
- Resnick, Mithel. (2014) “Palestra no evento Transformar 2014”. 46'30". Disponível em: <<https://www.youtube.com/watch?v=hRGJUc2opQ4>>. Acesso em: 4 mar.2016.
- Rus, Daniela. (2006) “Teaching roboTIC everywhere”. IEEE RoboTIC & Automation Magazine, v. 13, n. 1, p. 15–94.
- Souza, Marcos A.M; Duarte, José R.R. (2015) “Low-cost educational roboTIC applied to physics teaching in Brazil”. Physics Education, v. 50, n. 4, p. 482.
- Souza, Paulo Roberto A. et al. (2014) “LabVad: Laboratório Remoto para o Desenvolvimento de Atividades Didáticas com Robótica”. TISE : XX Congresso Internacional de Informática Educativa.
- Wing, Jeannette,M. (2006) "Computational thinking". Communications of the ACM, v. 49, n. 3, p. 33-35.