

block.ino: Um experimento remoto para ensino de lógica de programação, robótica e eletrônica básica

Lucas Mellos Carlos¹, João Paulo C. Lima¹, José Pedro S. Simão¹, Juarez B. Silva¹

¹Laboratório de Experimentação Remota – Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 150 – 88.906-072 – Araranguá – SC – Brasil

{mellos, lucas}@grad.ufsc.br

***Resumo.** Este artigo apresenta um protótipo criado para ensino de lógica de programação, robótica e eletrônica básica por dispositivos móveis que usa linguagem de programação visual construída por blocos para controle de um laboratório remoto de Arduino, com vista ao auxílio de disciplinas que se relacionem as áreas STEM. O protótipo chamado de block.ino consiste em uma interface adaptada para desenvolvimento de programas computacionais para uma placa Arduino controlando sensores e atuadores por meio de acesso remoto ao recurso. Ao final deste artigo, apresenta-se a relevância deste protótipo no cenário educacional considerando seu público alvo de alunos de ensino fundamental II e ensino médio.*

1. Cenário de Uso

Atualmente, figura-se um interesse em ensinar conceitos de ciência da computação para jovens. Esse interesse, não é apenas um treinamento para o futuro ou para à alta demanda que o mercado da área apromixima-se de requerer, mas sim, um avanço em benefícios que o ensino de lógica de programação tende a oferecer no processo de ensino-aprendizagem já no ensino fundamental. O conhecimento sobre programação, não é apenas uma habilidade para ciência da computação, mas sim, um mecanismo para resolver problemas que envolvam pensamentos algorítmicos [Sáez-López et al. 2016].

Para [McPherson and Anid 2014] o ensino das áreas STEM (*Science Technology, Engineering and Mathematics*) logo cedo na educação básica é a chave do sucesso para o mercado de trabalho no século 21. Durante décadas este conceito vem se fortalecendo com a competitividade e o desenvolvimento econômico. O governador do estado do Arizona nos EUA já destacava no ano de 1983 que este tipo de conhecimento estava se tornando o diferencial no desenvolvimento para outras nações que haviam colocado ênfase em disciplinas de educação, engenharia e matemática frente o desenvolvimento econômico [Contreras and Siu 2015].

Conforme [Cross et al. 2015], uma das formas de atrair estudantes para as áreas de engenharias é fazendo com que estes tenham vivências pré-universitárias já na sua educação básica, ajudando-o a desenvolver a fluência tecnológica. O uso de *Programming Assistance Tools* (PATs) pode ser uma das alternativas incorporadas por professores no desenvolvimento estudantil, explorando conhecimentos como de raciocínio lógico, robótica, eletrônica básica, entre outros para a motivação de discentes. Atualmente, o uso de linguagens visuais de programação tem se tornado muito pertinente, desta forma discute-se o uso destas como sendo uma representação visual que facilita a aprendizagem de programação [Booth and Stumpf 2013, Price and Barnes 2015].

O ensino de tal prática nas escolas acarreta no desenvolvimento de um conjunto de habilidades e conhecimentos que constituem um ser. As habilidades constituídas são o favorecimento do pensamento para a resolução de problemas, bem como o desenvolvimento de atividades em modo “faça você mesmo” [Koorsse et al. 2015].

Atividades práticas de ciências são imprescindíveis tanto na educação superior como no ensino fundamental, por vez, segundo os dados do Censo da Educação [INEP 2014] apenas 8% das escolas públicas brasileiras possuem laboratório de ciências e ainda, somente 45% do total de escolas possuem laboratório de informática em suas instalações.

Comumente, o ensino de programação não é muito acessível para algumas instituições, principalmente aquelas que se encontram em condições vulneráveis de infraestrutura ou outros recursos, estas tendem a sofrer mais com ações que demandem mais esforços que o convencional para adesão de materiais educacionais.

Desta forma, o uso da experimentação remota pode potencializar o ensino tanto na educação básica como na educação superior, ofertando acesso ao ensino de conceitos de lógica de programação, robótica e eletrônica básica, através da Internet para disciplinas que se relacionam as áreas STEM, corroborando com as informações descritas por [Cross et al. 2015].

2. Desenvolvimento

O protótipo desenvolvido no Laboratório de Experimentação Remota (RExLab) da Universidade Federal de Santa Catarina tem o intuito de auxiliar professores e alunos de educação básica no processo de ensino-aprendizagem em disciplinas relacionadas as áreas STEM através da programação remota de uma placa Arduino por meio de linguagem visual construída por blocos.

O desenvolvimento deste protótipo segue a arquitetura cliente-servidor e seu processo de desenvolvimento encontra-se dividido em quatro etapas, sendo: (1) laboratório remoto Arduino, (2) aplicação cliente, que incorpora o desenvolvimento da interface de programação e material auxiliar ao usuário, (3) desenvolvimento da *lab server application*, responsável pela comunicação entre a aplicação e o laboratório remoto e (4) criação de sequências didáticas para apoio docente.

2.1. Laboratório Arduino

O laboratório Arduino consiste em um Kit convencional de Arduino, sendo, uma placa Arduino Uno - ou qualquer modelo pertinente - conectada a uma *protoboard* equipada com sensores e atuadores que situam-se postos sobre um case de acrílico. A Figura 1 demonstra o esquema dos pinos dos componentes utilizados neste projeto.

Cada componente é montado e posteriormente mapeado nos arquivos de configuração das aplicações. Os componentes utilizados nesta instância de laboratório remoto de Arduino são: 1 Placa Arduino UNO, 1 visor LCD (16x2), 1 potenciômetro, 1 Servo Motor, 1 sensor de temperatura e umidade DHT11, 1 sensor Hall e um conjunto de 4 LEDs. A Tabela 1 apresenta o custo envolvido na construção de uma instância desse laboratório remoto de Arduino.

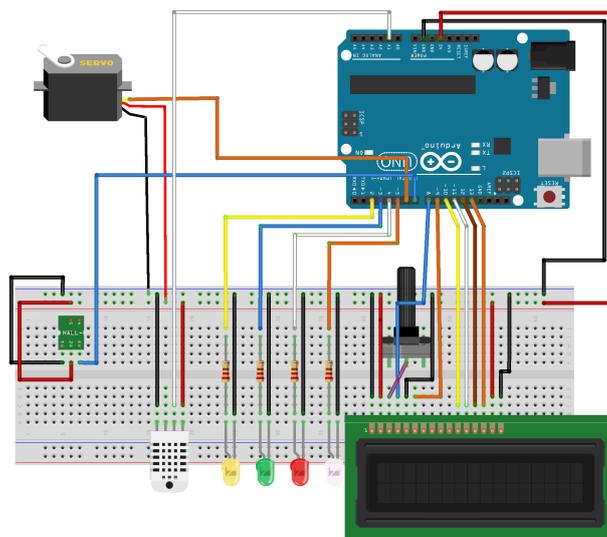


Figura 1. Diagrama expondo o circuito elétrico

Tabela 1. Custo de uma instância de laboratório Arduino

Item	Custo (R\$)
Raspberry Pi 2 model B	290.00
Arduino Uno	50.00
Web cam	99.00
Case em Acrílico	65.00
Fonte de energia	30.00
Componentes eletrônicos	92.50
Total	625.50

2.2. Aplicação Cliente

A aplicação cliente é desenvolvida usando tecnologias Web e design responsivo para permitir o ajuste automático a diversos tamanhos de telas dos dispositivos. A escolha pelo uso de tecnologias web se deu pela disponibilidade da biblioteca Google Blockly em formato web, sendo que, esta é a responsável pela geração dos blocos, logo este tipo de desenvolvimento permite que a aplicação seja utilizada tanto em *browsers* como em aplicativos móveis através do desenvolvimento de aplicações híbridas.

A aplicação utiliza o *framework Front-End* Materialize CSS para desenvolvimento de sistemas web utilizando o padrão Google Material Design. O aplicativo para a plataforma Android utiliza o mesmo *Front-End* da aplicação para *browsers*, porém utiliza o *framework* Phonegap para fazer a *bridge* entre as páginas web e os recursos do dispositivo móvel [Phonegap 2016].

O aplicativo desenvolvido baseado no modelo *Embed WebView* tem acesso a funções do dispositivo através de plugins, a Tabela 2 detalha os plugins contidos na aplicação e suas respectivas funções.

Tabela 2. Plugins contidos no aplicativo móvel

Identificação	Função
<i>cordova-plugin-whitelist</i>	Controle de navegações em URLs
<i>cordova-plugin-network-information</i>	Estado da conexão com a Internet
<i>cordova-plugin-x-toast</i>	Mensagens em formato de Toast
<i>cordova-plugin-file</i>	Escrita e leitura de arquivos

2.3. Lab Server Application

o laboratório remoto é um dispositivo independente conectado a Internet, e opera baseado em uma *Application Programming Interface* (API). A aplicação *stand-alone* roda em um Raspberry Pi model 2B. O Raspberry Pi é uma placa de baixo custo que porta a aplicação rodando Node.js, essa proporciona o funcionamento da API mencionada, responsável pelo controle de usuários através de uma fila, controle da placa Arduino e gerenciamento das ferramentas de desenvolvimento.

O WebSocket é o protocolo utilizado para o transporte dos dados trocados entre o cliente e o servidor, os dados são estruturados em formato JSON (*Java Script Object Notation*). O WebSocket é um protocolo assíncrono que juntamente com o paradigma orientado a eventos provido pelo Node.js é muito recomendado para uso em aplicações de laboratórios remotos [Salzmann et al. 2015]. Esta ligação é importante para trocas de mensagens em tempo real como eventos de detecção de campos magnéticos, ou envio da temperatura por sensores, por exemplo.

O processo de compilação e *upload* depende de ferramentas providas pela Atmel para microcontroladores AVR e bibliotecas Arduino que também são usadas na IDE original da plataforma. A placa Arduino é conectada a porta USB do *host* servidor da aplicação que também permite o envio e recepção de mensagens via porta serial. Apesar de haver botão para *reset* manual, o microcontrolador pode ser reiniciado via software pelo usuário na aplicação cliente.

O *lab server application* também é responsável por fazer a gerência da fila de requisições de usuários, autorizando a operação em apenas uma seção. Desta forma, cada usuário tem um tempo limitado para enviar e testar seu código no laboratório de Arduino, este tempo pode ser estendido caso não haja outros usuários requisitando acesso no momento.

2.4. Teste de usabilidade

O teste utilizou a metodologia *System Usability Survey* (SUS) para avaliação. Originalmente, criada por [Brooke et al. 1996] o SUS é uma metodologia de avaliação amplamente utilizada para qualificar serviços, incluindo *hardware*, *websites* e aplicações. O presente teste consiste em um questionário composto por 12 questões em uma escala de avaliação de 5 pontos que varia de discordo totalmente até concordo totalmente. Em complemento ao questionário, foram incluídas 8 questões de múltipla escolha sobre os atributos de usabilidade, aspectos técnicos e sugestões.

18 usuários foram convidados para participar deste teste, sendo 5 professores de ensino básico e técnico de idades entre 25 e 38 anos, e 13 alunos frequentadores do ensino médio com idades entre 14 e 16 anos. Em relação a conhecimento sobre linguagens de

Tabela 3. Questões e resultados do questionário SUS

Questão	Média
Você usaria essa aplicação frequentemente	4.11
É muito difícil usar este app	1.94
É muito fácil usar este app	4.11
Eu preciso de suporte técnico para usar este app	2.72
Todas as funções estão bem integrada	4.22
O app é inconcistente	2.05
Eu acho que maioria das pessoas aprenderiam a usar esse app facilmente	4.11
Eu me senti confiante usando este app	4.11
Eu preciso aprender muito para continuar usando este app	2.11
Foi fácil encontrar o que precisava nesse app	4.05
A interface do app é boa	4.22
As informações são bem organizadas na tela	3.83

programação para robôs ou outros microcontroladores, cerca de 38.9% dos entrevistados afirmaram ter tido contato prévio esse tipo de aplicação. 61% destes já haviam tido contato com alguma linguagem de programação visual como Scratch ou App Inventor, e cerca de 11% nunca havia tido contato algum com nenhum tipo de linguagem de programação. A respeito do tamanho da tela dos dispositivos, cerca de 50% dos usuários utilizaram o aplicativo em tablets médios (tamanho entre 7 e 9 pol.), 28% usaram smartphones de tamanho grande (de 4,5 a 6 pol.) e o restante usou smartphone de tamanho pequeno, entre 4 e 5 pol.

A Tabela 3 resume o que foi respondido pelos usuários deste teste em cada questão usando média aritmética, em suma, as respostas indicam a satisfação dos usuários em relação a disposição de informações e necessidade de suporte técnico para usar o aplicativo. De tal modo, as características de interface gráfica, expectativas de usuário, aprendizado, memorização e compreensibilidade apresentam-se acima da média.

2.5. Sequência didática de aprendizagem

Com vistas à oferta de apoio aos docentes, elaborou-se uma sequência didática de aprendizagem utilizando a metodologia de aprendizagem baseada em problema disponibilizando-a em Ambiente Virtual de Aprendizagem (AVA), neste caso, o Moodle.

Esta propõe que o professor incorpore-a em seu curso em um AVA e o aluno realize uma determinada sequência de passos sobre conteúdos relacionados com a plataforma Arduino, podendo testar seus conhecimentos em questionários e discutindo problemas com demais colegas através de fóruns e chats.

3. Apresentação do software

O aplicativo é disponibilizado em três idiomas, inglês, português e espanhol e contém funções de acessibilidade como um acesso guiado passo a passo dentro da aplicação que apresenta as funções principais que o usuário poderá explorar durante o seu uso. Buscando uma melhor experiência para o usuário, elaborou-se uma página de documentação

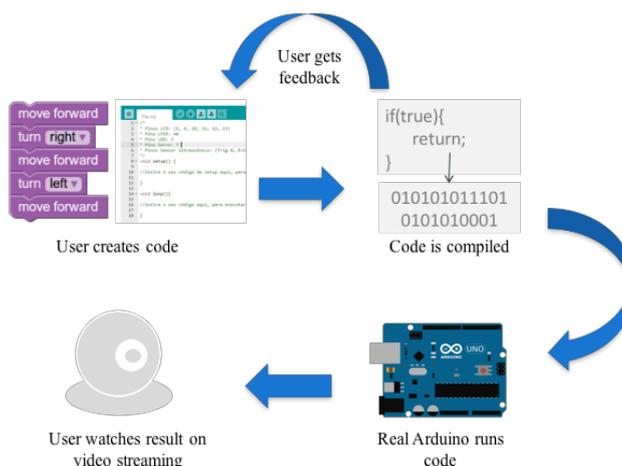


Figura 2. Diagrama de sequência de uso do aplicativo block.ino

que demonstra como realizar a utilização dos principais blocos contidos na aplicação, bem como exemplos de utilização de algumas funcionalidades que podem ser exploradas no laboratório de Arduino como o controle de sensores e atuadores.

O uso da aplicação se dá por duas formas distintas, sendo via aplicativo móvel para Android ou via Web pelo browser. A construção do código acontece com o *drag and drop* dos blocos localizados no menu da aplicação, possibilitando assim a construção de um programa, após o usuário realiza o processo de compilação e *upload* do código que é convertido da linguagem de blocos para C/C++ para ser enviado a placa Arduino através do protocolo WebSocket, neste passo o usuário pode acompanhar a execução do seu código através de um *streaming* de vídeo MJPEG como demonstra a Figura 2 que exhibe a arquitetura de funcionamento do protótipo.

Conforme exhibe a Figura 3, os menus à esquerda separam os blocos por categorias, uma vez que estes são arrastados para a área central da interface onde encontra-se o editor de código. Por vez, o código construído usando linguagem visual também é apresentado no formato de linguagem para Arduino na guia Código Fonte, o usuário ainda pode verificar o posicionamento de cada pino preestabelecido no laboratório de Arduino na guia Diagrama do circuito que ainda possui o recurso de zoom quando o usuário clica e arrasta o cursor sobre a imagem.

Os botões localizados ao topo do editor de código, servem respectivamente para expandir a área de edição de código, realizar *upload* e verificação do código, já o botão localizado no menu superior esquerdo serve para dar acesso a um menu de funções como salvar arquivos, mudar o idioma, explorar exemplos, além dos botões de Guia e Docs que auxiliam no descobrimento de funções e auxiliam no desenvolvimento.

4. Considerações finais

O desenvolvimento do raciocínio lógico a partir do ensino de programação e robótica durante o período do ensino básico desperta a atenção de pesquisadores e educadores. A fim de atingir esse objetivo, muitas ferramentas foram desenvolvidas com o propósito de facilitar e tornar flexíveis ambientes de desenvolvimento, sem tirar o foco da aprendizagem dos conceitos principais.

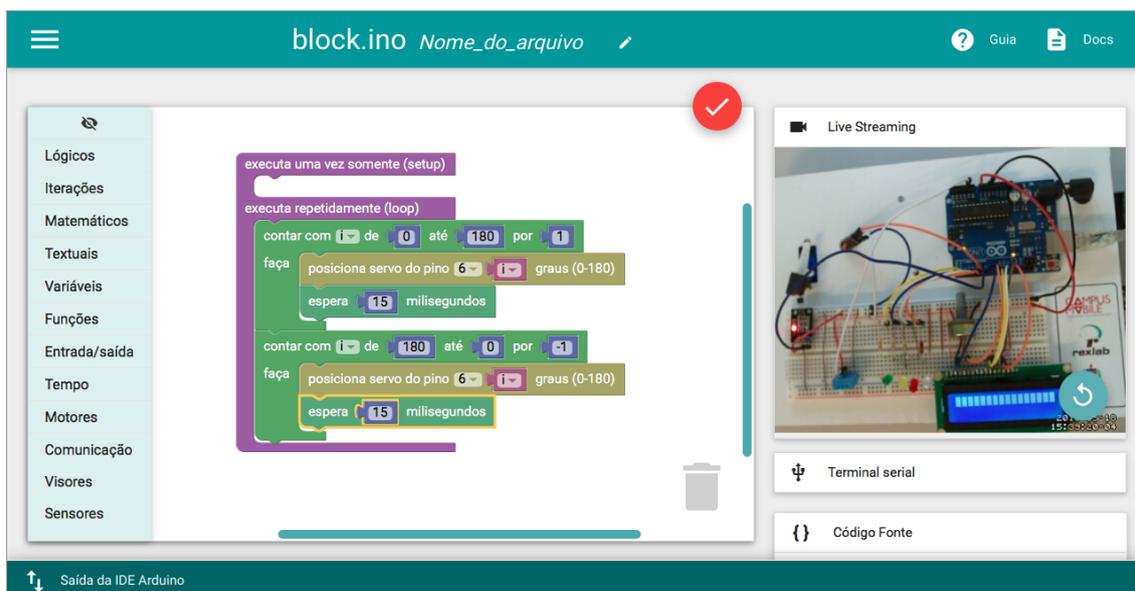


Figura 3. Interface do aplicativo block.ino

A combinação de Arduino, ensino de lógica e laboratório remoto tenta suprimir as dificuldades e complexidades encontradas no ambiente escolar atual, focando no desenvolvimento do raciocínio lógico para resolução de problemas reais, a partir do compartilhamento de recursos pela Internet. Portanto, aliar o recurso digital com um conjunto físico, mesmo que mediado a distância, permite uma aproximação do descobrimento prático.

Deste modo, o usuário programa e interage em “tempo real” com a placa Arduino e seus componentes, uma vez que a interação por meio da linguagem de blocos é considerada um grande estímulo aos usuários, em especial, usuários que não possuem experiência na área de programação, ou iniciantes em linguagens de programação e lógica. Desta forma, o ensino de lógica tem um grande apelo educacional no novo modelo de economia que esta se projetando no Século 21. Sendo assim, o aplicativo pode ser inserido por professores nos seus planos de aulas em disciplinas como física e matemática para ensino de álgebra, trigonometria, entre outros.

O ensino de lógica de programação para ensino o fundamental e médio possui um forte apelo por ajudar na construção do raciocínio lógico e criatividade, além de haver relatos de que o mesmo facilita o aprendizado em disciplinas de português e matemática. Soluções similares a esta encontram-se disponíveis no mercado por um preço muito superior, como por exemplo, o Kit Lego Mindstorms, que além de apresentar um alto preço necessita de tempo hábil para montagem e ainda de um computador exclusivo para a programação, além de permitir a prática de apenas um aluno por vez.

Uma vez implementado o ambiente, este projeto tem a possibilidade de ir diretamente para uso em cursos técnico em informática e eletrônica, e em escolas públicas, onde iniciativas de programação são muito raras. Para o uso do ambiente podem ser utilizados laboratórios de informática já disponíveis nas escolas ou o próprio dispositivo que os estudantes levam para a sala de aula, já que o requisito mínimo é suportar um navegador web. Além disso, o usuário pode trabalhar por conta própria em qualquer lugar e a

qualquer momento, a partir de uma conexão com a internet.

Referências

- Booth, T. and Stumpf, S. (2013). End-user experiences of visual and textual programming environments for arduino. In *End-User Development*, pages 25–39. Springer.
- Brooke, J. et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.
- Contreras, G. J. and Siu, K. W. M. (2015). Computer programming for all: A case-study in product design education. *Procedia - Social and Behavioral Sciences*, 182:388 – 394. 4th {WORLD} {CONFERENCE} on {EDUCATIONAL} {TECHNOLOGY} {RESEARCHES} (WCETR-2014).
- Cross, J. L., Hamner, E., Bartley, C., and Nourbakhsh, I. (2015). Arts amp; bots: Application and outcomes of a secondary school robotics program. In *Frontiers in Education Conference (FIE), 2015. 32614 2015. IEEE*, pages 1–9.
- INEP (2014). Matrículas e infraestrutura - qedu. <http://www.qedu.org.br/brasil/censo-escolar?year=2014>. Acessado em: 03-05-2016.
- Koorsse, M., Cilliers, C., and Calitz, A. (2015). Programming assistance tools to support the learning of {IT} programming in south african secondary schools. volume 82, pages 162 – 178.
- McPherson, S. and Anid, N. M. (2014). Preparing stem teachers for k-12 classrooms: Graduate certificate evaluation and innovation. In *Integrated STEM Education Conference (ISEC), 2014 IEEE*, pages 1–6.
- Phonogap (2016). About. <http://phonogap.com/about/>. Acessado em: 17-05-2016.
- Price, T. W. and Barnes, T. (2015). Comparing textual and block interfaces in a novice programming environment. In *Proceedings of the eleventh annual International Conference on International Computing Education Research*, pages 91–99. ACM.
- Sáez-López, J.-M., Román-González, M., and Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “scratch” in five schools. *Computers & Education*, 97:129–141.
- Salzmann, C., Govaerts, S., Halimi, W., and Gillet, D. (2015). The smart device specification for remote labs. In *Remote Engineering and Virtual Instrumentation (REV), 2015 12th International Conference on*, pages 199–208. IEEE.