

## Uma solução livre e de baixo custo para prática e aprendizagem de programação e robótica

Leonardo Costella, Gabriel Paludo Licks, Adriano Canabarro Teixeira

Instituto de Ciências Exatas e Geociências – Universidade de Passo Fundo (UPF)  
Passo Fundo – RS – Brasil

{134383, 138119, teixeira}@upf.br

***Abstract.** This article describes the process of designing, developing and testing a low-cost minicomputer in order to support learning processes of computer programming and robotics. The prototype, called EduPi, is based on the Raspberry Pi platform, Raspbian operating system and 11 programming and robotics learning environments, along with study material. The tests were performed with two young men of 12 and 13 respectively on the dependencies of the Group of Study and Research on Digital Inclusion. With positive feedback, EduPi is reported as a higher performance computers as their schools' and preferred when the focus activity is programming and robotics.*

***Resumo.** Este artigo relata o processo de concepção, desenvolvimento e teste de um minicomputador de baixo custo com o objetivo de suportar processos de aprendizagem de programação de computadores e de robótica. O protótipo, denominado EduPi, é baseado na plataforma Raspberry Pi, com sistema operacional Raspbian e 11 ambientes de aprendizagem de programação e robótica, com material de estudo. Os testes foram realizados com dois jovens de 12 e 13 anos respectivamente nas dependências do Grupo de Estudo e Pesquisa em Inclusão Digital. Com feedback positivo, relatam que o EduPi possui maior desempenho que os computadores de suas escolas, bem como o preferem quando o foco são atividades de programação e robótica.*

### 1. Introdução

Vivemos em um mundo permeado pela tecnologia, onde o contato com tais aparatos deixa de ser opcional para tornar-se, em muitos casos, vital. O ambiente em que estamos inseridos nos traz de forma inerente a necessidade de interagirmos com tecnologias digitais nas tarefas mais básicas do dia-a-dia. Por consequência, encontra-se uma grande quantidade de pesquisas científicas que mostram a maneira como a tecnologia está influenciando a sociedade atual qualitativamente até dados quantitativos que nos trazem os altos índices de sua presença no mundo contemporâneo.

Porém, já não basta hoje e, principalmente no futuro, sermos meros usuários dessas tecnologias. A demanda que a constante evolução dessas ferramentas vem nos trazendo, através de sua influência atual, exige que passemos do simples estado de usuários para o patamar de autores. Douglas Rushkoff (2012) nos traz a ideia de que

Quando nós, os humanos, adquirimos linguagem, aprendemos não somente a ouvir, mas a falar. Quando ganhamos a escrita, nós aprendemos não apenas a

ler, mas a escrever. E, na medida em que nos movemos em direção a uma realidade crescentemente digital, nós precisamos aprender não apenas a usar programas, mas a fazê-los também. No panorama emergente, altamente programado, ou você criará o *software* ou será o *software*. Simples assim: programe, ou será programado. Escolha a primeira opção e ganhe acesso ao painel de controle da civilização. Escolha a última, e poderá ser sua última escolha real. (RUSHKOFF, 2012, p. 7-8)

Ao analisarmos essas afirmações, no futuro teremos duas possibilidades: ou estaremos seguindo um caminho já programado por alguém ou estaremos no estado de autoria, onde poderemos ter total controle sobre nossas ações, construindo o mundo através de linhas de código. Nesse sentido, essas tecnologias vêm para caracterizar o modo como vivemos e trabalhamos, onde as pessoas que as programam acabam por moldar o nosso mundo (RUSHKOFF, 2012).

A programação de computadores é algo que dificilmente é visto por usuários leigos de tecnologia, visto que ela se apresenta somente no processo de construção do *software*, sendo algo abstrato e não visível depois da finalização do código. Já os aparatos robóticos, por sua vez, podem ser vistos em funcionamento e fornecem uma maior tangibilidade ao *software* que os anima. Porém, esses aparatos aparecem hoje em dia majoritariamente nas linhas de produção da indústria, na automação de processos em diversas e variadas organizações, bem como em projetos de centros pesquisa.

Sendo assim, dois elementos justificam o desenvolvimento desta pesquisa. O primeiro diz respeito à forma como a informática educativa está presente nas escolas brasileiras, geralmente desvinculada de qualquer direcionamento autoral e, mais distante ainda, de processos que se valham da programação e da robótica para tanto. A segundo está relacionado à necessidade de se potencializar e popularizar a programação de computadores e a robótica para todas as camadas da sociedade, dentro ou fora do ambiente formal de ensino.

Buscando criar uma solução livre e de baixo custo para o desenvolvimento e a potencialização dessas habilidades, o Grupo de Estudo e Pesquisa em Inclusão Digital da Universidade de Passo Fundo desenvolveu um protótipo baseado em *hardware* e *softwares* livres. O protótipo, denominado EduPi, é baseado na plataforma *Raspberry Pi* com mudanças no sistema operacional nativo. Conta com 11 ambientes de aprendizagem de programação e robótica, material de estudo. Posto isto, este artigo tem por objetivo relatar o processo de concepção, desenvolvimento e teste do EduPi, bem como apresentar os passos futuros no seu desenvolvimento.

## **2. Programação de computadores e protagonismo**

Na mesma lógica de Rushkoff (2012), o filósofo italiano Umberto Eco (s. d.), em entrevista, aponta que a sociedade do futuro será composta por três castas: a primeira, mais numerosa, na base da pirâmide, será formada pelas pessoas que percebem o mundo pelos meios de comunicação de massa. O segundo grupo, no meio da pirâmide, será composto pelas pessoas que utilizam computadores, ou seja, utilizam e-mail, redes sociais, etc. Por fim, a elite intelectual da sociedade será composta pelas pessoas que saibam programar computadores, tendo sua posição no topo da pirâmide.

A partir dessa tendência, surge a demanda de fornecer alternativas que auxiliem e viabilizem a transição de indivíduos que são usuários de tecnologia e se transformarem em protagonistas de uma sociedade organizada em torno de códigos de computador. Nesse contexto se torna necessário entrar em contato com as ferramentas tecnológicas que nos permitem atingir esse objetivo. O ponto de partida para isso é o aprendizado da programação de computadores, posto que a mesma é base mister para exercitar a autoria nesse processo de transição. A importância da programação no mundo de hoje vem se confirmando através do aparecimento de diversas iniciativas formais e informais para estimular o interesse pela mesma, sendo que várias escolas vêm incluindo essas atividades no currículo escolar - principalmente nos Estados Unidos da América.

Quando pensamos especificamente em programação, acreditamos que o seu uso tem grande destaque como ferramenta educacional, pois por intermédio da resolução de problemas via uma linguagem de programação, tem-se descrição do processo utilizado pelo aluno para resolver uma tarefa. (BARANAUSKAS et al., 1999, p. 63)

Tudo isso reflete diretamente no rendimento escolar, onde estudos recentes apontam que o ensino da programação de computadores aos alunos aumenta em 30% o rendimento escolar, principalmente nos conteúdos de Matemática e Interpretação de Textos (CODE.ORG, 2013). Assim percebe-se que a programação nos auxilia a desenvolver habilidades cognitivas que vão além dos saberes ligados a essas tecnologias, entendendo-se aos âmbitos multidisciplinares.

Além disso, a programação de computadores vem correlacionada com a robótica educacional, que segundo Papert et al. (1980), sua utilização permite que áreas que sempre pareceram muito abstratas passem a serem visualizadas de forma mais clara e tangível pelas crianças. A robótica, por si só, é um exemplo de ferramenta multidisciplinar. Zapata, Novales e Guzmán (2004) citam a robótica educacional como uma ferramenta que cria ambientes de aprendizagem interessantes e motivadores, onde coloca o professor como facilitador da aprendizagem e o aluno como construtor ativo.

Ao programar, o aluno é levado a pensar de maneira lógica e objetiva. Também faz com que problemas complexos, para serem resolvidos, precisem ser decompostos e uma série de problemas menores, mais fáceis de serem compreendidos e ultrapassados. Assim, mesmo que estes alunos no futuro não venham seguir carreira em áreas correlatas a da robótica, tais aptidões serão benéficas em suas atividades profissionais nesse mundo digital onde encontram-se inseridos. Já em atividades de robótica educacional, a fim de agilizar os resultados obtidos pelos alunos e não ter que lidar com conceitos complexos e ainda não aprendidos de eletrônica e mecânica, são utilizados kits de robótica educativa, ou até mesmo motores, sensores e/ou atuadores onde as questões relativas a mecânica e eletrônica já estão resolvidas, ficando a ênfase na programação do aparato.

Assim, a criação de soluções de baixo custo para a aprendizagem de programação e robótica podem contribuir significativamente não somente para a criação de contextos favoráveis à aprendizagem de programação e robótica, mas também, para

se construir conhecimento acerca do processo e seus desdobramentos sobre o desenvolvimento cognitivo dos indivíduos.

### 3. O projeto EduPi

Tendo em vista toda essa demanda e analisando possíveis soluções, foi desenvolvido um protótipo de computador pessoal com diversos ambientes de programação de computadores e robótica junto ao sistema. Com isso, objetiva-se verificar qual é o potencial que a utilização dessa ferramenta - baseada em *hardware* de baixo custo e soluções de *software* livre para a aprendizagem de programação e robótica - possui na criação espontânea dessas competências e no alcance do protagonismo.

A análise da apropriação desse dispositivo trata-se da ampliação de um recente estudo do Grupo de Estudo e Pesquisa em Inclusão Digital<sup>1</sup>, onde a partir da placa *Raspberry Pi* foi criado um ecossistema com ambientes de programação de computadores e robótica voltados para diferentes idades e níveis de conhecimento, todos com tutorias explicativos que vão desde o mais básico até situações mais complexas. A ideia desta etapa é avaliar em que medida o protótipo desenvolvido suporta uma apropriação não guiada por representantes do público a que se destina.

#### 3.1. Desenvolvimento do protótipo

Na construção do protótipo, teve-se como componente principal uma placa *Raspberry Pi*, compacta e de baixo custo, inclusive atendendo a todos os requisitos mínimos para instalação do sistema operacional e de todos os ambientes de programação e robótica. Essas características fizeram com que a placa se encaixasse muito bem no objetivo que se tinha: a construção de um dispositivo de fácil escalabilidade; de baixo custo; e de grande praticidade.

Em questão de *software*, todos os ambientes de programação e robótica disponíveis possuem um tutorial sobre o funcionamento da ferramenta e que servirá de guia para aprendizagem. Os mesmos estão disponibilizados no formato PDF, visando um fácil manuseio e um menor uso de memória e processamento comparado a utilização de vídeos. Para os testes dos ambientes, foram utilizados exemplos simples que possibilitaram a validação de todos os recursos dos mesmos. A seguir são detalhadas as tecnologias e soluções utilizadas no protótipo, bem como detalhes sobre o *hardware* empregado em caráter mais técnico.

#### 3.2. Soluções embarcadas

A versão inicial do EduPi<sup>2</sup> possuía como sistema operacional a versão do *Raspbian*<sup>3</sup> disponibilizada no dia 16 de fevereiro de 2015, baseado no *Debian Wheezy*. Esse sistema operacional é livre e otimizado para o *hardware* do protótipo. A fim de se configurar como uma plataforma completa para o aprendizado de programação e

---

<sup>1</sup> Página oficial: <http://gepid.upf.br/>.

<sup>2</sup> Página oficial: <http://gepid.upf.br/edupi/>.

<sup>3</sup> Livre e baseado em *Debian*, o qual é otimizado para o *hardware* do dispositivo. Mais informações em <https://www.raspbian.org/>.

robótica, nesse primeiro release foram disponibilizados ambientes para a prática de robótica educativa e para programação. Cada um deles com o objetivo de oferecer alternativas em diferentes complexidades, a fim de auxiliar e torna-se atrativo a diferentes perfis de usuários. As soluções embarcadas no aparato estão detalhadas na tabela abaixo.

**Tabela 1. Características dos ambientes**

<b>Scratch for Arduino (S4A)</b>	<a href="http://s4a.cat/">http://s4a.cat/</a>
S4A é uma modificação do <i>software Scratch</i> que permite uma programação simples para a plataforma de <i>hardware open source Arduino</i> . Ele fornece novos blocos para o gerenciamento de sensores e atuadores conectados ao <i>Arduino</i> .	
<b>Ardublock (S4A)</b>	<a href="http://blog.ardublock.com/">http://blog.ardublock.com/</a>
O <i>Ardublock</i> é uma linguagem gráfica, semelhante ao S4A, utilizada para programação da placa de prototipação <i>Arduino</i> . Instalado como um <i>plug-in</i> no ambiente <i>Arduino IDE</i> .	
<b>Arduino IDE</b>	<a href="https://www.arduino.cc">https://www.arduino.cc</a>
Ambiente de desenvolvimento aberto, que torna possível uma fácil programação para a Placa de prototipação <i>Arduino</i> . A programação é feita em uma linguagem baseada na linguagem C/C++.	
<b>NewProg</b>	<a href="http://www.newprog.com.br/">http://www.newprog.com.br/</a>
O NewProg é um ambiente web desenvolvido pela Universidade Federal do Espírito Santo - UFES, com o propósito de auxiliar crianças, de cinco a oito anos, na aprendizagem inicial de programação.	
<b>BotLogic</b>	<a href="http://botlogic.us/">http://botlogic.us/</a>
BotLogic é um jogo educativo que desafia, desde crianças até adultos. Este se resume em resolver desafios complexos de lógica, usando conceitos básicos de programação.	
<b>Scratch</b>	<a href="https://scratch.mit.edu/">https://scratch.mit.edu/</a>
O <i>Scratch</i> é um ambiente de programação desenvolvido pelo Massachusetts Institute of Technology, que possibilita a criação de histórias interativas, animações, jogos, músicas, dentre outros. Foi criado com o intuito de ensinar conceitos de Lógica de Programação.	
<b>KidsRuby</b>	<a href="http://kidsruby.com/">http://kidsruby.com/</a>
É uma ferramenta de programação, a qual as crianças podem aprender e criar seus programas, em uma interface simples explorando alguns recursos da linguagem Ruby.	
<b>Python</b>	<a href="https://www.python.org/">https://www.python.org/</a>
É uma linguagem de programação <i>open source</i> , onde uma de suas principais características é ser uma linguagem de fácil aprendizado e programação. Estão inclusos as versões Python 2 e Python 3.	
<b>CodeBlocks</b>	<a href="http://www.codeblocks.org/">http://www.codeblocks.org/</a>
É uma plataforma <i>open source</i> para desenvolvimento de <i>softwares</i> em C/C++ e <i>Fortran</i> . A mesma é extensível e altamente configurável.	
<b>Geany</b>	<a href="https://www.geany.org/">https://www.geany.org/</a>
Ambiente de desenvolvimento leve e rápido com suporte para múltiplas linguagens de programação.	
<b>Wolfram</b>	<a href="https://www.wolframalpha.com/">https://www.wolframalpha.com/</a>
É uma linguagem projetada para uma nova geração de programadores, com implementação local ou na nuvem. Possui uma vasta coleção de dados embutidos, algoritmos e métodos.	

Os critérios para escolha dos ambientes de programação foram os ambientes disponíveis para o *hardware* do *Raspberry Pi* e para o sistema operacional *Raspbian*, de fácil usabilidade e que proporcionam aos usuários a possibilidade de aprender a programar de maneira fácil e intuitiva. Como citado anteriormente, todos os ambientes possuem tutorial explicativo que serve de guia para aprendizagem, disponibilizado junto ao sistema. Para os testes dos ambientes, utilizamos exemplos simples que possibilitaram a validação de todos os recursos dos mesmos.

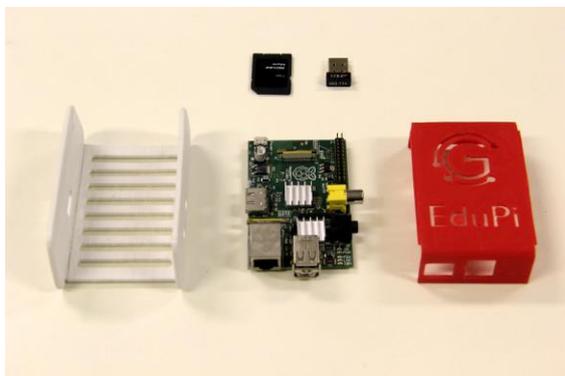
Após a seleção desses ambientes iniciou-se uma nova etapa no desenvolvimento do protótipo. Com o objetivo de proporcionar que o aluno tenha a possibilidade de

iniciar seus estudos e melhorar habilidades que demandem autonomia, os pesquisadores envolvidos no processo buscaram desenvolver ou inserir tutoriais Creative Commons. Esses tutoriais além de explicativos possuem como premissa a facilidade de leitura e entendimento bem como a ideia de engajar o aluno a buscar conhecimento.

Outro ponto, referente à acessibilidade, é a possibilidade de utilizar um teclado virtual, para que, mesmo na ausência do teclado, possa ser utilizado somente com um mouse. Além disto, está disponível também o pacote de escritório LibreOffice e navegador para acesso à *internet*.

### 3.3. Primeira versão

Para a construção da primeira versão do protótipo, decidiu-se pela utilização dos seguintes recursos de *hardware*: uma placa *Raspberry Pi*, um *SD Card* e uma impressora 3D. Abaixo são descritas as características de cada uma delas.



**Figura 1. Componentes do EduPi**



**Figura 2. Protótipo EduPi (primeira versão)**

O *Raspberry Pi Model B* é um minicomputador de baixo custo, com tamanho similar a um cartão de crédito, que pode se conectar a um computador, monitor ou a uma televisão. Foi desenvolvido no Reino Unido pela Fundação *Raspberry Pi*. É *all-in-one*, ou seja, possui todo o *hardware* integrado numa única placa. Seu principal uso está ligado a educação.

Dentre os diferentes modelos existentes, o EduPi está equipado com o modelo *Raspberry Pi Model B Rev 1*, com um processador ARMv6 de 700 MHz, 256 *megabytes* de memória RAM, duas entradas USB 2.0, conexão Ethernet de 100 *megabits* por segundo, saídas *HDMI Full HD*, vídeo composto, saída de som analógica, entrada micro USB de 5 volts e 0.7 amperes para alimentação e slot para cartão SD, usado para o armazenamento.

O *SD Card* foi utilizado para o armazenamento do sistema operacional e dos dados de usuário. O protótipo desenvolvido utiliza um cartão de memória de 16 *gigabytes*, onde  $\frac{1}{4}$  do seu tamanho está sendo utilizado para o sistema operacional, programas e tutorias embarcados. Por fim, a impressora 3D foi utilizada para a impressão da caixa de proteção dos componentes do manuseio intenso e para o transporte do dispositivo.

### 3.4. Versão atual

Com vistas a otimizar a usabilidade, melhorar o desempenho do protótipo inicial, bem como acompanhar as inovações e atualizações das tecnologias nele utilizadas, percebeu-se a necessidade de uma segunda versão. Após algumas pesquisas, o EduPi recebeu novos recursos de *hardware* com maior poder de processamento. Equipado desta vez com o *Raspberry Pi 2 Model B*, o protótipo agora possui 4 entradas USB, 1 *gigabyte* de memória RAM, processador ARM Cortex-7 *quad-core* trabalhando a 900 MHz e uma *GPU Broadcom VideoCore IV*.

As melhorias também ocorreram em nível de sistema operacional, recebendo a atualização do sistema *Raspbian* liberada no dia 18 de março de 2016, a qual traz inúmeras melhorias em comparação à versão anterior do sistema, especialmente no que diz respeito a usabilidade. Também, não menos importante, outras melhorias como atualização de programas e recursos que acompanham o sistema operacional. Os ambientes de programação e robótica também passaram por atualizações e alguns tutoriais foram repaginados, a fim de acompanhar a evolução dos ambientes a que eram destinados. Nessa versão, também houve o acréscimo do ambiente *ArduBlock*.

## 4. Teste de aplicação e resultados

Após um período de amadurecimento do protótipo, como um uso extenso da ferramenta e troca de *hardwares* com vistas à melhora de performance e correção de possíveis problemas passíveis de acontecer durante o uso, tomou-se providências a fazer um teste real com usuários pertinentes ao objetivo do mesmo.

Contando com o auxílio de dois jovens estudantes de escolas públicas do sétimo ano de 13 e 12 anos idade, denominados nesta pesquisa de P1 e P2, respectivamente. P1 possuía conhecimento prévio em informática básica e programação de computadores e já utilizava a ferramenta *Scratch*. O aluno P2, por sua vez, não possui conhecimento básico de programação, estando, portanto, mais próximo do perfil da grande maioria das crianças. Com isso, foi realizado um experimento a fim de explorar o potencial do projeto na visão dos participantes bem como para a obtenção de um *feedback* que embasará possíveis melhorias no EduPi.

Munido de dois EduPis e *hardwares* periféricos iguais, foi fornecido um para cada criança, para que pudessem utilizar a ferramenta de forma livre e assim analisar que tipo de reação teriam com relação aos mais diversos aspectos da ferramenta, desde facilidade de manuseá-la até a performance do *software* em si. As duas crianças utilizaram a ferramenta por um período de uma hora e meia. Logo após, foi realizada uma entrevista com um roteiro preparado de perguntas. A mesma foi registrada em áudio e feita a transcrição. Os principais resultados estão disponibilizados a seguir:

**Tabela 2. Questionário aos participantes**

Participante 1	Participante 2
<b>Para quê utiliza o computador?</b>	
Jogar, redes sociais, <i>Scratch</i> e <i>Construct</i> .	<i>Scratch</i> e redes sociais.
<b>Tem laboratório na escola? Se sim, como ele é?</b>	
Sim, costumam usar as vezes e os computadores são mais ou menos. Laboratório é Linux Educacional.	Sim, computadores <i>desktop</i> com Linux Educacional e bom desempenho.

<b>Dos ambientes presentes no EduPi, qual deles já conhecia?</b>	
Scratch e Python.	Scratch.
<b>Gostou do protótipo?</b>	
Sim, inclusive achou melhor que os computadores existentes na escola, mais rápido e com o sistema operacional mais amigável. Problema: alguns ambientes desatualizados.	Gostou, relatou um problema de um desligamento inesperado enquanto rodava o jogo <i>Minecraft</i> .
<b>Encontrou algum problema referente a desempenho?</b>	
Não, exceto o navegador de <i>internet</i> .	Não.
<b>Encontrou algum problema referente ao visual e a usabilidade?</b>	
Não.	Não.
<b>Achou o visual atrativo?</b>	
Sim, achou o visual bem atrativo.	Visual atrativo igual aos outros computadores que estava acostumado a usar.
<b>Quais os ambientes e/ou programas que utilizou durante o experimento?</b>	
Na grande maioria do tempo <i>Scratch</i> .	Na grande maioria do tempo <i>Scratch</i> .
<b>Encontrou os tutoriais? Se sim, fez uso de algum?</b>	
Sim, mas não utilizou nenhum nesse experimento.	Sim, mas não os utilizou.
<b>Acharia interessante possuir um EduPi?</b>	
Sim eu gostaria de ter um e se tivesse usaria.	Sim.
<b>Na sua opinião, possuir um EduPi lhe auxiliaria no aprendizado de programação e robótica?</b>	
Sim, pelo fato da diversidade de ambientes e principalmente dos tutoriais.	Sim.
<b>E se a escola possuísse um laboratório de EduPis, acharia interessante?</b>	
Sim e acha que os colegas gostariam.	Sim, acha que a maioria dos colegas também.
<b>Entre o computador que você está habituado a usar e o EduPi, qual você escolheria?</b>	
Depende o uso, se fosse programação escolheria o EduPi. Mas para outros usos prefere o computador normal por estar acostumado.	Entre um computador e o EduPi, preferiria usar o EduPi.
<b>Sugestões</b>	
Inserir atalhos na área de trabalho e atualizar os ambientes.	Nenhuma sugestão.

Através das respostas obtidas pelos participantes, identifica-se a necessidade de atualização dos equipamentos que são disponibilizados para os alunos, sendo isso algo inerente ao se considerar a velocidade em que a tecnologia evolui nos dias de hoje. Ademais, o fato de um dos participantes relatar uma amigabilidade e desempenho superior aos computadores disponibilizados pela sua escola agrega muito ao potencial da ferramenta, principalmente por requerer um custo deveras menor se comparado à implantação de laboratórios de computadores *desktop*<sup>4</sup>.

Além disso, o desenvolvimento deste projeto, até o momento, cumpriu com as expectativas dos pesquisadores de provir melhorias ao aparato - conforme as dicas de possíveis usuários - e ao amadurecimento da proposta de disponibilização para um número maior de usuários. O *feedback* gerado pelos participantes, como pôde ser observado acima, serve de grande estímulo na busca de aportes para o desenvolvimento de um experimento educacional completo.

<sup>4</sup> Custo aproximado da atual versão do EduPi: R\$280,00.



Figura 3. Versão atual do EduPi



Figura 4. Estudantes fazendo uso do EduPi

## 5. Considerações finais

De posse dos resultados finais, pode-se chegar à conclusão de que o teste auxiliou os desenvolvedores a angariarem uma nova percepção sobre o protótipo, sendo essa oriunda de possíveis usuários da ferramenta proposta. Esse experimento serviu, ainda, para algumas correções no projeto, bem como para alavancar a ideia inicial de realizar um programa educacional envolvendo o EduPi.

Dessa maneira, como próximos passos, prevê-se a realização de um experimento educacional envolvendo a distribuição de vários protótipos em um ambiente informal de ensino com jovens de 10 a 12 anos. Diante disso, identificar o potencial e a efetividade do protótipo em sustentas processos espontâneos e não direcionados de aprendizagem de programação e robótica, analisando sua apropriação por este público através de um processo de coleta de dados<sup>5</sup> que detalham seu uso.

Como resultado, os possíveis benefícios a serem obtidos através dessas práticas são muitos, desde o favorecimento à criatividade, o trabalho em equipe, a autonomia e a responsabilidade, além de ter de integrar conhecimentos teóricos em situações práticas. Ainda, um outro benefício oportunizado pela robótica educativa se dá no desenvolvimento do raciocínio lógico e abstrato, questões estas derivadas das necessidades envolvidas na programação.

Tomando como inspiração o projeto do MIT já mencionado e tomando proveito do grande leque de possibilidades que a programação de computadores e a robótica educativa podem proporcionar, o EduPi se enquadra perfeitamente em situações onde os recursos disponíveis para essa aquisição de conhecimento sejam menos favoráveis. Sendo assim, a possibilidade de disponibilizar o aparato em escolas públicas, bem como em comunidades carentes, torna-se não só um desafio, mas sim um atrativo extra em busca da massivação da inclusão digital.

---

<sup>5</sup> Através de um *logger* que, com a concessão do participante, ficará reunindo informações a respeito do seu uso e gerará um arquivo o qual conterá os *softwares* e tutoriais que o usuário utilizou, bem como o tempo de uso de cada um deles. O mesmo funciona através de um script que faz uso dos monitores de processos do sistema *Raspbian*, executando em *background* durante a utilização do sistema.

Por fim, vale ressaltar que a experiência de desenvolvimento, teste dos protótipos e realização da atividade com crianças utilizando o dispositivo robótico demonstram o potencial desta tecnologia como alternativa didática de ensino e aprendizagem que, além de criar alternativas para a ação do professor, motiva os alunos a assumir o controle sobre suas aprendizagens a partir da programação do robô.

## Referências

- Arduino. Disponível em: <<http://www.arduino.cc>>. Acesso em 09 de jun. 2015.
- Baranauskas, Maria Cecília Calani et al. Uma taxonomia para ambientes de aprendizado baseados no computador. Valente, JA O computador na sociedade do conhecimento. Campinas, SP: UNICAMP/NIED, 1999.
- BotLogic. Disponível em: <<http://botlogic.us>>. Acesso em 10 de jun. 2015.
- CodeBlocks. Disponível em: <<http://www.codeblocks.org/>>. Acesso em 10 de jun. 2015.
- Elinux, Raspberry Hardware. Disponível em: <[http://elinux.org/RPi\\_Hardware](http://elinux.org/RPi_Hardware)>. Acesso em 11 de jun. 2015.
- Geany. Disponível em: <<http://www.geany.org/Main/About>>. Acesso em 10 de jun. 2015.
- KidsRuby. Disponível em: <<http://kidsruby.com/>>. Acesso em 09 de jun. 2015.
- LibreOffice. Disponível em: <<https://ptbr.libreoffice.org/descubra/libreoffice/>>. Acesso em 10 de jun. 2015.
- NewProg. Disponível em: <<http://www.newprog.com.br/home.php>>. Acesso em 09 de jun. 2015.
- O Que A Maioria Das Escolas Não Ensina. Direção: Lesley Chilcott. Estados Unidos: Grasp lab, University of Pennsylvania, 2013. 1 filme (9min 33seg), son., color. Disponível em: <<http://vimeo.com/73688194>>.
- Papert, Seymour; Valente, Jose Armando; BITELMAN, Beatriz. Logo: computadores e educação. Brasiliense, 1980.
- Python. Disponível em: <<https://www.python.org/about/>>. Acesso em 10 de jun. 2015.
- Raspberry. Disponível em: <<https://www.raspbian.org/>>. Acesso em 09 de jun. 2015.
- Rushkoff, Douglas. As 10 questões essenciais da era digital. Progame seu, 2012.
- S4A. Disponível em: <[http://s4a.cat/index\\_pt.html](http://s4a.cat/index_pt.html)>. Acesso em 09 de jun. 2015.
- Scratch. Disponível em: <<https://scratch.mit.edu/about/>>. Acesso em 10 de jun. 2015.
- Wolfram Alpha. Disponível em: <<http://www.wolframalpha.com/about.html>>. Acesso em 10 de jun. 2015.
- Zapata, Nibaldo; Novales, Miguel; Guzmán, J. La robótica educativa como herramienta de apoyo pedagógico. Concepción: Universidad de Concepción, 2004.