



Redução de Custos na Implantação de Laboratórios de Inclusão Digital com o uso de Software e Hardware Livres: Um Estudo de Caso

Robson de Sousa Martins

Tema: Relação Governo e Sociedade

Folha de Rosto

Título do Trabalho: Redução de Custos na Implantação de Laboratórios de Inclusão Digital com o uso de Software e Hardware Livres: Um Estudo de Caso.

Tema: Relação Governo e Sociedade

Autor: Robson de Sousa Martins

Currículo: Robson de Sousa Martins é MBA em Desenvolvimento de Soluções Corporativas em Java/SOA, pela Faculdade de Informática e Administração Paulista (FIAP) e Bacharel em Sistemas de Informação, pela Faculdade Batista de Administração e Informática (FBAI). Atuou durante sete anos como Técnico Eletrônico, em projetos de sinalização metroferroviária e manutenção eletrônica, e dez como Analista Programador, no desenvolvimento de softwares comerciais. Desde 2010 é Analista no Serpro, lotado na Superintendência de Suporte a Tecnologias (SUPST), em São Paulo/Luz, onde atuou em projetos como o ALM (*Application Lifecycle Management*) e ECM *Alfresco* (Processo Verde). Atualmente participa dos projetos de implantação das tecnologias de BPMS (*Business Process Management Suite*) e da plataforma *Mobile Marketing*.

Resumo

A inclusão digital sempre foi uma das principais bandeiras sociais levantadas pelo Serpro (Serviço Federal de Processamento de Dados), que através de seu Programa de Inclusão Digital (PSID), permite a muitos cidadãos brasileiros obter acesso a computadores e *internet* nas diferentes e mais remotas localidades do país. Esse fato comprova seu posicionamento como uma empresa parceira nas relações governo/sociedade. Assim sendo, este trabalho se propõe a apresentar alguns conceitos já consagrados no Serpro, como o uso de *software* livre, e outros mais inovadores, como a utilização de placas educacionais de baixo custo e alto poder computacional, mas com especificações de *hardware* livres – normalmente empregadas nas áreas de robótica educacional ou projetos de aficionados por IoT (*Internet of Things* – *Internet* das Coisas); como a junção dessas ideias podem contribuir para a construção de computadores modernos, poderosos e de baixíssimo custo; e por fim, demonstrar um estudo de caso da implantação de um pequeno laboratório comunitário para inclusão digital, priorizando o baixo custo na aquisição e manutenção de componentes de *hardware* e custo zero na aquisição de licenças de *software*, além de oferecer computadores de desempenho muito superior e custo de manutenção inferior se comparado aos computadores normalmente reutilizados nesse tipo de laboratório de informática, tipicamente obsoletos por serem adquiridos através de reciclagem ou doações.

Palavras-chave: Inclusão Digital, *Software* Livre, Redução de Custos, *Hardware* Livre, Computadores Educacionais.

Lista de ilustrações

Figura 1 – Computadores obsoletos.....	7
Figura 2 – Software Livre.....	8
Figura 3 – Hardware Livre.....	9
Figura 4 – Placa Arduino.....	11
Figura 5 – System on Chip (SoC).....	12
Figura 6 – Raspberry Pi.....	13
Figura 7 – Raspberry Pi, Modelos B+ e A+.....	14
Figura 8 – Desktop Raspbian com o Raspberry Pi.....	16
Figura 9 – Estação meteorológica conectada (IoT), com o Raspberry Pi.....	18
Figura 10 – Caixa para fixar o Raspberry Pi no monitor LCD.....	22
Figura 11 – Adaptador HDMI / DVI conectado ao Raspberry Pi.....	22
Figura 12 – Raspbian num grande monitor LCD.....	23

Lista de abreviaturas e siglas

ARM: *Advanced RISC Machine.*

BSD: *Berkeley Software Distribution.*

CDI: Comitê para a Democratização da Informática.

CI: Computadores para Inclusão.

CISC: *Complex Instruction Set Computer.*

CPU: *Central Processing Unit* (Unidade Central de Processamento).

DIY: *Do-It-Yourself* (Faça Você Mesmo).

DSP: *Digital Signal Processor* (Processador Digital de Sinais).

DVI: *Digital Visual Interface.*

FSF: *Free Software Foundation.*

Full HD: *Full High Definition.*

GPIO: *General Purpose Input/Output* (Entrada/Saída para Uso Geral).

GPL: *General Public License.*

GPU: *Graphics Processing Unit* (Unidade de Processamento Gráfico).

H.264/MPEG-4 AVC: *H.264 MPEG level 4, Advanced Video Coding Standard.*

HDMI: *High-Definition Multimedia Interface.*

I2C: *Inter-Integrated Circuit.*

I2S: *Integrated Inter-chip Sound.*

I/O: *Input/Output* (Entrada/Saída).

IoT: *Internet of Things* (Internet das Coisas).

MicroSD: *Micro Secure Digital Card.*

MIPI: *Mobile Industry Processor Interface.*

MPEG: *Moving Picture Experts Group.*

OpenGL ES: *Open Graphics Library for Embedded Systems.*

OSHA: *Open Source Hardware Association.*

OSI: *Open Source Initiative.*

PSID: Programa Serpro de Inclusão Digital.

PSSL: Programa Serpro de *Software* Livre.

RISC: *Reduced Instruction Set Computer.*

SD: *Secure Digital Card.*

SDRAM: *Synchronous Dynamic Random Access Memory.*

Serpro: Serviço Federal de Processamento de Dados.

SoC: *System On Chip* (Sistema num *Chip*).

SPI: *Serial Peripheral Interface.*

TIC: Tecnologia de Informação e Comunicação.

UART: *Universal Asynchronous Receiver/Transmitter.*

USB: *Universal Serial Bus* (Barramento Serial Universal).

VC-1: *Video Codec One.*

Sumário

1	Introdução.....	6
1.1	Justificativa e Objetivo.....	6
2	Iniciativas Existentes.....	6
3	Software Livre.....	8
4	Hardware Livre.....	9
5	Placas Educacionais e Single Board Computer.....	10
5.1	Placa Raspberry Pi.....	13
5.1.1	Hardware.....	14
5.1.2	Sistema Operacional.....	16
5.1.3	Principais Usos.....	17
6	Estudo de Caso.....	19
6.1	Raspberry Pi Versus Desktop PC Tradicional.....	19
6.2	Montagem de um Desktop Raspberry Pi.....	21
6.3	Implantação do Laboratório de Inclusão Digital.....	23
7	Conclusões.....	24
7.1	Melhorias / Trabalhos Derivados.....	25
8	Referências.....	26

1 Introdução

A inclusão digital é uma forma de se democratizar o acesso à informação – e às tecnologias de informação e comunicação (TIC) – a muitos cidadãos, especialmente no Brasil, onde ainda há dificuldades de acesso a computadores e *internet* nas diferentes e mais remotas localidades do país.

Nesse contexto, o Serpro (Serviço Federal de Processamento de Dados) instituiu em 2003 o PSID (Programa Serpro de Inclusão Digital), a partir do qual, algumas ações de responsabilidade social são mantidas até hoje, dentre as quais se destacam: a disponibilização de telecentros comunitários, o Espaço Serpro Cidadão e a Escola de Inclusão Sociodigital (BRASIL, 2003).

Além do Serpro, outras entidades de governo (nas diferentes esferas, como estaduais e municipais), e ainda muitas organizações não-governamentais, têm mantido espaços com laboratórios, tanto para acesso à *internet* (telecentros), assim como também para o ensino de *software*, programação, ou outras tecnologias de informação às comunidades – tanto para crianças, como para jovens ou adultos (SOARES; ALVES, 2013).

1.1 Justificativa e Objetivo

Este trabalho tem por objetivo apresentar o estudo de caso da montagem de um pequeno laboratório, voltado à inclusão digital e ao ensino de tecnologias (como suítes de escritório, *design* gráfico, criação de *sites web*, produção digital de vídeos, e programação de computadores), realizado por uma organização não-governamental e sem fins lucrativos.

Duas premissas nortearam a criação desse laboratório: o baixo custo de aquisição, instalação e manutenção de equipamentos (*hardware*) e de *software*, já que a sua finalidade de oferecer cursos abertos e gratuitos à comunidade local não permitiu a obtenção de substanciais recursos financeiros para investimento; e, em segundo lugar, a necessidade de performance computacional compatível com *softwares* modernos e atualizados.

2 Iniciativas Existentes

Muitos dos laboratórios ou telecentros voltados à inclusão digital têm uma característica em comum: a utilização de equipamentos obtidos através de doações, ou reciclados (recuperados), normalmente provenientes de empresas

que os descartam devido à troca de parque de estações de trabalho. Um exemplo disso é o projeto CI (Computadores para Inclusão), do Governo Federal (BRASIL, 2012).

Se por um lado, esse tipo de iniciativa traz benefícios, como evitar o descarte inadequado de eletroeletrônicos e incluir projetos sociais para realizar a reciclagem e a manutenção dos computadores recebidos (CDI, 1995) – por outro lado, causa a obsolescência tecnológica dos equipamentos em uso nos laboratórios, pois o aumento do poder de processamento (MOORE, 1965; BROCK, 2006) e a evolução do *hardware* e do *software* (DISCO, 1998) seguem numa velocidade espantosa.



Figura 1 – Computadores obsoletos

Outro problema gerado por esse tipo de abordagem é a frequente quebra de equipamentos, exigindo manutenções constantes; pois como são fruto de doações, podem ter a vida útil de seus componentes eletrônicos e mecânicos bastante comprometida já no início da implantação do laboratório.

Desta forma, é muito comum a existência de laboratórios com equipamentos produzidos há cerca de dez anos, ou até mais, o que os torna úteis somente para tarefas simples, como acesso à *internet* e uso de suítes de escritório, mas que inviabilizam a execução de *softwares* para design gráfico, produção de vídeos, ambientes de programação, e outros que exijam um maior poder computacional – que é justamente o núcleo do estudo de caso apresentado neste trabalho.

3 Software Livre

É inegável que o *software* livre traz muitos benefícios em abordagens como a proposta por este trabalho (implantação de um laboratório para inclusão digital), conforme Silveira (2003, p. 442):

O uso do *software* livre nos telecentros e unidades de inclusão digital pode ser um grande incentivo ao surgimento de inúmeras empresas locais capacitadas a configurar e até a desenvolver soluções adequadas aos interesses das empresas e órgãos públicos locais. As duas vantagens mais destacadas no uso do *software* livre para o desenvolvimento econômico-social local são o código-fonte aberto e a inexistência do pagamento de *royalties* pelo seu uso. O código aberto permite que qualquer programador habilitado crie soluções que melhor atendam as necessidades do seu cliente. A inexistência de *royalties* permite que toda a renda gerada pela empresa local de suporte e desenvolvimento fique com ela.

Em primeiro lugar, a inexistência de custo de aquisição ou licenciamento de um *software* livre o torna competitivo na implantação de projetos onde há sérias restrições orçamentárias. Segundo, o código-fonte aberto e o alto poder de personalização e configuração de um *software* livre permite adaptá-lo às necessidades específicas de cada projeto.



Figura 2 – *Software* Livre
(FSF, 2015)

O Serpro, como uma empresa provedora de soluções para o governo federal e para a sociedade brasileira, também se posiciona no direcionamento e implementação de tecnologias de *software* livre, tendo um papel fundamental na disseminação dessas tecnologias em âmbito governamental e estratégico para o país (BRASIL, 2013).

4 Hardware Livre

Um conceito mais recente do que o *software* livre, com surgimento em meados dos anos 2000, é o chamado *hardware* livre, ou *open-source hardware*.

A ideia do *hardware* livre nasceu a partir da mobilização de engenheiros eletrônicos e aficionados pelo movimento DIY (*Do-It-Yourself* – Faça Você Mesmo), que desejavam compartilhar com outras pessoas os diagramas esquemáticos, *layouts* de placas de circuito, definições de *firmware* e outros módulos presentes em suas criações de *hardware*. Esse compartilhamento de informações permite a outros conhecedores de eletrônica a reprodução livre de seus inventos, e a disseminação do uso de seus dispositivos – além de permitir a colaboração mútua em projetos comunitários de *hardware* (GIBB, 2015, p.253-277).



Figura 3 – Hardware Livre
(OSHWA, 2015)

Assim como há diversas organizações e associações em torno do *software* livre, que também oferecem diferentes tipos de licenciamento – por exemplo, alguns totalmente permissivos, como a licença BSD; outros nem tão permissivos, que exigem de volta a contribuição de quem alterar o código original e não permitem o fechamento do código para uso proprietário, como é o caso da licença GPL (OSI, 2015) – assim também é no mundo do *hardware* livre: há uma série de organizações e associações, e vários tipos de licenciamento, alguns incluindo questões complexas, como patentes de propriedade intelectual já existentes para determinados módulos ou semicondutores utilizados nos projetos livres (GIBB, 2015).

Entretanto, há uma diferença fundamental entre *hardware* e *software* livres. Enquanto existe a possibilidade de um *software* livre também ser gratuito (completamente sem custos para sua aquisição), um *hardware* livre não será de

fato gratuito, por ser um produto tangível. Suas especificações até podem ser completamente livres e gratuitas, mas sempre haverá um custo para a sua fabricação (HALFBAKERY, 2004).

Sendo assim, por mais benefícios que um *hardware* livre possa trazer para o desenvolvimento e evolução de seu projeto, além do não aprisionamento da tecnologia nas mãos de uma empresa proprietária, os utilizadores desse *hardware* ainda dependerão de processos industriais de manufatura para obter um produto eletrônico com qualidade e confiabilidade garantida (obviamente fora dos laboratórios de garagem dos *hobistas* amantes do DIY – Faça Você Mesmo).

Dentro desse contexto, e preocupadas com essa questão, algumas empresas e organizações passaram a oferecer produtos de *hardware* livre, porém já montados, encomendados por elas em grande escala para fabricantes de placas de circuito eletrônicos, o que permite o barateamento no custo de produção, e mantendo a qualidade do produto final.

Assim, surgiram associações, como a criadora da placa eletrônica Arduino (ARDUINO, 2005), que além de desenvolver um produto de *hardware* e *software* livres, também passaram a comercializar suas placas industrializadas em larga escala e a baixo custo, favorecendo o usuário final (que tem a opção de montar a sua própria placa seguindo as especificações de *hardware* livres, ou de adquiri-la já montada e testada).

5 Placas Educacionais e *Single Board Computer*

Dentro da filosofia de *hardware* livre apresentada, algumas empresas e organizações criaram projetos de placas e módulos eletrônicos para finalidades educacionais, didáticas, ou mesmo comerciais – especialmente empregados em áreas como robótica, automação residencial e predial, IoT (*Internet of Things* – *Internet* das Coisas), retrocomputação, e muitas outras.

Uma das características mais comuns dessas placas eletrônicas é a presença de um microcontrolador ou microprocessador. Ambos são componentes semicondutores fabricados com a finalidade de controlar componentes periféricos através da execução de um *software* especificamente desenvolvido (normalmente chamado de *firmware*, se estiver gravado dentro deles). Um microprocessador normalmente contém apenas registradores de memória internos e uma unidade de execução de instruções, e necessita do apoio de outros componentes, como memórias voláteis e não-voláteis, controladores de

entrada/saída (I/O) e outros. Já um microcontrolador contém, em um único componente (*chip*), além do microprocessador, vários desses componentes de apoio.

Um exemplo popular de placa eletrônica microcontrolada, com especificações de *hardware* livres é o Arduino (ARDUINO, 2005), criado com a finalidade de facilitar a incorporação de funcionalidades de *hardware* (através de módulos eletrônicos) e de *software* (através de bibliotecas previamente testadas) por quem não conhece profundamente eletrônica e desenvolvimento de *firmware*, ou mesmo para quem necessita de uma solução rápida para projetos maiores. É muito utilizado para fins educacionais, e em áreas como automação, robótica, controle residencial ou industrial, etc.

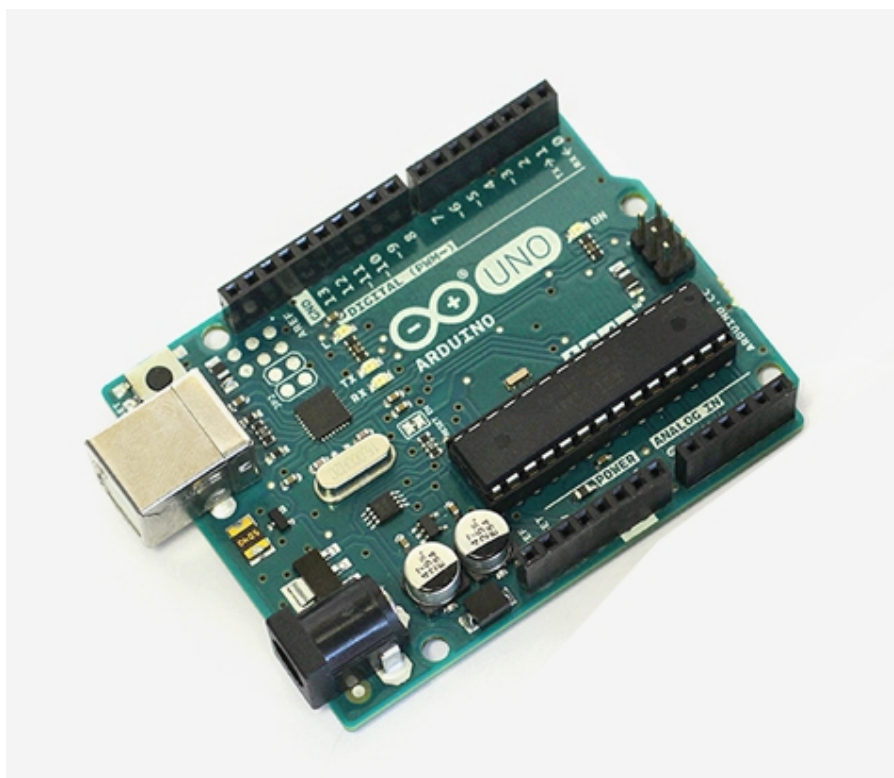


Figura 4 – Placa Arduino
(ARDUINO, 2005)

Atualmente, um conceito eletrônico mais amplo tem sido utilizado, chamado de SoC (*System On Chip* – Sistema num *chip*), onde um único componente semicondutor possui tudo que é necessário para se rodar um sistema computacional completo (desde o microprocessador até placa gráfica com aceleração de *hardware* integrada, incluindo periféricos de conectividade, rede, multimídia, etc.).

Um exemplo atual da aplicação de SoC está nos *smartphones* e *tablets*, pequenos dispositivos em dimensões e peso, com baixo consumo de energia, porém, com um poder computacional cada vez mais espantoso – capazes de rodar aplicações e jogos antes viáveis somente em computadores maiores em tamanho e consumo energético.

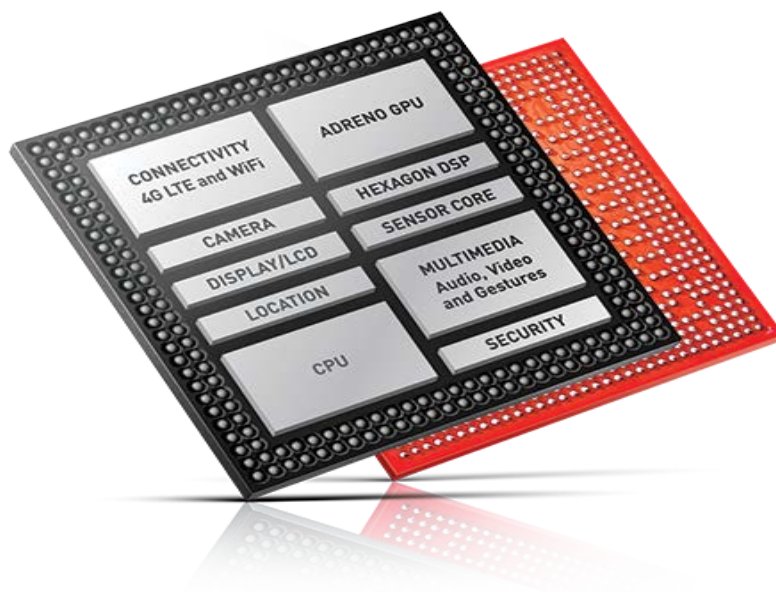


Figura 5 – *System on Chip* (SoC)
(QUALCOMM, 2015)

O conceito de SoC também dá origem a projetos de placas eletrônicas pequenas, de baixo consumo, mas muito poderosas, chamadas de *Single Board Computer* (computador numa única placa).

Várias empresas e organizações criaram projetos de *hardware* livre de placas utilizando SoC, ou seja, placas que contém um computador completo, com especificações de *hardware* livres. Algumas dessas empresas são fabricantes dos próprios *chips* SoC, como a Freescale (uma divisão da Motorola), a Microchip, a Texas Instruments, e outras – como uma forma de promoção e disseminação de uso dos seus próprios componentes dentro da comunidade de *hardware* livre.

Entretanto, um projeto *hardware* livre de *single board computer* para fins educacionais ganhou popularidade nos últimos anos, devido a uma série de características particulares. A seguir são descritas as principais funcionalidades dessa placa, chamada de Raspberry Pi.

5.1 Placa Raspberry Pi

O projeto Raspberry Pi nasceu em 2006, na Universidade de Cambridge, no Reino Unido, quando acadêmicos perceberam dificuldades crescentes de aprendizado pelos alunos de Ciência da Computação, ano após ano. Decidiram então criar uma fundação, e assim, arrecadar recursos para o desenvolvimento de uma placa computacional, com o objetivo de ensinar técnicas de programação às crianças, e treinar seu raciocínio lógico, para que estivessem prontas futuramente para cursar a universidade (RASPBerryPI, 2006).

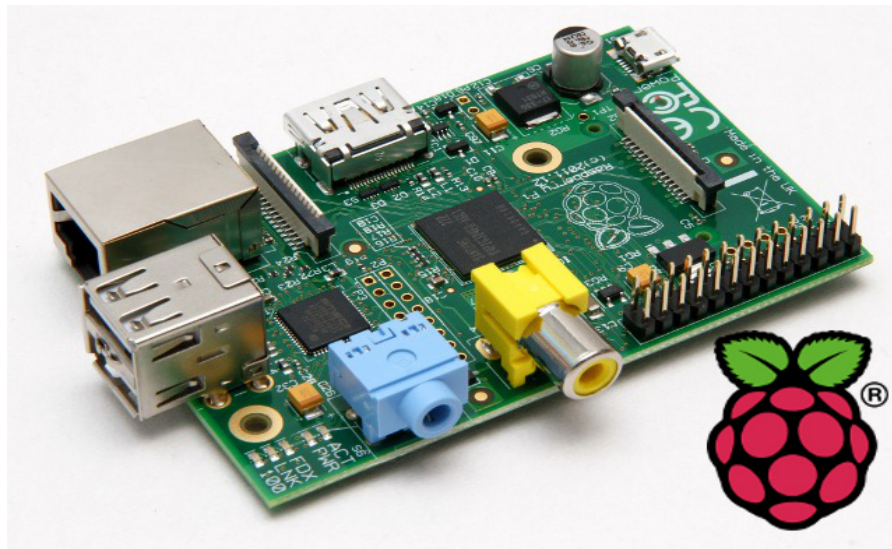


Figura 6 – Raspberry Pi
(RASPBerryPI, 2006)

O projeto de *hardware* livre tomou proporções maiores do que as inicialmente pretendidas e, em 2013, a fundação já havia vendido mais de dois milhões de unidades em todo o mundo. Esse sucesso se deu em grande parte pelo baixo custo da placa (25 dólares, na versão inicial, modelo A); pela sua capacidade de expansão e modularidade (conectores permitem a incorporação de dispositivos como câmera de vídeo, adaptadores de rede sem fio, *bluetooth*, e outros); e ainda pelo seu poder computacional capaz de rodar sistemas operacionais modernos, como *GNU/Linux* – além de permitir programação em linguagens de alto nível, como *Python*, além das tradicionais C e *Assembly*, muito comuns em placas eletrônicas microcontroladas. Todas essas características ainda foram potencializadas ao longo dos anos pela expansão das comunidades de *hardware* e *software* livres que surgiram em torno do Raspberry Pi (UPTON; HALFACREE 2013).

5.1.1 Hardware

O Raspberry Pi tem basicamente três modelos principais de *hardware*: o modelo A, de mais baixo custo, que não conta com *interface* de rede cabeada (*ethernet*) e tem menores dimensões, o modelo B, que oferece a *interface ethernet*, a um custo levemente maior, e o “*compute module*”, que é basicamente somente o chip SoC soldado a uma placa com conectores semelhantes a um módulo, destinado a ser embarcado em aplicações eletrônicas de caráter industrial ou comercial (RASPBERRYPI, 2006). Este último modelo não será abordado por não ser o foco deste trabalho.

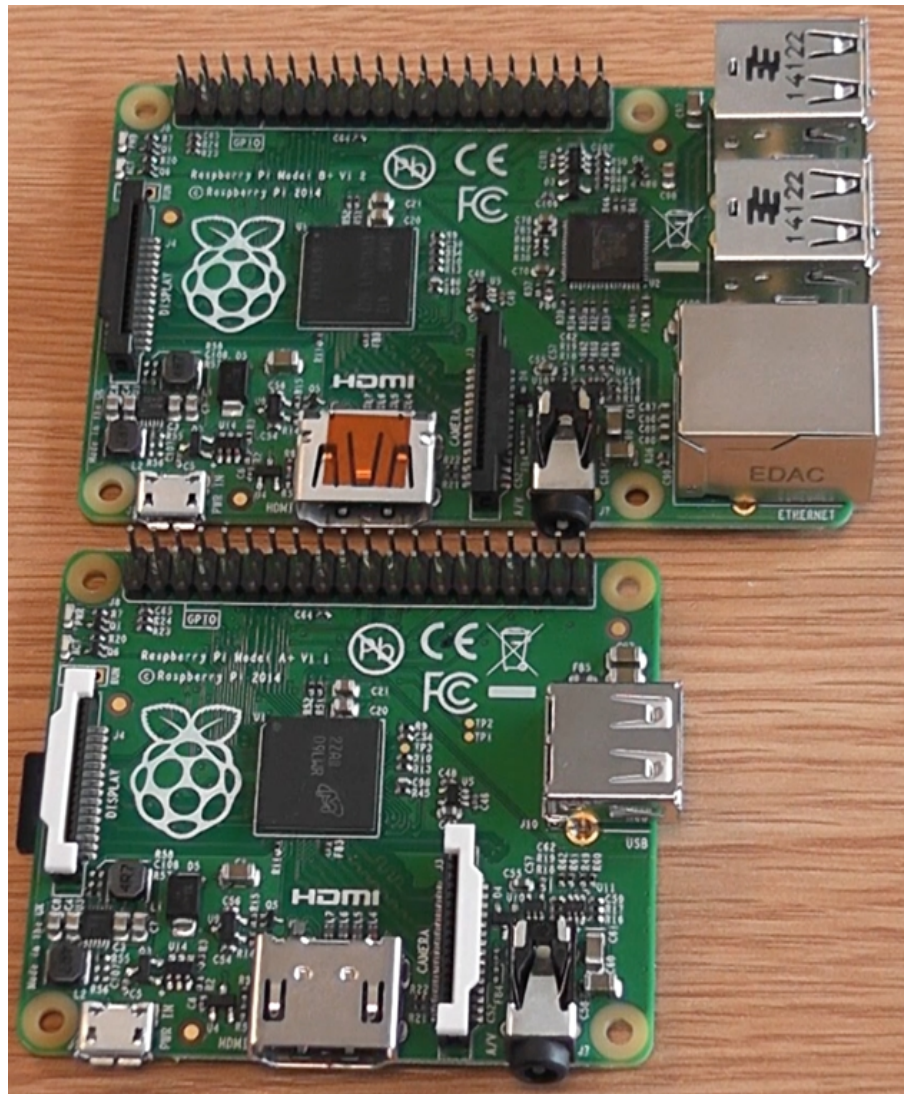


Figura 7 – Raspberry Pi, Modelos B+ e A+ (RASPBERRYPI, 2006)

As principais especificações de *hardware* de ambos modelos são (RASPBERRYPI, 2006):

- **SoC:** Broadcom BCM2835, no Raspberry Pi 1; e Broadcom BCM2836 no Raspberry Pi 2 (possuem CPU, GPU, DSP, SDRAM, e uma porta USB, integrados num único *chip*).
- **CPU:** Arquitetura RISC: ARM11(v6), *single-core*, com *clock* de 700 MHz no Raspberry Pi 1; ARM(v7) Cortex-A7 *quad-core*, com *clock* de 900 MHz no Raspberry Pi 2.
- **GPU:** Broadcom VideoCore IV, com *clock* de 250 MHz (suporta *OpenGL ES 2.0*; decodificação acelerada de H.264/MPEG-4 AVC, MPEG-2 e VC-1; resolução até 1080p – *Full HD*).
- **SDRAM (Memória):** 256MB no Raspberry Pi 1, Modelos A e A+; 512MB nos Modelos B e B+; 1GB no Raspberry Pi 2 Modelo B. Em qualquer modelo, a memória é compartilhada entre a CPU e a GPU.
- **USB:** 1 Porta nos modelos A e A+; 2 Portas no Raspberry Pi 1 Modelo B; 4 portas no Raspberry Pi 1 Modelo B+ e no Raspberry Pi 2 Modelo B (é usado um *chip* de *hub* USB na placa para prover as portas além da única integrada no SoC).
- **Saídas de Vídeo:** Vídeo composto ou HDMI até a resolução 1920x1200.
- **Saídas de Áudio:** Analógico (estéreo) ou digital via HDMI.
- **Rede (Ethernet):** 1 porta 10/100 Mbit (somente nos modelos B e B+).
- **Armazenamento:** Entrada para cartão MicroSD (SD nos modelos A e B do Raspberry Pi 1).
- **Alimentação:** Fonte única de 5V, consumo entre 1W a 1,5W para os modelos A e A+ e entre 3W a 3,5W para os modelos B e B+, e 4W para o Raspberry Pi 2 Modelo B.
- **Conectividade Extra:** GPIO – Pinos de I/O para uso geral (controle de outros dispositivos); UART; barramentos I2C e SPI; I2S; interface MIPI para câmera de vídeo.
- **Custo:** O Raspberry Pi 1 Modelo A+ custa 20 dólares. O Modelo B+ é vendido por 25 dólares. Já o Raspberry Pi 2 Modelo B está 35 dólares. Estes são os preços internacionais praticados atualmente pelos principais parceiros comerciais da fundação Raspberry Pi (PREMIER FARNELL, 2015).

5.1.2 Sistema Operacional

O Raspberry Pi suporta a execução de qualquer sistema operacional que conte com uma versão compilada para processadores de arquitetura ARMv6, para Raspberry Pi 1; ou ARMv7, para Raspberry Pi 2 (RASPBERRYPI, 2006). Alguns deles são:

- **GNU/Linux** (várias distribuições – como por exemplo o Raspbian, que é um Debian GNU/Linux empacotado especialmente para o Raspberry Pi, o Ubuntu, o OSMC, o OpenElec, o Arch Linux, etc.);
- **RISC OS** (sistema não-Unix desenvolvido especialmente para a arquitetura ARM, com otimizações em código *Assembly*);
- **BSD Unix** (FreeBSD, NetBSD);
- **Plan 9, Inferno** (sistemas não-Unix criados pela *Bell Labs* e AT&T);
- **AROS** (originário do AmigaOS);
- **Microsoft Windows 10 IoT Core** (somente para o Raspberry Pi 2).

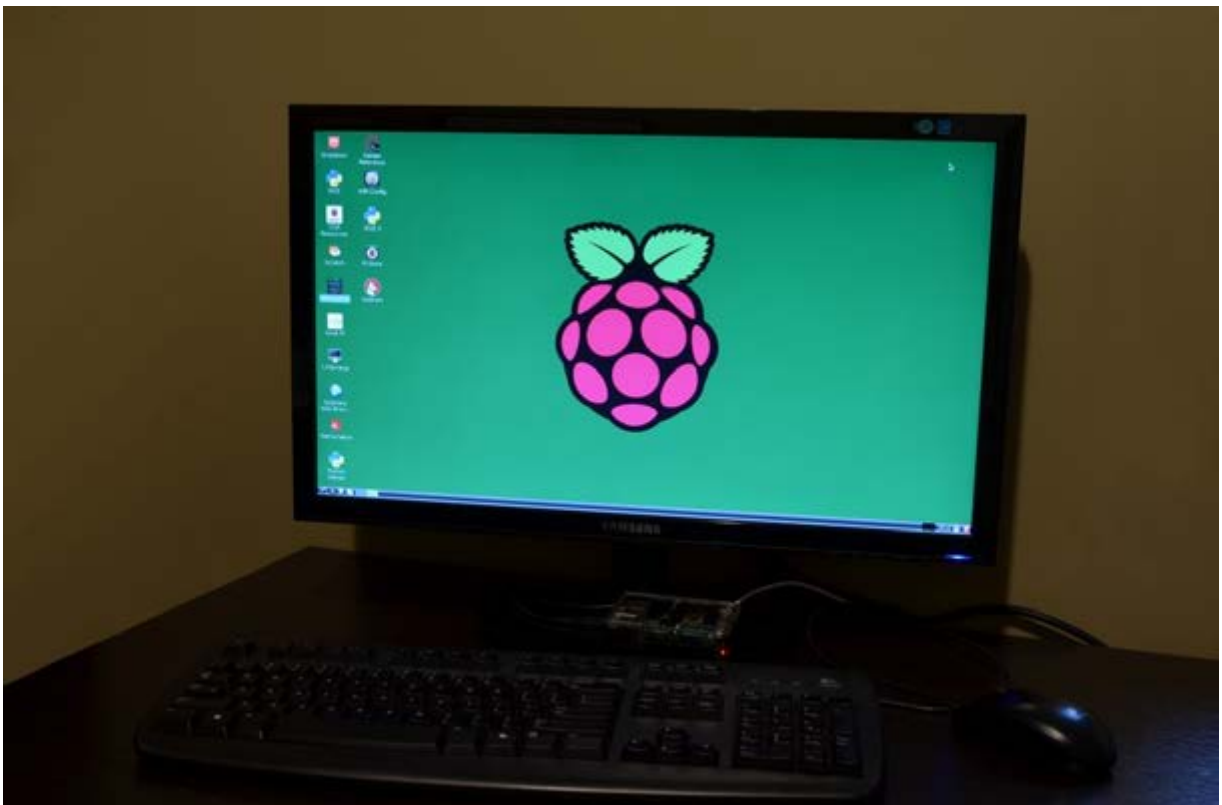


Figura 8 – Desktop Raspbian com o Raspberry Pi

5.1.3 Principais Usos

Apesar do projeto Raspberry Pi ter nascido para fins educacionais, entusiastas de *software* e *hardware* livres, estudantes, hobistas, engenheiros – e até mesmo indústrias e empresas – encontram a cada dia, novas aplicações para esse poderoso *single computer board* (HACKADAY, 2015; INSTRUCTABLES, 2015). Aqui estão alguns exemplos mais comuns:

- **Retrocomputação:** através de emuladores ou virtualização, é possível transformar o Raspberry Pi em um dispositivo capaz de rodar sistemas e programas antigos, por exemplo, *softwares* ou jogos para os arcaicos MSX, Atari, Megadrive, MS-DOS, ou outros. Há quem recrie famosos fliperamas ou outras saudosas máquinas, usando um Raspberry Pi como cérebro.
- **Robótica:** devido ao alto poder de processamento e conectividade do Raspberry Pi, ele se torna um ótimo controlador para diversos tipos de robôs ou máquinas automatizadas. É possível realizar o controle de motores de passo, a leitura de sensores, o processamento de imagens obtidas através de uma câmera e a comunicação remota através de *bluetooth* ou algum tipo de rede sem fio.
- **Media Center:** como o Raspberry Pi possui um decodificador de vídeo com aceleração de *hardware*, e saídas HDMI e vídeo composto, é possível utilizá-lo como uma central de mídia para residências. Através das portas USB é possível conectar um HD (disco rígido) externo para armazenamento de músicas, fotografias, vídeos e filmes; um conjunto de teclado/mouse sem fio para controle, ou mesmo conectar um sensor infravermelho para que ele seja operado a partir de um controle remoto comum da TV.
- **Servidor de Arquivos (*storage*):** conectando um HD externo USB, e instalando os *softwares* apropriados, é possível criar um servidor de arquivos, *web*, de impressão, ou um servidor de compartilhamento de arquivos P2P (*Point-To-Point*), com a vantagem de ter um baixo consumo energético, se comparado a um computador convencional.
- **Segurança e Automação Residencial:** pelo seu baixo consumo energético, bom poder de processamento e fácil integração com outros dispositivos (como redes sem fio, câmeras, placas com relés, sensores, etc.), o Raspberry Pi também tem diversas aplicações na área de monitoramento por câmeras, controle de acesso e de presença, controle de iluminação e

de outros aparelhos domésticos e eletroeletrônicos em uma residência (casas inteligentes).

- **Automação Industrial e Controle de Máquinas:** integrados a CLP's (Controladores Lógicos Programáveis), ou no controle de máquinas computadorizadas – como os tornos e fresadoras CNC (Comando Numérico Computadorizado), os *single computer boards*, como o Raspberry Pi, têm encontrado aplicação dentro das indústrias. Outra área bastante comum dentro das indústrias e laboratórios de desenvolvimento de produtos está na instrumentação eletrônica e no monitoramento de ambientes (como controle de temperatura, pressão, umidade; medidas de tensão, corrente ou potência elétrica).
- **IoT (*Internet Of Things* – *Internet* das Coisas):** pelo seu poder de conectividade com redes sem fio (através de adaptador USB) e a presença de vários tipos de interfaces para controle de circuitos externos, o Raspberry Pi se torna uma placa muito utilizada para projetos de IoT. A possibilidade da programação em *Python* para controle dos pinos de I/O e conexão fácil com os protocolos *internet* mais consagrados (como HTTP, REST, SOAP, etc.) proporciona uma ótima ferramenta para se conectar qualquer dispositivo eletrônico à *internet* e disponibilizar seus dados e controles como serviços acessíveis, até mesmo através da nuvem (*cloud computing*).

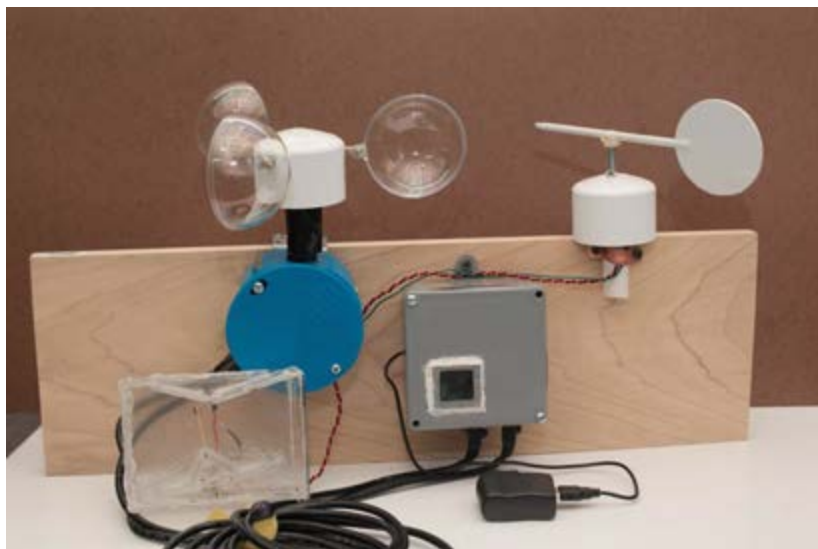


Figura 9 – Estação meteorológica conectada (IoT), com o Raspberry Pi (INSTRUCTABLES, 2015)

6 Estudo de Caso

Após apresentar todas as características e vantagens proporcionadas por um *single computer board* como o Raspberry Pi, a ideia central deste trabalho é demonstrar como computadores *desktop* montados a partir dessa placa podem ajudar na criação de um laboratório de inclusão digital moderno, pronto para rodar *softwares* atuais e ao mesmo tempo, de baixo custo de implantação e manutenção.

Para tanto, é necessário fazer uma observação sobre comparativos entre computadores PC tradicionais e uma placa Raspberry Pi, em termos de performance e custo.

6.1 Raspberry Pi Versus Desktop PC Tradicional

Uma das dificuldades de comparação técnica e metodológica em termos de *benchmarking* (performance) entre um computador arquitetura x86 PC e uma placa baseada na arquitetura ARM, como é o Raspberry Pi, é exatamente a diferença entre as arquiteturas de CPU. Enquanto os computadores tradicionais baseados na arquitetura Intel x86 são CISC (*Complex Instruction Set Computer* – Computador com um Conjunto Complexo de Instruções), os SoC ARM são RISC (*Reduced Instruction Set Computer* – Computador com um Conjunto Reduzido de Instruções). Isso faz toda a diferença na forma como os *softwares* são executados, pois normalmente uma CPU CISC possui um número limitado de registradores internos e precisa contar com mais memória (que é externa), e uma boa velocidade de *clock* para realizar as operações com velocidade aceitável. Em contrapartida, uma CPU RISC possui um número grande de registradores internos, e muitas vezes, a própria memória é tratada como se cada endereço seu também fosse um registrador interno à CPU. Com isso, a tendência é de que uma CPU RISC possa operar com menos memória e menor velocidade de *clock* para realizar suas operações (BHANDARJAR; CLARK, 1991).

Por esse motivo, uma placa como o Raspberry Pi 2 Modelo B, funciona bem com 900MHz de *clock*, mesmo sendo *quad-core*, e memória de apenas 1GB (ainda compartilhada com a GPU), enquanto computadores x86 atuais passam de 3GHz, com memória de pelo menos 3GB, para que consigam rodar os *softwares* mais modernos.

Devido a essa dificuldade metodológica de comparação (*benchmarking*) de performance entre as duas plataformas, alguns testes práticos foram realizados em laboratório, para experimentar a sensação real de um usuário em três sistemas distintos.

Em um dos testes, um usuário não-avançado foi convidado a utilizar a *internet* (usando o *browser* Firefox) e aplicativos de escritório (LibreOffice), numa máquina AMD FX *octal-core*, *clock* de 3,5GHz, 8GB de RAM, placa gráfica AMD Radeon, sistema operacional Debian 8.1 com *desktop* Mate para plataforma amd64 (64 bits). A experiência do usuário foi fantástica, apesar dele não saber que estava utilizando esse computador de última geração da arquitetura x86 de 64 bits.

Num segundo momento, o mesmo usuário realizou as mesmas atividades, porém, em vez de utilizar a máquina AMD FX, foi conectado no mesmo conjunto monitor, teclado, mouse e caixas de som, um computador *single-box* com CPU Intel Atom *dual-core* com 1,5GHz de *clock* e 1GB de memória. O sistema operacional foi o mesmo, porém para a arquitetura x86 de 32 bits. Novamente o computador foi ocultado do usuário para que ele não soubesse qual o *hardware* empregado. O resultado: o usuário reclamou da lentidão em todas as atividades, em especial ao abrir a suíte LibreOffice e ao navegar em *sites web* com mídia (vídeos em *streaming*).

No terceiro teste, foi conectado um Raspberry Pi 2 Modelo B, com um cartão microSD de 8GB, no mesmo conjunto monitor, teclado, mouse e caixas de som. O sistema operacional foi o Raspbian, com um *desktop* levemente diferente dos anteriores. O usuário novamente efetuou as mesmas tarefas, mas nesse caso, ficou muito mais satisfeito do que no segundo teste.

Após os testes, o usuário preencheu um pequeno questionário, pontuando algumas experiências nos três testes. O resultado final demonstrou que o AMD FX foi o sistema ideal para ele, em termos de velocidade de resposta (conforme esperado, por se tratar de um sistema computacional *high-end*), e em segundo lugar, o Raspberry Pi. O sistema Intel Atom foi frustrante e rejeitado em todos os quesitos – mais até do que o esperado, por se tratar de um equipamento comercialmente vendido por uma conceituada marca de computadores.

Em uma análise mais técnica, foram apurados os custos e consumos de energia (aproximados) de cada um desses sistemas. Os resultados dessa análise estão demonstrados a seguir.

Tabela 1 – Comparativo entre computadores tradicionais e o Raspberry Pi

Sistema	Experiência do Usuário	Custo Aproximado (sem periféricos)	Consumo Energético Aproximado
AMD FX 8320 (3,5GHz)	Ótima	R\$ 3.000,00	125W (só CPU)
Intel Atom (1,5GHz)	Péssima	R\$ 600,00	8,5W (só CPU)
Raspberry Pi 2 (900 MHz)	Boa	R\$ 180,00	4W (toda a placa)

Analisando os resultados desse teste comparativo, pode-se perceber que há outros fatores que influenciam na velocidade obtida pelo Raspberry Pi, além da própria CPU e da arquitetura. Por exemplo, o sistema operacional reside em um cartão de memória microSD, naturalmente mais veloz do que discos rígidos tradicionais (mais ainda do que o HD que equipa o sistema com Intel Atom, que é um modelo de 5400 RPM). Outro fator preponderante é a velocidade gráfica, já que a GPU integrada ao SoC do Raspberry Pi possui aceleração de *hardware*, muito superior ao controlador gráfico da Intel integrado à CPU Atom (para o usuário, as experiências de *video streaming* foram tão fluidas no Raspberry Pi como no sistema AMD FX com placa gráfica dedicada).

6.2 Montagem de um Desktop Raspberry Pi

A montagem de um computador *desktop* a partir do Raspberry Pi se torna uma tarefa simples, pois requer somente os seguintes componentes:

- Placa Raspberry Pi (utilizado neste caso, o Raspberry Pi 2 Modelo B);
- Cartão MicroSD para armazenamento do sistema operacional (utilizado um cartão de 8GB);
- Monitor LCD (utilizado um monitor de 15 polegadas com entrada DVI);
- Adaptador HDMI / DVI (porque o monitor utilizado não tem entrada HDMI);
- Teclado e mouse USB; caixas de som;
- Fonte de alimentação de 5V x 2,5A;
- Um *hub* USB para permitir ao usuário conectar pendrives;
- Uma caixa (*case*) para fixar o Raspberry Pi na parte traseira do monitor.

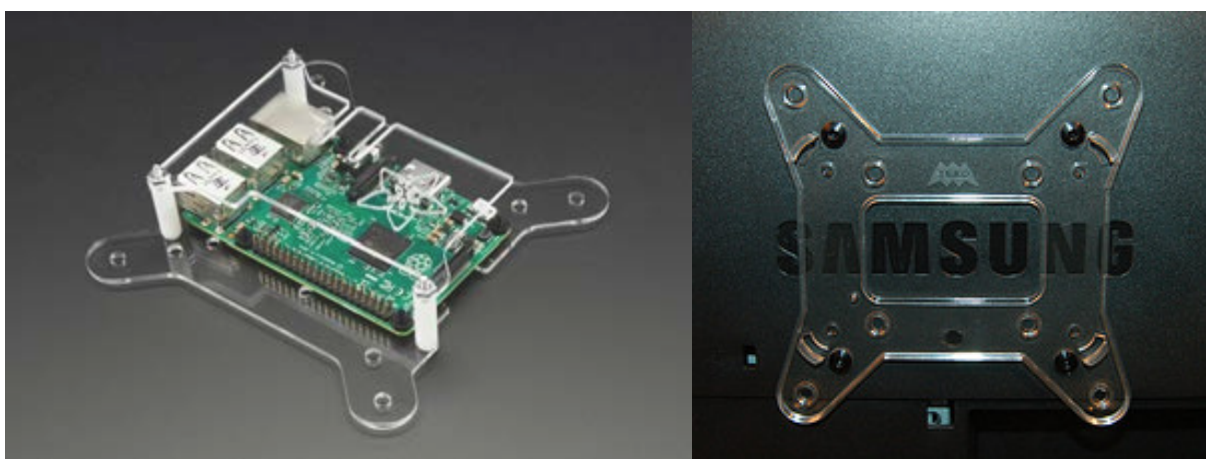


Figura 10 – Caixa para fixar o Raspberry Pi no monitor LCD



Figura 11 – Adaptador HDMI / DVI conectado ao Raspberry Pi

Como sistema operacional, optou-se pelo Raspbian, que já traz por padrão as ferramentas para desenvolvimento em *Python* e compiladores GCC para desenvolvimento em C e C++. Alguns outros *softwares* livres foram também instalados, como o Firefox, LibreOffice, Audacity, Inkscape, Gimp, Cinelera, Pitivi, Apache, ProFTPd.

6.3 Implantação do Laboratório de Inclusão Digital

O laboratório de inclusão digital implantado, alvo do estudo de caso apresentado neste trabalho, conta com cinco computadores montados a partir do Raspberry Pi, interligados em rede cabeada a um roteador conectado à *internet* via banda larga (esse roteador possui um *firewall* integrado, e ainda a possibilidade de gerenciar bloqueio de conteúdos impróprios). Um dos computadores utiliza um monitor/TV LCD de 42 polegadas, fixado na parede, interligado ao Raspberry Pi diretamente via conexão HDMI, e serve para que um instrutor possa exibir conteúdo para os alunos, durante a ministração de aulas.

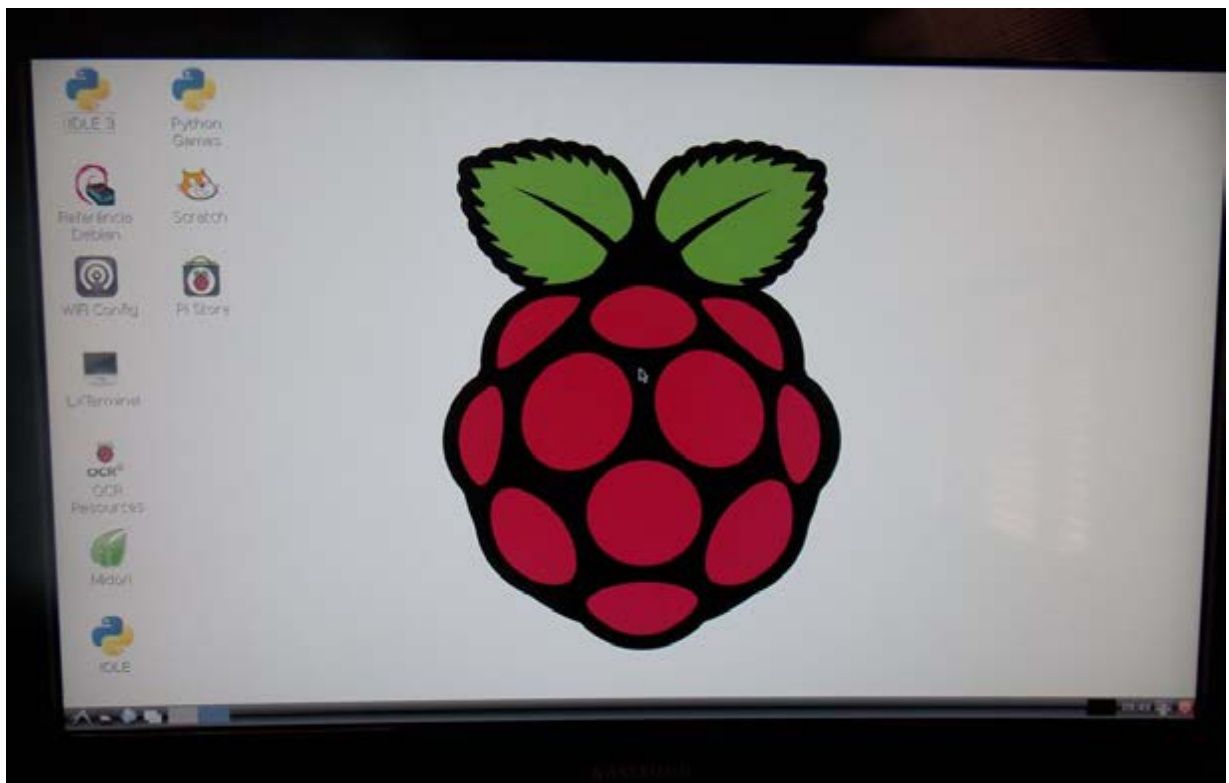


Figura 12 – Raspbian num grande monitor LCD

A aparência das máquinas são de *all-in-one computer*, ou seja, um “computador tudo em um”, pois não há um gabinete além do próprio monitor LCD. Outra característica é o silêncio das máquinas, pois não há necessidade de ventiladores (*fan cooler*) no Raspberry Pi.

Uma alteração de configuração foi realizada no sistema operacional para que o diretório “/home” dos usuários fosse montado em um *RAM Disk* (“disco em

memória”), de forma que o cartão microSD não seja “consumido” pelos arquivos gerados pelos usuários. Os *hubs* USB são justamente deixados à frente dos monitores para permitir que os usuários conectem pendrives, e armazenem o seu material produzido.

Atualmente nesse laboratório, são ministradas aulas de informática básica (suítes de escritório e *internet*), aulas de design gráfico e *web*, edição e produção de vídeos e de programação de computadores (em linguagens *Python* e C/C++), todas abertas à comunidade local.

Em operação há cerca de seis meses, ainda não houve necessidade de substituição de placas ou monitores. Problemas mais comuns ocorrem com mouses ou teclados defeituosos, que são facilmente substituídos.

7 Conclusões

Diante do exposto, e considerando as experiências descritas por outros usuários da comunidade Raspberry Pi, como Hartley (2015), é plenamente viável utilizar o poder computacional desse tipo de *single computer board*, aliado aos benefícios do *hardware* e *software* livres, para a criação de computadores *desktop* modernos, de baixo custo e baixo consumo energético.

Claro que a performance desse tipo de solução não está equiparada a computadores *high-end* (topo de linha), servindo então somente para aplicações leves ou intermediárias em termos de desempenho.

Entretanto, ao se realizar uma comparação entre um *single board computer* como o Raspberry Pi 2 e um computador PC x86 popular fabricado há mais de cinco anos, pode-se perceber alguma competitividade na performance, ganhos consideráveis no custo (especialmente de manutenção), e na diferença de consumo energético.

Sendo assim, essa é uma solução viável, de baixo custo, e moderna para a implantação de pequenos laboratórios de inclusão digital que têm investimento limitado, mas que lutam para compartilhar o conhecimento da informação com a sociedade e a comunidade onde estão inseridos.

7.1 Melhorias / Trabalhos Derivados

No Brasil, o principal problema na utilização de *hardware* livre é o custo desse tipo de produto, adquirido somente via importação, o que acarreta no pagamento de taxas e impostos extras. Por exemplo, um Raspberry Pi 2 Modelo B, que custa 35 dólares nos Estados Unidos, é encontrado no mercado nacional, em lojas que fazem a importação legal, por mais de 280 reais.

Uma possibilidade de disseminação dessa cultura e filosofia de *hardware* livre, com participação mais ativa dos brasileiros nessas comunidades, seria a montagem das placas – cada vez mais populares – dentro do país, como são feitos com os produtos eletroeletrônicos na Zona Franca de Manaus.

O caso de uso da implantação do laboratório de inclusão digital é apenas uma possível aplicação das tecnologias mais recentes de *hardware* e *software* livres para a resolução de problemas encontrados na sociedade atual. Por isso, este trabalho também serve como ponto de partida para o desenvolvimento de outros temas em favor do cidadão, como por exemplo:

- O uso de IoT (*internet Of Things* – *Internet* das Coisas) para solucionar problemas cotidianos nas cidades, como sensores e semáforos interligados para melhorar ou fiscalizar o fluxo de automóveis em uma via, e informar em tempo real aos motoristas as condições reais de trânsito;
- Monitoramento por imagens dos bairros, com reconhecimento de rostos de suspeitos, em aplicações de segurança pública;
- Desenvolvimento de placas eletrônicas educacionais para escolas, para desenvolvimento do aprendizado e raciocínio lógico de crianças.

8 Referências

- ARDUINO, Arduino Project. **Arduino Project Home Page**. 2005. Disponível em: <<https://www.arduino.cc/>> Acesso em: 07 ago. 2015.
- BHANDARJAR, Dileep; CLARK, Douglas W. **Performance from Architecture: Comparing a RISC and a CISC with Similar Hardware Organization**. New York: Communications of the ACM, 1991.
- BRASIL, Ministério das Comunicações. **Projeto CI (Computadores para Inclusão)**. Brasília, 2012. Disponível em: <<http://www.mc.gov.br/acoes-e-programas/centros-de-recondicionamento-de-computadores-crcs/256-temas/centros-de-recondicionamento-de-computadores-crcs/24988-computadores-para-inclusao>> Acesso em: 07 ago. 2015.
- _____, Serviço Federal de Processamento de Dados. **Programa Serpro de Inclusão Digital**. Brasília, 2003. Disponível em: <<https://www.serpro.gov.br/sobre/responsabilidade-social/inclusao-digital>> Acesso em: 07 ago. 2015.
- _____, _____. **Programa Serpro de Software Livre**. Brasília, 2013. Disponível em: <https://www.serpro.gov.br/sobre/responsabilidade-social/conteudo/folder-software-livre/@@download/file/FolderA3Serpro_SoftwareLivre.pdf> Acesso em: 07 ago. 2015.
- BROCK, David C. **Understanding Moore's law: four decades of innovation**. Philadelphia, Pa: Chemical Heritage Press, 2006. ISBN: 094-190141-6.
- CDI, Comitê para a Democratização da Informática. **Doação de equipamentos**. Rio de Janeiro, 1995. Disponível em: <<http://www.cdi.org.br/doacao-de-equipamentos>> Acesso em: 07 ago. 2015.
- DISCO, Cornelius; VEN DER MEULEN, Barend. **Getting new technologies together**. New York: Walter de Gruyter, 1998. ISBN: 311-015630-X.
- FERREIRA, Rayssa R. et al. **Uma Visão sobre Laboratórios On-line**. Anais dos Simpósios de Informática do IFNMG, Campus Januária, 2014. Disponível em: <<http://200.131.5.234/ojs/index.php/anaisviiiisimposio/article/view/44>> Acesso em: 07 ago. 2015.

FSF, Free Software Foundation. **Working together for free software**. 2015. Disponível em: <<http://www.fsf.org/>> Acesso em: 07 ago. 2015.

GIBB, Alicia. **Building Open Source Hardware: DIY Manufacturing for Hackers and Makers**. New York: Addison-Wesley, 2015.

HACKADAY, Hackaday Community. **Projects tagged with "raspberry pi"**. 2015. Disponível em: <<https://hackaday.io/projects/tag/raspberry%20pi>> Acesso em: 07 ago. 2015.

HALFBACKERY, The Halfbakery Blog. **Open Source Hardware Initiative**. 2004. Disponível em: <http://www.halfbakery.com/idea/Open_20Source_20Hardware_20Initiative> Acesso em: 07 ago. 2015.

HARTLEY, Matt. **Raspberry Pi As Your Next Linux PC**. 2015. Disponível em: <<http://www.datamation.com/open-source/raspberry-pi-as-your-next-linux-pc.html>> Acesso em: 07 ago. 2015.

INSTRUCTABLES, DIY How To Make Instructions. **Raspberry Pi Projects**. 2015. Disponível em: <<http://www.instructables.com/id/Raspberry-Pi-Projects/>> Acesso em: 07 ago. 2015.

MOORE, Gordon E. **Cramming more components onto integrated circuits**. IEEE, 1965. Disponível em: <www.cs.utexas.edu/~fussell/courses/cs352h/papers/moore.pdf> Acesso em: 07 ago. 2015.

OSHWA, Open Source Hardware Association. **The Open Source Hardware Association**. 2015. Disponível em: <<http://www.oshwa.org/>> Acesso em: 07 ago. 2015.

OSI, Open Source Initiative. **The Open Source Initiative**. 2015. Disponível em: <<http://opensource.org/>> Acesso em: 07 ago. 2015.

PREMIER FARNELL, Element 14 Community. **Raspberry Pi**. 2015. Disponível em: <<http://www.element14.com/community/community/raspberry-pi?ICID=hp-raspberrypi-topgroups>> Acesso em: 07 ago. 2015.

QUALCOMM, Qualcomm Technologies, Inc. **Qualcomm Snapdragon**. 2015. Disponível em: <<http://www.qualcomm.com.br/products/snapdragon>> Acesso em: 07 ago. 2015.

RASPBERRYPI, The Raspberry Pi Foundation. **Teach, Learn and Make with Raspberry Pi**. 2006. Disponível em: <<https://www.raspberrypi.org/>> Acesso em: 07 ago. 2015.

RICHARDSON, Matt; WALLACE, Shawn. **Primeiros Passos com o Raspberry Pi**. Tradução: Patricia Tieme Maeda. São Paulo: Novatec, 2013. ISBN: 978-85-7522-345-1.

SILVEIRA, Sérgio Amadeu. Inclusão digital, software livre e globalização contra-hegemônica. In: SILVEIRA, Sérgio Amadeu; CASINO, João. (Org.). **Software Livre e Inclusão Digital**. 1 ed. São Paulo: Conrad Editora do Brasil, 2003.

SOARES, Cristiane da Silva; ALVES, Thays de Souza. **Sociedade da informação no Brasil: inclusão digital e a importância do profissional de TI**. 2013. Disponível em: <<http://monografias.brasilecola.com/computacao/sociedadeinformacao-no-brasil-inclusao-digital-a.htm>> Acesso em: 07 ago. 2015.

UPTON, Eben; HALFACREE, Gareth. **Raspberry Pi: Manual do Usuário**. Tradução: Celso Roberto Paschoa. São Paulo: Novatec, 2013. ISBN: 978-85-7522-351-2.