

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
ESCOLA POLITÉCNICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**EDUARDO BORSA
HECTOR LUIS BUCH
VICTOR ROMANO PIZZATO**

**RELATÓRIO FINAL DE PROJETO INTEGRADOR
PROJETO LIGHT SWITCH LAMP**

**CURITIBA
2014**

**EDUARDO BORSA
HECTOR LUIS BUCH
VICTOR ROMANO PIZZATO**

**RELATÓRIO FINAL DE PROJETO INTEGRADOR
PROJETO LIGHT SWITCH LAMP**

Relatório de Projeto apresentado ao Curso de Engenharia de Computação da Pontifícia Universidade Católica do Paraná, como requisito parcial para a disciplina de Resolução de Problemas em Engenharia.

Orientador: Prof. MSc Afonso Ferreira Miguel

**CURITIBA
2014**

AGRADECIMENTOS

Gostaríamos de agradecer a Eduardo Lippmann por nos auxiliar com a plataforma Raspberry Pi e com os problemas elétricos.

Em especial, agradecemos o professor Afonso Ferreira Miguel por sua dedicação na orientação do projeto e acompanhamento nas atividades do cronograma, tanto na construção do projeto como na documentação.

Um agradecimento especial aos familiares e amigos dos integrantes da equipe, principalmente ao Sr. Gustavo Luis Buch, por nos ceder completamente o espaço que contribuiu para o bom andamento do projeto.

RESUMO

A ideia do projeto é aliviar em pequenas escalas o trabalho de tarefas mundanas do dia-a-dia do ambiente domiciliar e evitar o desperdício de energia elétrica. O Projeto de light switch lamp permite que controlemos lâmpadas a partir de dispositivos eletrônicos como computador, celular ou mesmo um tablet. A plataforma raspberry pi foi utilizada pois ela apresenta uma grande gama de componentes eletrônicos desenvolvidos para projetos de home automation, além uma vasta quantidade de linguagens programáveis e de ser compatível com quase todo sistema operacional do mercado. Os componentes eletrônicos do projeto foram dispostos na protoboard para que os dados de controle passem pelo cobblerkit até a switch tail. Este componente de alimentação da lampada apresenta um relé interno que é acionado pelo comando do computador e, caso ocorra algum erro indevido, a saída do aplicativo alertará qual é esse erro. Por fim, após o devido comando, a lâmpada acende.

Palavras-chave: Domiciliar. Raspberry Pi. Lampada.

ABSTRACT

The idea of the project is to alleviate at small scales the work of mundane tasks of day- to-day home environment and avoid electric energy waste. The light switch lamp design allows us to control lights from electronic devices such as computer, mobile phone or even a tablet. The raspberry pi platform was used because it offers a wide range of electronic components developed for home automation projects, plus a vast amount of programmable languages and be compatible with almost any operating system market. Electronic components of the project were arranged in the breadboard so that the control data to pass through the cobblerkit to the tail switch. This power component of the lamp has an internal relay that is triggered by computer command, in case of any improper error, exit the application will prompt what is this error. Finally, after due command, the lamp lights up.

Key words: Home Environment. Raspberry pi. Lamp.

LISTA DE ILUSTRAÇÕES

Figura 1 – Construção eletrônica I, de Marco Schwartz.....	11
Figura 2 – Construção eletrônica II, de Simon Monk.....	12
Figura 3 – Interface do Raspberry Pi por HDMI.....	17
Figura 4 – Diagrama de Software.....	18
Figura 5 – Programa em Python.....	19
Figura 6 – Raspberry Pi e SD Card	19
Figura 7 – Cobbler kit.....	20
Figura 8 – Estrutura eletrônica finalizada.....	20
Figura 9 – Lâmpada de teste desligada.....	21
Figura 10 – Lâmpada de teste ligada	21

LISTA DE ABREVIATURAS E SIGLAS

HDMI	<i>High Definition Multimedia Interface</i>
LSL	<i>LightSwitchLamp</i>
PUCPR	Pontifícia Universidade Católica do Paraná
RAM	<i>Random Access Memory</i>
SD	<i>Secure Digital</i>
SO	Sistema operacional
VNC	<i>Virtual Networking Computing</i>

SUMÁRIO

1	INTRODUÇÃO	8
1.1	HISTÓRICO DO PROJETO	9
1.2	OBJETIVOS	9
1.2.1	Objetivo Geral	9
1.2.2	Objetivos Específicos	9
2	ESTADO DA ARTE	11
3	REFERENCIAL TEÓRICO	13
3.1	SISTEMA OPERACIONAL	13
3.2	PLATAFORMA DE PROGRAMAÇÃO	13
3.3	DISPOSITIVO RASPBERRY PI	14
3.4	COBBLER KIT	14
4	METODOLOGIA	15
4.1	MATERIAIS ELETRÔNICOS	15
4.1	SOFTWARE	16
5	O PROJETO	17
3.1	PROJETO DE SOFTWARE	18
3.1	PROJETO ELETRÔNICO	19
6	RESULTADOS	22
7	IMPACTO AMBIENTAL	23
8	CONSIDERAÇÕES FINAIS	24
	REFERÊNCIAS	25
	ANEXO A – CÓDIGO FONTE DO SOFTWARE	26

1 INTRODUÇÃO

O projeto surgiu com a finalidade de levar os conhecimentos de computação e eletrônica para o ambiente caseiro. A ideia de lâmpadas controladas via internet das coisas é apenas o começo de uma grande quantidade de projetos que podem ser criados na automatização de atividades em uma residência.

O projeto em si consiste em utilizar um cabo ethernet para conectar a placa até a lâmpada e, assim, ligar ou desliga uma lâmpada. Para isso, usa-se a plataforma Raspberry PI, um dispositivo de controle de corrente e uma lâmpada. Isso tudo coordenado por um cobbler kit e jumpers.

1.1 HISTÓRICO DO PROJETO

O aluno Victor Romano apresentou a ideia de desligar e ligar a chave de uma lâmpada sem a necessidade de um interruptor físico comum, assim, a equipe desenvolveu o projeto LSL, que trata de automatização de luzes de uma residência, utilizando uma conexão entre um dispositivo controlador e a lâmpada, visando o controle do usuário através de um software nesse dispositivo.

O projeto visou à economia de energia elétrica na residência onde ele foi implementado, tanto como monitoramento e a praticidade. O projeto oferece ao usuário uma possibilidade de controlar as lâmpadas de sua residência de qualquer cômodo da mesma, reduzindo custos financeiros em situações aparentes de “esquecimento” de lâmpadas acessas em outros cômodos.

Como esperado, houveram problemas na tentativa de conexão wireless e na criação de uma interface para o programa de controle.

1.2 OBJETIVOS

Os objetivos do projeto compõem um destino ao quais as atividades serão realizadas para compri-lo. O destino desse projeto é o de facilitar o monitoramento de lâmpadas na residência, economizando energia elétrica quando possível e facilitando o controle dessas lâmpadas.

1.2.1 Objetivo Geral

O objetivo geral deste projeto consiste em auxiliar na economia de luz elétrica da casa, utilizando um meio remoto como interruptor, a fim de controlar as lâmpadas da residência onde o projeto está instalado.

1.2.2 Objetivos Específicos

Os objetivos específicos do trabalho são:

- a) Identificar um meio de conexão e acessibilidade;
- b) Apresentar uma maneira de acender uma lâmpada remotamente;
- c) Relacionar o raspberry PI com um dispositivo de controle remoto;

- d) Monitorar a atividade das lâmpadas no local que estão instaladas;
- e) Desenvolver um software simples que controle e apresente as lâmpadas instaladas.

2 ESTADO DA ARTE

Dentre os projetos encontrados sobre controlar lâmpadas via internet das coisas, os projetos seguem sempre um mesmo princípio:

- 1) Existe algum dispositivo programável que receberá e enviará algum comando.
- 2) Uma chave de energia, cuja principal componente é um relé.
- 3) Uma lâmpada.

Dois exemplos muito similares, do autor Marcos Swartz, se diferenciaram na plataforma e no código fonte final do software. Um projeto feito com Arduino (<https://www.openhomeautomation.net/arduino-wifi-switch>) e outro com Raspberry Pi presente no e-book *Home Automation with the Raspberry Pi Build Home Automation Systems Using The Power of The Raspberry Pi*. O projeto pode ser feito tanto com a plataforma raspberry, como com a arduino. No projeto, os modelos utilizados são raspberry PI model B ou B+ e arduino zen. A vantagem de utilizar o arduino zen é que ele já apresenta shield wi-fi imbutido.

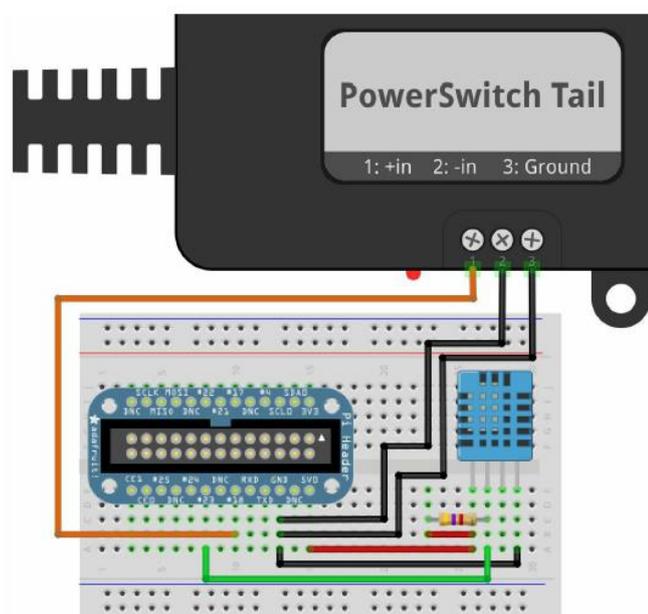


Figura 1 - Construção eletrônica I, de Marco Schwartz

Outro projeto similar aos já citados acima, é do autor Simmons Monk (<https://learn.adafruit.com/downloads/pdf/adafruits-raspberry-pi-lesson-13-power->

control.pdf) , a diferença basicamente ocorre na configuração eletrônica. Um componente singular usado nesse projeto e no LSL é o Power switch tail, que substituiria o relé.

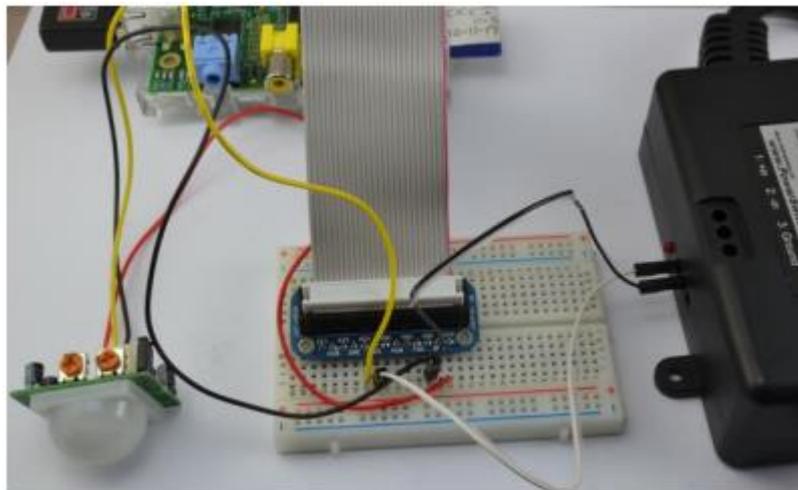


Figura 2 - Construção eletrônica II de Simon Monk.

3 REFERENCIAL TEÓRICO

Para o projeto LSL, foram utilizados diversos conhecimentos que abordaram construção de hardware, software, conexão entre dispositivos, uso do dispositivo Raspberry Pi, entre outros.

3.1 SISTEMA OPERACIONAL

O sistema operacional padrão do Raspberry Pi é baseado em Linux, por ser um sistema de otimização rápida, e ter uma grande gama de drivers e aplicações, ele é fortemente recomendado para ser utilizado com o Raspberry Pi, que é limitado nos aspectos de memória RAM e armazenamento. O sistema operacional tem o nome de Raspbian (Raspberry Pi + Debian), que é integrada como um sistema de código livre.

Outros sistemas operacionais são menos aconselháveis para utilizar com esse microcomputador. Tanto o Windows como o Mac, utilizam e necessitam de mais memória e armazenamento para suas utilidades.

3.2 PLATAFORMA DE PROGRAMAÇÃO

Para a plataforma de programação, uma escolha de boa integração com o Raspberry Pi deveria ser escolhida, para que o software a ser desenvolvido tivesse uma otimização coerente com as funcionalidades propostas.

Dentre as opções de plataformas:

a) Python, uma linguagem de script, padrão para Raspberry Pi, tem rápida edição de código fonte, a leitura é feita de forma simples e eficiente. Um dos contras dessa opção é a de que, projetos complexos que envolvem um alto nível de programação e estrutura dificultam a programação nessa plataforma. Há uma variedade imensa de biblioteca a serem utilizadas, uma delas é a GPIO (General Purpose Input/Output) que pode ser utilizada para fazer a comunicação entre uma lâmpada e o software

b) Java: É uma linguagem mais difícil de ser interpretada, porém os pacotes e bibliotecas integrados ajudam o usuário a criar projetos maiores e mais complexos, a interface de desenvolvimento ajuda na detecção de erros no código fonte mesmo

antes de compilar o programa. Muito utilizado em orientação a objetos. Para projetos simples não é recomendado, pois exige um nível de instrução muito maior e suas linhas de códigos são consideradas pelo mercado de software como muito extensas.

c) C/C++: Suas vantagens e desvantagens são similares com a do Java, sua integração é muito bem feita com Windows, porém seu desenvolvimento detalhado torna essa opção desnecessária para projetos de funcionalidades básicas.

Depois de analisar as opções, a plataforma Python foi escolhida pois sua integração com o Raspberry Pi é ótima e o escopo desse projeto confere um nível não muito alto de programação.

3.3 DISPOSITIVO RASPBERRY PI

O Raspberry Pi é um microcomputador de diversas utilidades. Ele tem a capacidade de fazer praticamente tudo que um computador desktop também faz, como navegar na internet, rodar mídias de áudio e vídeo, jogar alguns jogos, entre outros. Talvez o mais importante desse dispositivos é o de interagir com objetos do mundo real e outros dispositivos, utilizando câmeras, motores, controles remotos, sensores de diversos tipos.

3.4 COBBLER KIT

É um kit composto por pequenas ferramentas físicas que interagem com o Raspberry Pi. A protoboard do kit vem com entradas para diversos componentes eletrônicos que são utilizados em diferentes projetos, a biblioteca GPIO no Raspberry Pi é utilizada para controlar os pinos dessa placa, podendo enviar e analisar dados de entrada e saída, controlando um fluxo de informações de forma a conectar a parte física de um projeto com a parte “imaginária” (software).

4 METODOLOGIA

No início foi realizado um *brainstorm* com os integrantes do grupo para decidir o que seria feito, após todas as ideias anotadas, começamos a selecionar a qual teria um melhor aproveitamento nesse semestre, assim surgiu como escolhemos a automatização de lâmpadas.

Com a ideia do projeto já decidido, pesquisamos os componentes necessários e como realizaríamos o trabalho, logo procuramos onde comprar e os melhores preços para esses componentes.

Já com as peças em mãos, seguimos o manual estudado para a construção do circuito base para *Lightswitch Lamp* que necessitou de ferro de solda, estanho, jumpers e uma protoboard, assim, concluindo uma das metas.

Próximo passo foi a construção do circuito em uma placa, um dos integrantes comprou uma placa para o circuito impresso e nela desenhou seu circuito, finalizando o processo de colocar no ácido, isso feito no laboratório de engenharia de computação dentro da Pontifícia Universidade Católica do Paraná.

Seguindo o tutorial (www.raspberrypi.org), foi possível instalar o sistema operacional *Raspbian* no *SD CARD* para a utilização da linguagem Python, sendo assim, o software necessário para o funcionamento do projeto pode ser desenvolvido.

4.1 MATERIAIS ELETRÔNICOS

- Raspberry pi (modelo B)
- Sensor DHT11 (para umidade e temperatura)
- Resistor de 4k7 ohm
- Powerswitch tail
- Cobbler kit
- Jumpers
- Estanho
- Ferro de solda 40w

4.2 SOFTWARES

- Raspbian (SO do Raspberry Pi)
- Python 3 (linguagem de programação escolhida)
- Windows 7 (para a conexão via VNC)

5 O PROJETO

Com o intuito de conectar a plataforma Raspberry Pi e um tipo de controle remoto, uma maneira de conectar um *notebook* via ethernet foi encontrada. A consequência disso foi de que a parte visual do sistema Raspbian não precisava mais de um dispositivo HDMI para ser visualizada. Além disso, o *mouse* e o teclado do *notebook* poderiam navegar pelo sistema operacional da plataforma facilmente, essa conectividade trouxe mobilidade e sofisticação de utilizar apenas um dispositivo para pelo menos 3 funções diferentes que outros dispositivos fariam.

A programação do *software* foi feita utilizando a linguagem Python (na versão 3.0 em diante), como já especificado antes, o software acabou se tornando simples e pouco complexo diante da problematização apresentada.

As importações de bibliotecas um tanto como incomuns foram necessárias para a realização do projeto na plataforma, bibliotecas para controlar sensores, pinos de entrada e de saída, corrente elétrica, até para criar conexões VNC (utilizada para conectar com o *notebook*). Entretanto, o Raspbian possui uma proficiência em atualizações e instalações de biblioteca e funções em seu sistema, trazendo rapidez e praticidade para diferentes tipos de necessidades.

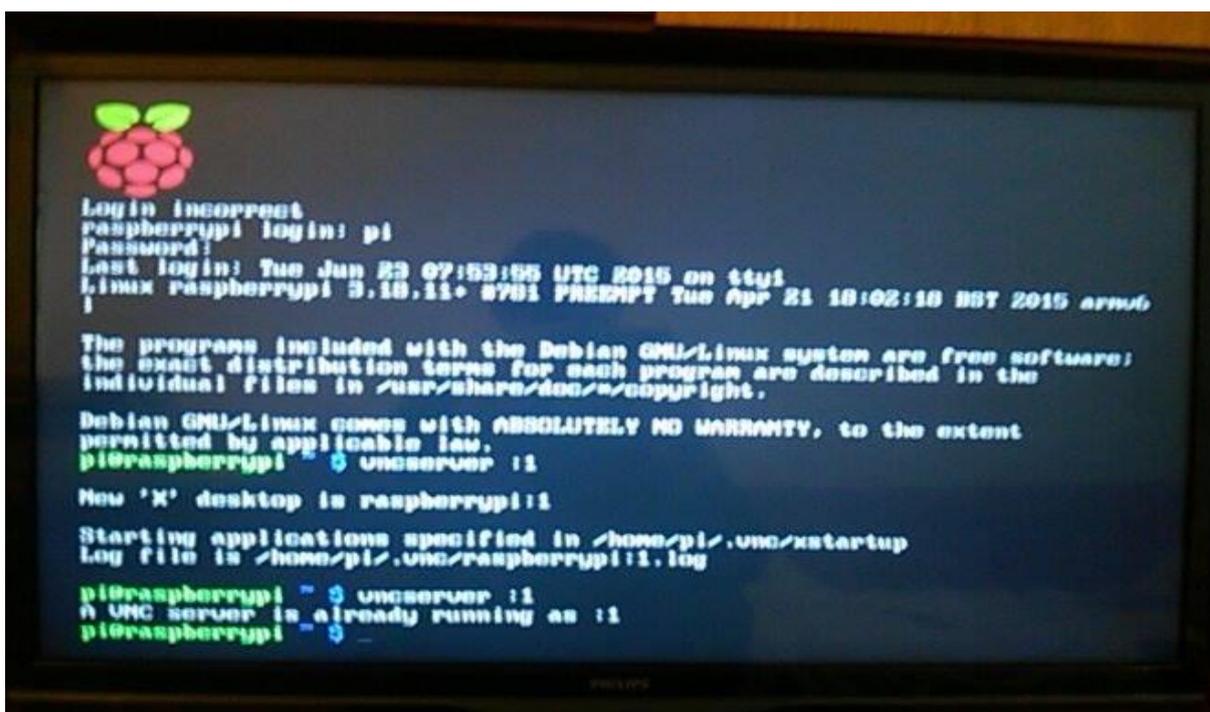


Figura 3 - Interface do Raspberry Pi por HDMI, dos autores

5.1 PROJETO DE SOFTWARE

Utilizando a linguagem Python, o projeto foi adaptado para o mundo virtual, assim trazendo os resultados esperados em ligar remotamente as lâmpadas. No menu do software, há 2 opções diferenciadas, uma entra no menu para lidar com as lâmpadas individuais, a outra para lidar com grupos de lâmpadas. Para ligá-las, é necessário que pelo menos um dos pinos do cobbler kit esteja conectado com um relé e a lâmpada.

No caso desse projeto, foi utilizado o pino número 18 e um Power Tail Switch como relé. Nos 2 menus seguintes, é possível inserir um novo item indicando qual pino ele utilizará para ser ligado e um nome fantasia qualquer. Um código é gerado para esse item, e conseqüentemente será utilizado como índice para ligar e desligar os pinos que conectam até as lâmpadas. Há uma opção para checar quais as lâmpadas instaladas e em qual estado se encontram (ligada/desligada). Na opção status então o usuário pode mudar o estado atual do pino para o oposto, assim controlando as lâmpadas.

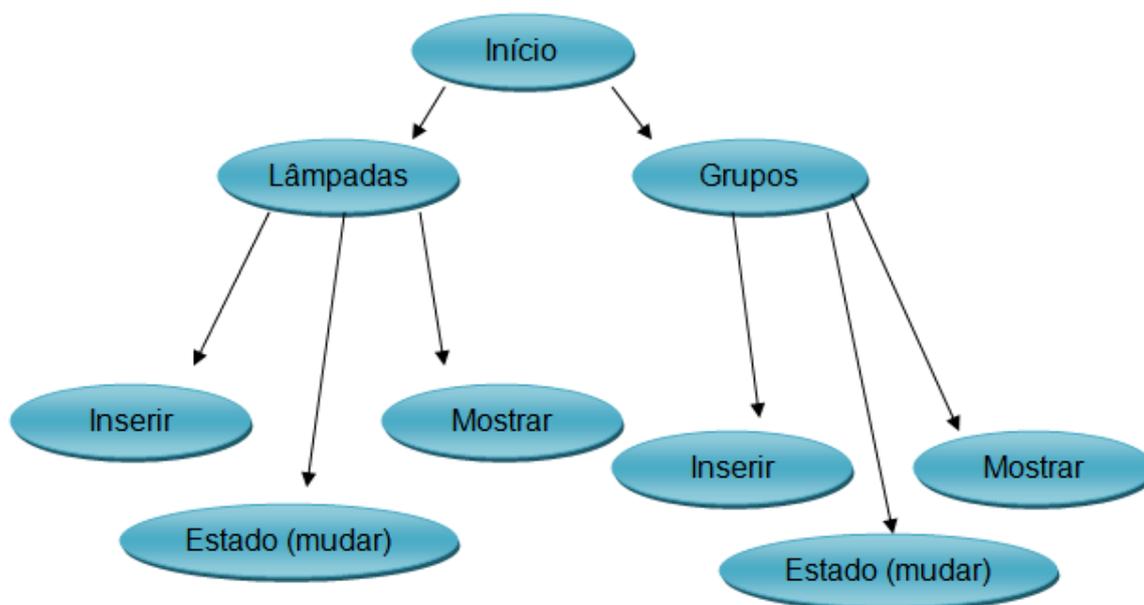
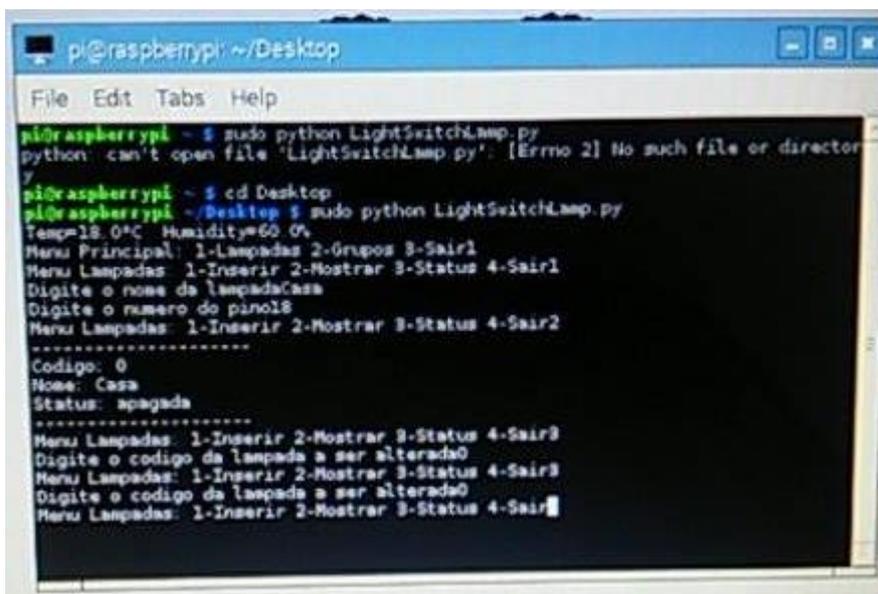


Figura 4 - Diagrama de Software, dos autores



```
pi@raspberrypi: ~/Desktop
File Edit Tabs Help
pi@raspberrypi ~$ sudo python LightSwitchLamp.py
python: can't open file 'LightSwitchLamp.py': [Errno 2] No such file or directory
pi@raspberrypi ~$ cd Desktop
pi@raspberrypi ~/Desktop$ sudo python LightSwitchLamp.py
Temp=18.0°C Humidity=60.0%
Menu Principal: 1-Lampadas 2-Grupos 3-Sair1
Menu Lampadas: 1-Inserir 2-Mostrar 3-Status 4-Sair1
Digite o nome da lampadaCasa
Digite o numero do pino18
Menu Lampadas: 1-Inserir 2-Mostrar 3-Status 4-Sair2
-----
Codigo: 0
Nome: Casa
Status: apagada
-----
Menu Lampadas: 1-Inserir 2-Mostrar 3-Status 4-Sair3
Digite o codigo da lampada a ser alterada0
Menu Lampadas: 1-Inserir 2-Mostrar 3-Status 4-Sair3
Digite o codigo da lampada a ser alterada0
Menu Lampadas: 1-Inserir 2-Mostrar 3-Status 4-Sair3
```

Figura 5 - Programa em Python, dos autores

5.2 PROJETO ELETRÔNICO

A montagem eletrônica foi similar das que foram pesquisadas e vistas em outros projetos. Essa montagem tem como função ativar o sensor DHT11, para fornecer a temperatura e umidade ambiente para propósitos de superaquecimento ou situações climáticas que danifiquem os componentes eletrônicos. O mais importante é a função do Power Switch Tail, que envia o pulso elétrico necessário para a lâmpada quando o pino correspondente for ativado.



Figura 6 - Raspberry Pi e SD Card, dos autores

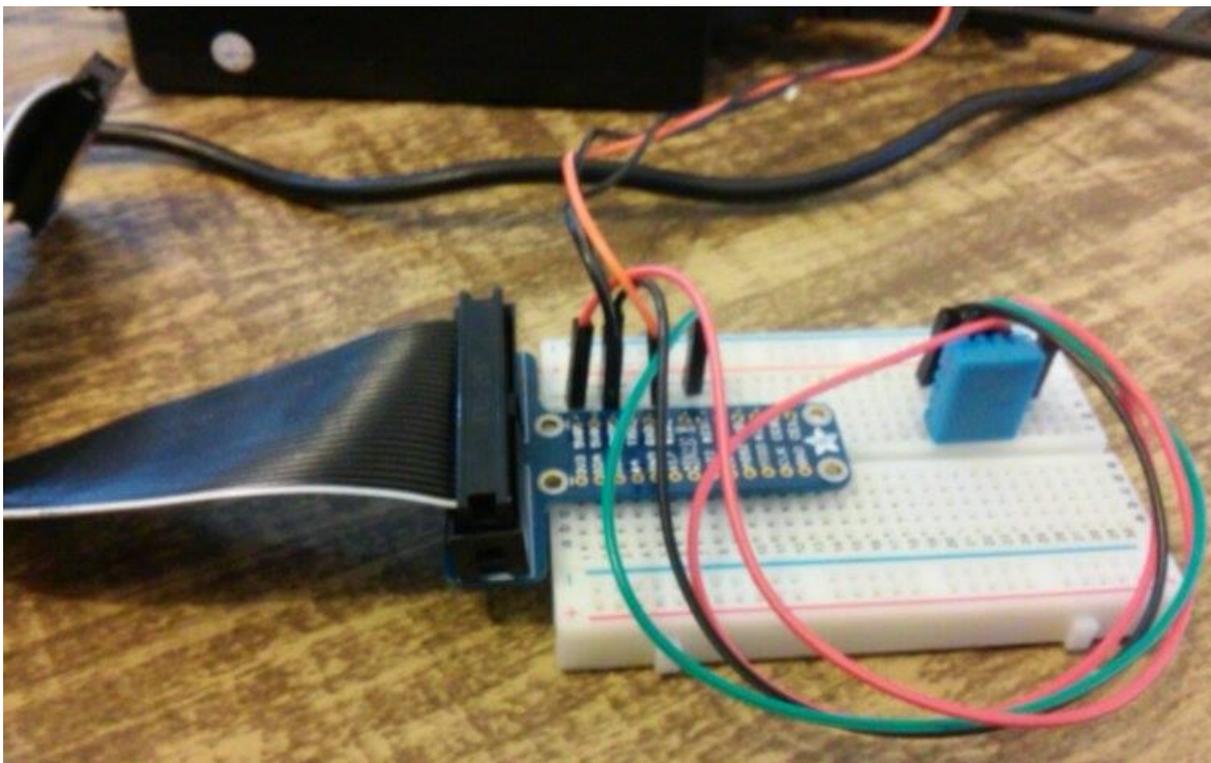


Figura 7 - Cobbler Kit, dos autores

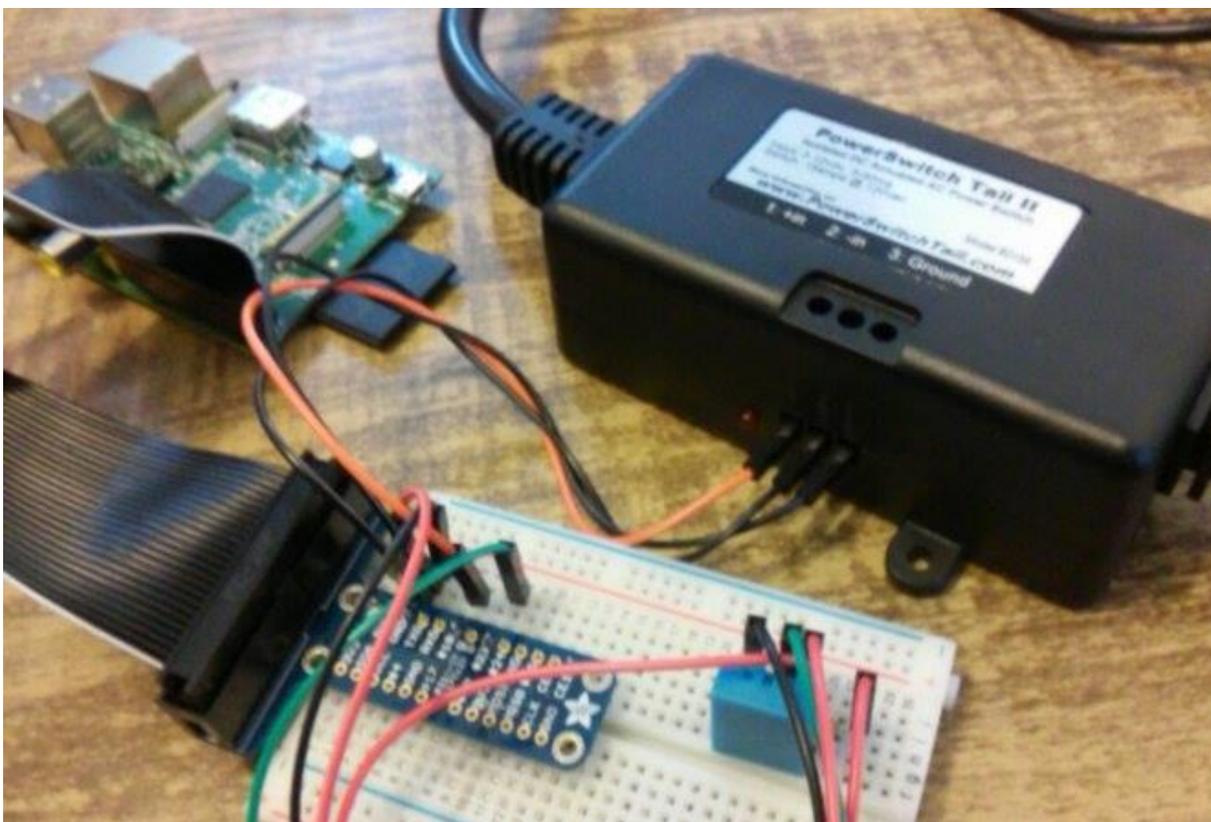


Figura 8 - Estrutura eletrônica finalizada, dos autores

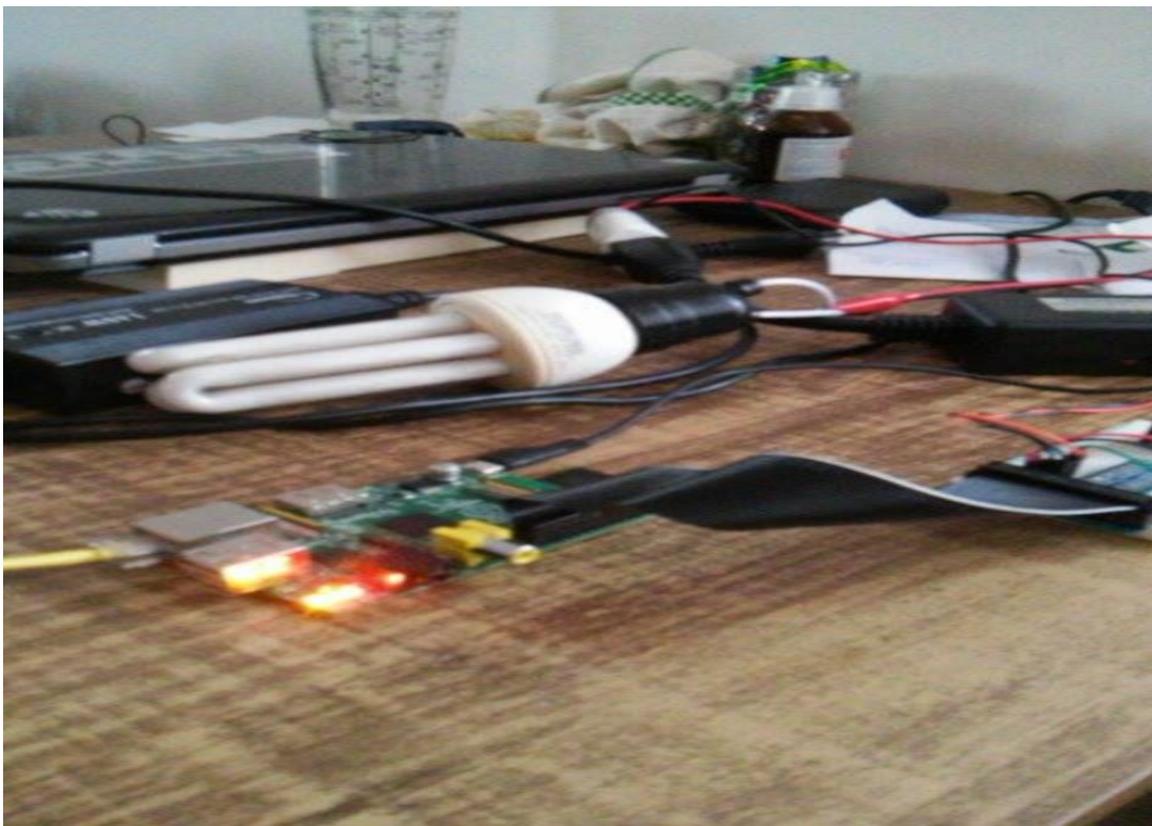


Figura 9 - Lâmpada de teste desligada, dos autores

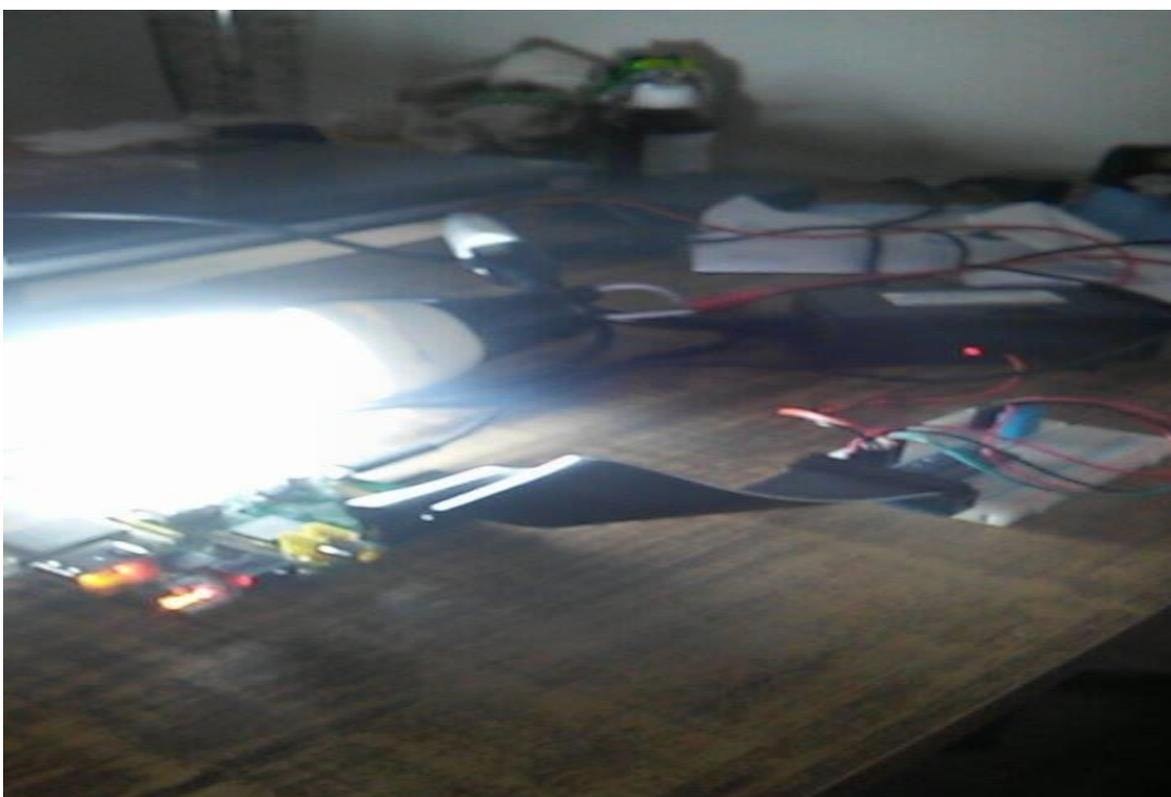


Figura 10 - Lâmpada de teste ligada, dos autores

6 RESULTADOS

Com a ideia de automatização de lâmpadas, começamos a pesquisar qual seria o modo de concretizar o projeto, assim, adquirimos o *raspberry pi b* pois no resultado de nossa pesquisa apresentou com o melhor desempenho e praticidade. Após a aquisição de todos os componentes eletrônicos pelo sítio oficial da empresa Adafruit, porque apresentou um ótimo custo benefício na compra, tanto com tempo de entrega e preço dos componentes, foram realizados testes que resultaram no funcionamento do circuito base para o projeto.

Conectando todos os itens, instalando o software no *raspberry pi b* e passando o código em python, o resultado foi a disponibilidade de acender e apagar uma lâmpada através de um interruptor digital que nós escolhemos, podendo ser trocado, ou seja, fica a escolha da pessoa com quais botões digitais desliga e liga.

Apesar de instalar apenas uma lâmpada o processo de adição é consideravelmente igual a de apenas um, assim, conseguimos realizar a ideia principal do projeto, que é um interruptor de algumas luzes da casa através do software, porém faltando o funcionamento em larga escala.

7 IMPACTO AMBIENTAL

O projeto busca melhor administrar como será gasta a energia elétrica para acender lâmpadas. A ideia de smart lamp permite a checagem remota de uma lâmpada ligada desnecessariamente em determinado momento, podendo desligá-la também remotamente. O projeto é 100% reciclável, sendo que o material mais poluente é a lâmpada. Para esse material, entretanto, é necessário um processo diferente para sua reutilização devido aos componentes químicos presentes no interior da parte de vidro.

Processos realizados em locais especializados são responsáveis por retirar o mercúrio das lâmpadas fluorescentes, assim elimina-se a possibilidade de contaminações ambientais e intoxicações. O mesmo vale para outros aparelhos eletrônicos. Até por isso, o descarte deve ser bem feito, procurando quais os lugares certos, isolando o material em caso de quebra e avisando sobre o conteúdo entregue. Em Curitiba, por exemplo, a Leroy Merlin realiza esse processo.

8 CONSIDERAÇÕES FINAIS

A proposta do projeto LightSwitch Lamp foi de controlar e monitorar lâmpadas de uma maneira simples. No decorrer dos procedimentos para construir esse projeto, foi possível através de uma conexão ethernet utilizando VNC (virtual network computing) conectar o raspberry PI com um notebook, podendo então comandar o raspberry PI utilizando o teclado do notebook e visualizando a interface do Raspbian (sistema operacional do raspberry PI) na tela. Utilizando o ponto de wireless do notebook, foi possível conectar um smartphone com sistema Android, e utilizar dele para controlar as lâmpadas.

Para um projeto futuro que envolva o mesmo escopo do LightSwitch Lamp, seria de interesse utilizar um shield wireless ou um adaptador USB wireless para conectar o raspberry PI e um dispositivo wireless para acessar o software de controle das lâmpadas.

Apenas uma lâmpada foi testada no projeto, essa lâmpada foi indexada a um dos pinos da placa GPIO (general purpose input/output), assim o software pode controlar esse pino enviando um pulso para ligar ou desligar a lâmpada. A função grupos do software, não foi bem implementada por não controlar mais de uma lâmpada por vez, pois havia só uma lâmpada. No próximo projeto, é de interesse conectar mais lâmpadas na placa, para então ver na prática o controle remoto simultâneo de todas.

REFERÊNCIAS

RICHARDSON, Matt; WALLACE, Shawn. Primeiros passos com o Raspberry Pi. Primeira Edição. Novatec Editora Ltda, 2013.

Raspbian. Disponível em: <<http://www.raspbian.org>>. Acesso em: 30 abr. 2015.

Comparando Python com outras linguagens. Disponível em: <<https://www.python.org/doc/essays/comparisons>>. Acesso em: 6 mai. 2015.

Raspberry Pi. Disponível em: <www.raspberrypi.org>. Acesso em: 30 abr. 2015.

PATEL, Sohil. Connect raspberry PI to laptop display. Disponível em: <<http://diyhacking.com/connect-raspberry-pi-to-laptop-display>>. Acesso em: 8 mai. 2015

SCHWARTZ, Marco. Arduino Wifi Power Switch. Disponível em <<https://www.openhomeautomation.net/arduino-wifi-switch>>. Acesso em: 10 mar. 2015.

MONK, Simon. Power Control, Adafruit's raspberry lesson 13. Disponível em: <<https://learn.adafruit.com/downloads/pdf/adafruits-raspberry-pi-lesson-13-power-control.pdf>>. Acesso em: 10 mar. 2015.

SCHWARTZ, Marco. Home automation with Raspberry Pi. Disponível em: <<https://www.openhomeautomation.net/home-automation-raspberry-pi>>. Acesso em: 10 mar. 2015.

ANEXO A – CÓDIGO FONTE DO SOFTWARE

```
import Rpi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)

import Adafruit_DHT
sensor= Adafruit_DHT.DHT11
pin=23
humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)

pinos=[]
menu_l=0
menu_g=0
cod_g=[]
lampadas=[]
status_l=[]
grupos=[]
status_g=[]
nome=""
pino=""
mudar=0
cod_l=[]
codigo=0
m=0
n=0
i=0
j=0
sair_m = False
sair_l = False
sair_g = False
pino_g=[]

while not sair_m:
```

```

if humidity is not None and temperature is not None:
    print 'Temp={0:0.1f}*C Humidity={0:0.1f}%'.format(temperature, humidity)
else:
    print 'Falha em medir a temperatura e umidade ambiente'

menu = input("Menu Principal: 1-Lampadas 2-Grupos 3-Sair")
if (menu==1):
    while not sair_l:
        menu_l=input("Menu Lampadas: 1-Inserir 2-Mostrar 3-Status 4-Sair")
        if (menu_l==1):
            nome=raw_input("Digite o nome da lampada")
            pino=input("Digite o numero do pino")
            cod_l.append(n)
            n=n+1
            lampadas.append(nome)
            status_l.append("apagada")
            pinos.append(pino)
            GPIO.setup(pino, GPIO.OUT)
            GPIO.output(pino, False)
        if (menu_l==2):
            for i in range(n):
                print("-----")
                print ("Codigo: %d" % cod_l[i])
                print ("Nome: " + lampadas[i])
                print ("Status: " + status_l[i])
                print("-----")
        if (menu_l==3):
            mudar=input("Digite o codigo da lampada a ser alterada")
            if (status_l[mudar]=="apagada"):
                GPIO.output(pinos[mudar], True)
                status_l[mudar]="acesa"
            else:
                GPIO.output(pinos[mudar], False)

```

```

        status_l[mudar]="apagada"
    if (menu_l==4):
        sair_l=True

if(menu==2):
    while not sair_g:
        menu_g=input("Menu Grupos: 1-Inserir 2-Mostrar 3-Status 4-Sair")
    if (menu_g==1):
        qnt=raw_input("Informe a quantidade de pinos")
        for i in range(qnt-1):
            codigo=input("Insira um pino a ser colocado no grupo")
            pinos_g.append(codigo)
            m=m+1
        nome=raw_input("Digite o nome do grupo")
        grupos.append(nome)
        cod_g.append(m)
        status_g.append("apagado")

    if (menu_g==2):
        for i in range((cod_g.lenght)-1):
            print("-----")
            print ("Codigo: %d" % cod_g[i])
            print ("Nome: " + grupos[i])
            print ("Status: "+ status_g[i])
        print("-----")
    if (menu_g==3):
        mudar=input("Digite o codigo do grupo a ser alterado")
        for i in range(((cod.lenght)-1):
            if codigo[i]==mudar:
                if status_g[i]=="apagado":
                    status_g[i]=="acesso"
                for j in range((codigo[i])-(codigo[i-1])-1):
                    GPIO.output(pinos[(codigo[i-1])+j], True)

```

```
    else:
        status_g[i]="apagado"
        for j in range((codigo[i])-(codigo[i-1])-(1)):
            GPIO.output(pinos[(codigo[i-1])+j], False)
    if (menu_g==4):
        sair_g=True
if(menu==3):
    sair_m= True
```