

Concepção e desenvolvimento de uma solução livre e de baixo custo para a aprendizagem de programação e robótica

Gabriel Paludo Licks, Leonardo Costella, Adriano Canabarro Teixeira

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

{138119, 134383, teixeira}@upf.br

***Abstract.** This article reports the designing and development process of a low-cost mini-computer, which uses free technology and is meant to be a platform that enables who uses it to learn how to program computers and robots in a practical and intuitive way. Following the idea of discovery and self-education, it offers a capable ecosystem to be responsible for turning its users into content producers, which makes them protagonists of the reality that surrounds them than just mere ordinary users. Finally, as a next demand, it is aimed an experience providing this technology in large-scale, in order to verify the effectiveness of the learning.*

***Resumo.** Este artigo reporta o processo de concepção e desenvolvimento de um minicomputador de baixo custo, que utiliza de tecnologias livres com vistas a ser uma plataforma que possibilite ao usuário aprender a programar computadores e robôs de forma prática e intuitiva. Seguindo a ideia da descoberta e do autodidatismo, o mesmo oferece um ecossistema capaz de ser responsável por transformar seus usuários em produtores de conteúdo, tornando-os protagonistas da realidade que os cerca do que simples meros usuários comuns. Por fim, como próxima demanda, visa-se uma experiência disponibilizando essa tecnologia em larga escala, buscando verificar a efetivação do aprendizado.*

1. Introdução

Vivemos em um mundo permeado pela tecnologia, no qual, estar inserido na mesma já vem se tornando em alguns momentos não mais opcional, e sim necessário. O ambiente em que estamos inseridos atualmente nos traz de forma inerente uma necessidade de lidarmos com tecnologias digitais nas tarefas mais básicas do dia-a-dia. É nesse sentido que pode ser encontrada uma grande quantidade de pesquisas científicas que avaliam desde dados qualitativos da maneira como as tecnologias estão influenciando a sociedade atual até dados quantitativos e estatísticos que nos trazem os altos índices de sua presença no ambiente em que vivemos.

Hoje, porém, já não basta que sejamos meros usuários dessa tecnologia. A demanda que a constante evolução dessas ferramentas vem nos trazendo por meio da sua influência nos exige passarmos do simples estado de participante/usuário 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 (usuário) ou estaremos no estado de autoria, no qual, poderemos ter total controle sobre nossas ações. Rushkoff (2012) também afirma que essas tecnologias vêm para caracterizar o modo como vivemos e trabalhamos, onde as pessoas que as programam acabam por moldar o nosso mundo.

Os aparatos robóticos existentes atualmente aparecem em especial nas linhas de produção da indústria, na automação de processos em diversas e variadas organizações, bem como em projetos de centros pesquisa. Os mesmos integram soluções de mecânica, eletrônica e programação. Já 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 de o programa estar pronto. Já os aparatos robóticos, por sua vez, podem ser vistos em funcionamento e fornecem uma maior tangibilidade.

Nesse sentido, a robótica e a programação são exploradas nas escolas por meio de ambientes apropriados, pelos quais os alunos conseguem controlar os aparatos robóticos e os programas criados de forma intuitiva. Nessas atividades, os estudantes constroem conhecimentos por meio da análise, observação, planejamento, e reflexão das práticas. Em muitas situações, isso se dá em um ambiente cooperativo entre os mesmos, o que é salutar para a aprendizagem.

Ao programar, o indivíduo 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 estudantes 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 relacionadas à robótica educacional, a fim de agilizar os resultados obtidos pelos estudantes 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, nos quais, as questões relativas a mecânica e eletrônica já estão resolvidas, ficando a ênfase na programação do aparato.

Dessa forma, os possíveis benefícios obtidos por meio dessas práticas são muitos, favorecendo a criatividade, o trabalho em equipe, a autonomia e a responsabilidade, além de ter de integrar conhecimentos teóricos em situações práticas. Ainda, um outro benefício que pode ser oportunizado pela robótica educativa se dá no

desenvolvimento do raciocínio lógico e abstrato, questões estas derivadas das necessidades envolvidas na programação.

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 *software* livres a ser testado em um contexto real de utilização.

A próxima seção (número dois) busca debater a importância que a programação de computadores e dispositivos robóticos possui na aquisição do protagonismo por parte de um indivíduo, o que influencia diretamente na motivação para a concepção (seção número três) e desenvolvimento (seção número quatro) do projeto apresentado neste artigo. Por último (seção número cinco), apresenta-se os resultados atingidos até o momento no que tange o desenvolvimento e teste da aplicação, junto às considerações finais referentes aos próximos passos da pesquisa.

2. Debatendo a importância do protagonismo

Com relação à importância do conhecimento em programação para a obtenção do protagonismo, 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 protagonizados por outros a se transformarem em protagonistas. 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 adquirir 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 estudantes aumenta em 30% o rendimento escolar, principalmente nos conteúdos de Matemática e Interpretação de Textos (O QUE A MAIORIA DAS ESCOLAS NÃO ENSINAM, 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, colocando o papel do professor como facilitador da aprendizagem e o aluno como construtor ativo.

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 papel do professor como facilitador da aprendizagem e o aluno como construtor ativo.

3. Processo de concepção do projeto

Tendo em vista a demanda trazida na seção anterior 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, no qual, 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.

O projeto tem como base a utilização de um dispositivo denominado EduPi - no qual, “Edu” se refere à palavra educação e “Pi” à placa *Raspberry Pi* - que consiste em um ecossistema, todo baseado em *software* e *hardware* livres, a ser disponibilizado a estudantes de escolas públicas e comunidades carentes, observando a forma como o mesmo foi apropriado no aprendizado de programação e robótica, por indivíduos que normalmente não possuem as devidas condições de inserção nesse tipo de aprendizado.

O aparato visa ser apresentado de forma similar ao que foi realizado pelo *Massachusetts Institute of Technology* (MIT), disponível em notícia¹, onde 40 *tablets* foram distribuídos a crianças de duas aldeias na Etiópia, contendo games, livros e filmes - todos em inglês, língua a qual não era dominada pelas crianças. A única instrução

¹ Notícia completa em: <https://goo.gl/aID17f>.

passada pelos pesquisadores às crianças era de como usar o carregador solar com os *tablets*. Sete meses depois do início desse experimento os pesquisadores observaram que algumas crianças já haviam aprendido sozinhas o alfabeto em inglês e já escreviam letras na tela sensível ao toque. Dessa forma, essa auto-alfabetização observada pelos pesquisadores do MIT serve como fonte de inspiração para o projeto EduPi.

Existem também outras experiências desenvolvidas com crianças no que tange a disponibilização de um dispositivo eletrônico e a verificação de sua apropriação: a iniciativa *One Laptop per Child*², que visa prover um *laptop* conectado e com conteúdo e ferramentas colaborativas para crianças pobres ao redor do mundo; ou o projeto Um Computador por Aluno³, no Brasil, com o objetivo de intensificar as tecnologias da informação e da comunicação nas escolas, por meio da distribuição de computadores portáteis aos alunos da rede pública de ensino.

Porém, o EduPi surge, além de um dispositivo para uso geral e de público-alvo de indiferente de idade, como uma plataforma especializada na aprendizagem e utilização para a programação de computadores e robótica.

4. Processo de desenvolvimento do protótipo

Na construção do protótipo, teve-se como componente principal uma placa *Raspberry Pi* (mais detalhes posteriormente), visto ser 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.

Com relação aos *softwares* incluídos no protótipo, 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, passamos ao detalhamento das tecnologias e soluções utilizadas no protótipo.

4.1. Tecnologias utilizadas

Para a construção 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.

O *Raspberry Pi* é um minicomputador de baixo custo, com tamanho similar a um cartão de crédito, que pode conectar-se a um computador, monitor ou a uma TV. Foi desenvolvido no Reino Unido pela Fundação *Raspberry Pi*. É *all-in-one*, ou seja,

² Mais detalhes em: <http://one.laptop.org/>.

³ Mais detalhes em: <http://goo.gl/MYv0jS>.

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 *MicroUSB* 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*, dos quais $\frac{1}{4}$ de sua capacidade está sendo utilizada 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.

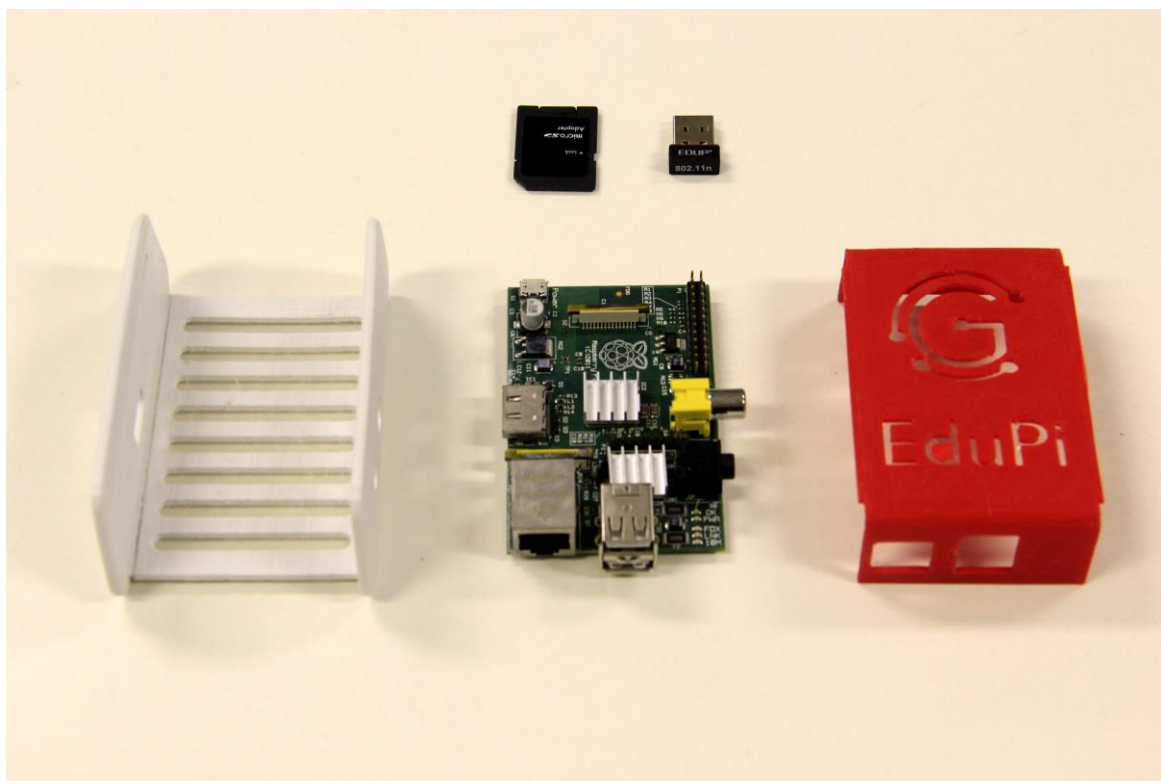


Figura 1. Componentes do EduPi



Figura 2. Protótipo EduPi

4.2. Soluções embarcadas

O EduPi possui como sistema operacional a versão do Raspian disponibilizada no dia 16 de fevereiro de 2015. Esse sistema operacional livre é baseado no Debian Wheezy, e é otimizado para o hardware do protótipo.

A fim de se configurar como uma plataforma completa para o aprendizado de programação e robótica, são disponibilizados ambientes para a prática de robótica educativa e para programação. Cada um deles tem o objetivo de oferecer alternativas com complexidades diferentes a fim de que o mesmo dispositivo possa servir a diferentes perfis de usuários.

As soluções embarcadas disponíveis no sistema do aparato, em ordem crescente de complexidade, são: S4A (Scratch for Arduino); Arduino IDE; NewProg; BotLogic; Scratch; KidsRuby; Python 2 e 3; CodeBlocks; Geany; e Wolfram. Entre os ambientes listados, se configuram como específicos para robótica os dois primeiros, sendo o restante para a prática de programação. Abaixo está uma tabela com características detalhadas sobre cada ambiente e o seu respectivo *website*.

Tabela 1. Características dos ambientes

Scratch for Arduino (S4A)	http://s4a.cat/
S4A é uma modificação do <i>software</i> Scratch que permite uma programação simples para a plataforma de <i>hardware open source</i> Arduino. Ele fornece novos blocos para o gerenciamento de sensores e atuadores conectados ao Arduino. O principal objetivo da inserção do S4A é atrair os estudantes para o mundo da programação de robôs em uma interface amigável e de alto nível.	

Arduino IDE	https://www.arduino.cc
Ambiente de desenvolvimento aberto, que torna possível uma fácil programação para a Placa de prototipação Arduino. A programação é feita em uma linguagem padrão, baseada na linguagem C/C++, a qual incluiu funções capazes de controlar os inúmeros sensores e atuadores.	
NewProg	http://www.newprog.com.br/
O NewProg é um ambiente web desenvolvido pela Universidade Federal do Espírito Santo - UFES, com o propósito de auxiliar crianças, da faixa etária de cinco a oito anos, na aprendizagem inicial de programação de computadores. O mesmo, proporciona uma sensação de entretenimento, enquanto favorece o desenvolvimento de habilidades cognitivas importantes em vários momentos da vida cotidiana, tais como, planejamento e resolução de problemas.	
BotLogic	http://botlogic.us/
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.	
Scratch	https://scratch.mit.edu/
O Scratch é um ambiente de programação desenvolvido pelo MIT - 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.	
KidsRuby	http://kidsruby.com/
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.	
Python	https://www.python.org/
É uma linguagem de programação <i>open source</i> , onde uma de suas principais características é ser uma linguagem de fácil aprendizado e programação. No EduPi, estão inclusos as versões Python 2 e Python 3.	
CodeBlocks	http://www.codeblocks.org/
É uma plataforma <i>opensource</i> para desenvolvimento de <i>softwares</i> em C/C++ e Fortran. A mesma é extensível e altamente configurável.	
Geany	https://www.geany.org/
É um ambiente de desenvolvimento leve e rápido com suporte para múltiplas linguagens de programação.	
Wolfram	https://www.wolframalpha.com/
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 uma nova etapa no desenvolvimento do protótipo. Com o âmbito de proporcionar que o estudante tenha a possibilidade de iniciar seus estudos e melhorar suas habilidades auto-didáticas, os pesquisadores envolvidos no processo buscaram desenvolver ou inserir tutoriais de licença *Creative Commons*. Esses tutoriais além de explicativos possuem como premissa a facilidade de leitura e entendimento bem como a ideia de engajar o estudante a buscar conhecimento.

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

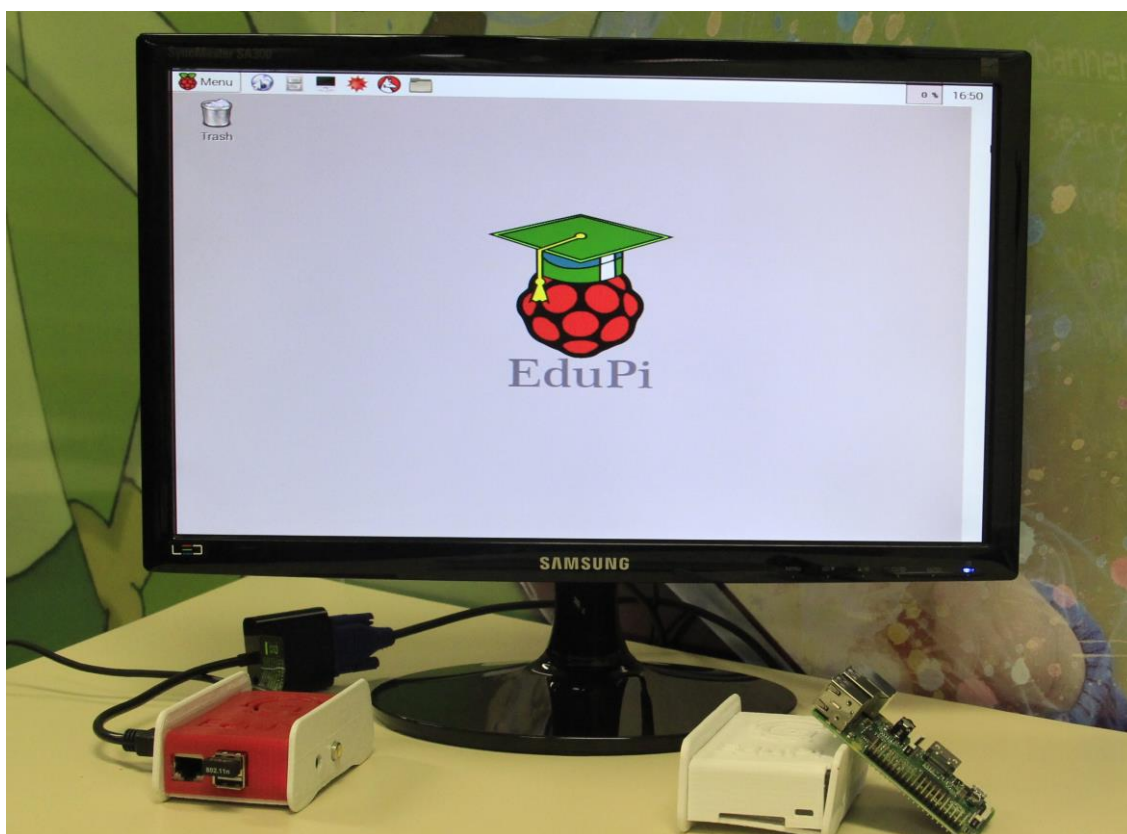


Figura 3. Protótipo em funcionamento

5. Resultados parciais e considerações finais

O desenvolvimento da plataforma, até o momento, para que a mesma se tornasse funcional e de acordo com o que se propunha se mostrou muito desafiador. Tudo isso em função da seletividade e de testes necessários, por exemplo, na busca e desenvolvimento de tutoriais completos e de licença pertinente para disponibilização, bem como na instalação e verificação de desempenho dos ambientes para programação e robótica. Além disso, visando validar a proposta de um dispositivo escalável e de fácil instalação, foram feitos testes tanto com monitores de vídeo conectados através de cabo

VGA e adaptador HDMI como diretamente conectados à entrada HDMI do dispositivo, tanto quanto da mesma forma utilizando televisões tela plana ou curva.

Como próximos passos, tendo o protótipo trabalhando de forma estável e com as funcionalidades propostas, prevê-se a realização de um experimento educacional envolvendo a distribuição de vários dispositivos EduPi em um ambiente informal de ensino envolvendo jovens de 10 a 12 anos e o acompanhamento da forma como ele foi apropriado por este público por meio de um processo de coleta de dados que possibilitem identificar o potencial do protótipo em sustentas processos espontâneos e não direcionados de aprendizagem de programação e robótica. As próximas versões EduPi, serão equipadas com um modelo de *hardware* mais atual, visando uma melhor performance, teste de outros ambientes e sistemas operacionais, bem como a ampliação de portas USB e inserção de rede sem fio melhorando assim sua portabilidade.

Com o apoio das ferramentas presentes nesse ecossistema, tem-se o objetivo de viabilizar aos usufruidores a possibilidade de se tornarem verdadeiros protagonistas. Aproveitando-se do grande leque de possibilidades que a programação e a robótica educativa podem proporcionar, os estudantes tendem a melhorar suas habilidades cognitivas de uma maneira desafiadora, mesmo que indiretamente. Outro ponto interessante a ser mencionado é o despertar de interesses dos estudantes a seguirem carreiras em área afins da computação, sedenta por profissionais.

Tomando como inspiração o projeto do MIT, mencionado na seção três, 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.

Por fim, a experiência de desenvolvimento, testagem do dispositivo e a realização da atividade com 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>. 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 Ensinam. 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. Programe seu, 2012.

S4A. Disponível em: <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.