

## **DOMÓTICA: DESENVOLVIMENTO DE PROTÓTIPO PARA AUTOMAÇÃO RESIDENCIAL COM SMARTPHONE ANDROID E RASPBERRY PI**

**João Pedro Fiorin<sup>1</sup>, Ricardo Antonello<sup>2</sup>**

<sup>1</sup>Instituto Federal Catarinense, Campus Luzerna / joaop.fiorin88@gmail.com

<sup>2</sup>Instituto Federal Catarinense, Campus Luzerna / ricardo.antonello@ifc.edu.br

*O projeto a seguir demonstra um sistema capaz de controlar dispositivos de uma residência como lâmpadas, televisores, ventiladores, cafeteiras, alarmes, cortinas e persianas através de um smartphone. O sistema utiliza um minicomputador Raspberry Pi 3 programado com um algoritmo em linguagem Python para receber os comandos do smartphone e acionar os dispositivos. Os conceitos aplicados neste projeto para automação residencial fazem parte do que recentemente foi denominado domótica. A domótica envolve a aplicação de automação e robótica dentro de residências e domicílios facilitando algumas tarefas e melhorando a acessibilidade do lar. O acionamento das funções do sistema foi feito por smartphones Android utilizando software livre. Além disso, é possível acionar as funções do sistema por qualquer computador conectado à internet. Como resultado, percebemos que viável a utilização de smartphones com sistema Android para acionar via rede intranet ou internet dispositivos conectados com um Raspberry Pi 3.*

*Palavras-Chave: Domótica, Automação Residencial, Raspberry Pi.*

### **1. INTRODUÇÃO**

Este projeto foi desenvolvido com objetivo de mostrar um sistema que envolvesse a utilização da área de domótica que se refere a robótica domiciliar (DOMÓTICA, 2017) para tentar desenvolver um ambiente onde o operador pode controlar os dispositivos de sua residência através de seu computador ou smartphone.

Com os conhecimentos da linguagem de programação Python (2017) e utilizando o minicomputador Raspberry Pi 3 (2017), foi montado o esquema elétrico de uma sala de estar afim de aplicar o método, mostrando-se funcional.

### **2. RECURSOS UTILIZADOS**

#### **2.1. Placa Arduino**

O Arduino (2017) foi criado em 2005 com objetivo de elaborar um dispositivo que fosse ao mesmo tempo barato, funcional e fácil de programar, sendo dessa forma acessível a estudantes e projetistas amadores.

Composto por um microcontrolador Atmel, a placa possui circuitos de entrada/saída e que podem ser facilmente conectados a um computador e programados via IDE (Integrated

Development Environment, ou Ambiente de Desenvolvimento Integrado) utilizando uma linguagem baseada em C/C++ (2017), sem a necessidade de equipamentos extras além de um cabo USB.

Figura 1 – Microcontrolador Arduino UNO.



Fonte: (ARDUINO, 2017).

## 2.2. Placa Raspberry Pi

Raspberry Pi (2017) é um minicomputador do tamanho de um cartão de crédito. É composto por um processador, processador gráfico, espaços para cartões de memória, interface USB, HDMI e seus respectivos controladores, memória RAM, entrada de energia e barramentos de expansão e pode ser visto como um computador completo. No topo da figura 2 temos 40 pinos que formam a GPIO<sup>1</sup> do Raspberry Pi, uma interface para acionamento de dispositivos em 3.3 volts.

Figura 2 – Minicomputador Raspberry Pi.



Fonte: (ELEMENT14, 2017).

<sup>1</sup> GPIO: General purpose input/output ou Interface de entrada e saída de propósito geral (tradução dos autores).

### 2.3. Módulos Relé

O módulo relé funciona como uma chave (interruptor) de três conexões: NA (Normalmente Aberto), C (Comum) e NF (Normalmente Fechado). Ou seja, quando o relé receber sinal VCC na conexão C, a conexão Comum estará ligada a NA, caso contrario estará ligada a NF.

Figura 3 – Módulo Relé de 2 canais.



Fonte: (FILIPEFLOP, 2017).

### 2.4. Android

Android (2017) é um sistema operacional baseado em linux e atualmente desenvolvido pela empresa de tecnologia Google. Com uma interface de usuário baseada na manipulação direta, o Android é projetado principalmente para dispositivos móveis como smartphones e tablets.

O Android é muito popular entre os que buscam um software pronto, de baixo custo e personalizável para dispositivos diversos. O software de código aberto do sistema operacional tem encorajado um grande numero de programadores a colocar uma fundação para o desenvolvimento de projetos feitos pela própria autoria que adicionam recursos para usuários mais avançados.

Figura 4 – Símbolo do Android.



Fonte: (CNET, 2017)

## 2.5. Linguagem Python

Python (2017) é uma linguagem de alto nível, interpretada, de script, imperativa e orientada a objetos. Possui tipagem dinâmica e uma de suas principais características é permitir a fácil leitura e exigir poucas linhas de código se comparado ao mesmo programa em outras linguagens. Devido às suas características, ela é principalmente utilizada para processamento de textos, dados científicos e criação de CGIs para páginas dinâmicas para a web.

## 2.6. Display LCD TFT

O display LCD TFT Touch foi desenvolvido para dar ao Raspberry Pi mais mobilidade. Com dimensões de 3,5 polegadas com resolução de 320x480 pixels, o display pode ser encaixado diretamente no Raspberry Pi com um conector de 26 pinos, e deixar os demais pinos do Raspberry livres para outras funções.

Esse display serve como um monitor portátil que pode realizar, sozinho as funções do monitor, do mouse e do teclado, liberando as portas USB do Raspberry para outros fins.

Figura 5 – Display LCD TFT Touch.



Fonte: (SAINSMART, 2017)

## 2.7. Diferenças entre a placa Arduino e Raspberry Pi 3

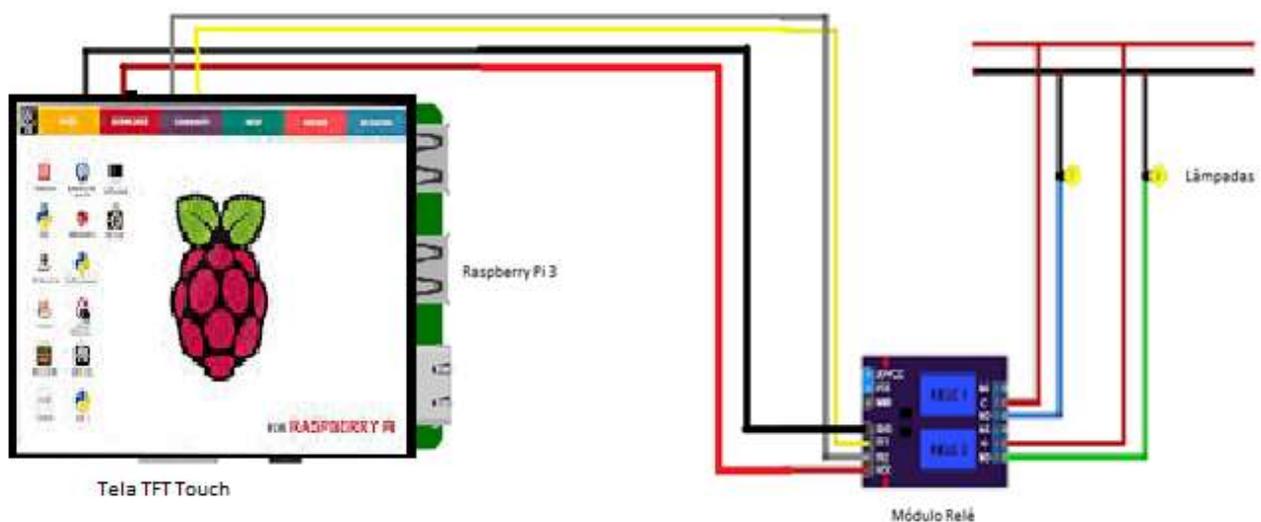
Na realização deste projeto, foi preciso escolher um ponto de partida para começar o trabalho, para isto, teve-se que optar pelo uso de uma das duas maneiras de controle: Utilização do Raspberry Pi 3 (2017) ou do Arduino (2017).

A diferença entre os dois foi analisada em detalhes por Sueiro (2017). Como se sabe, o Arduino nada mais é do que um microcontrolador construído para que pessoas com pouco conhecimento sejam capazes de utilizá-lo, pode ser conectado com o computador e trabalhar com este em total sincronia, sendo que sua utilização é simples e bastante facilitada. De contra partida, o Raspberry Pi não faz parte da família dos microcontroladores, mas sim da família dos microprocessadores. Ao invés de funcionar em conjunto com o computador domiciliar, esta placa pode ser considerada um computador próprio. Utilizando o sistema operacional Linux (2017), pode ser conectada diretamente ao monitor, a rede de internet e a fonte de energia funcionando como um computador. Sua velocidade de operação e processamento variam e podem ser bem mais rápidas se comparadas ao Arduino.

### 3. METODOLOGIA

Para este trabalho, o hardware escolhido foi o minicomputador Raspberry Pi 3 como centro de comando. O Raspberry Pi 3 possui um ambiente de trabalho Linux e é capaz de realizar cálculos mais complexos do que o Arduino. Além disso, o Raspberry Pi 3 possui conexão Bluetooth (2017) e Wifi (2017) já integrados, além de possuir conexão HDMI (2017) e conexão serial para telas multi-toque. Outra vantagem é ser multi-tarefa permitindo o envio e recebimento de comandos de vários dispositivos ao mesmo tempo.

Figura 6 – Esquema elétrico.



Fonte: Os autores.

Inicialmente foi realizada a montagem da parte elétrica do sistema. Para conexão a rede elétrica com tensão de 220 volts foi utilizado o Módulo Relé. Este módulo foi conectado à interface GPIO do Raspberry com tensão de 3.3 volts para comandar o acionamento. Um sistema de tomadas conjuntas foi conectado ao Módulo Relé para o acionamento em 220 volts e conexão com a rede elétrica da residência. O esquema elétrico montado pode ser visto na figura 5 abaixo:

### **3.1. Configuração do Display LCD**

Segue a baixo um tutorial de como configurar o Display LCD TFT Touch 3,5'' para o Raspberry Pi 3:

Primeiro, é preciso configurar o Raspberry da seguinte maneira:

- No terminal digite: `sudo raspi-config`
- Vá até Advanced Options
- Habilite a opção SPI
- Habilite a opção Resolution para 640x480 60Hz
- De volta ao terminal, reinicie o Raspbian digitando: `sudo reboot`

Após a reinicialização, plugue o display na placa Raspberry Pi, abra o terminal e execute os seguintes comandos:

- `Sudo apt-get update`
- `Sudo apt-get upgrade`

Após este comando o raspberry começará a atualizar suas funções e isso pode demorar algum tempo. Em seguida, ainda no terminal digite o comando para baixar os drivers necessários para o raspberry:

- `wget http://www.waveshare.com/w/upload/9/9d/LCD-show-151020.tar.gz`

Agora é preciso descompactar os arquivos baixados utilizando o comando:

- `tar xvf LCD-show-151020.tar.gz`

Para ir até o diretório digite:

- `cd LCD-show`

E por fim, basta executar o driver:

- `sudo ./LCD35-show`

### 3.2. Procedimento

Na interface de programação do Raspberry Pi foi elaborado um algoritmo em Python para realizar o controle do Módulo Relé via ligação em 3.3 volts com a GPIO do Raspberry Pi.

O algoritmo abaixo na Tabela 1 permite a ativação do Módulo Relé a partir de duas interfaces, quais sejam:

- Botões virtuais na tela multi-toque do Raspberry Pi 3;
- Ativação remota pela internet via outro computador ou smartphone utilizando um aplicativo Android.

Os comentários nas linhas da Tabela 1 são precedidos do caractere '#’.

**Tabela 1** – Código de ativação dos Módulos Relés e acionamento das luzes e eletrodomésticos.

Linha	Código
1	tempo = 2 #define variável com o valor inteiro 2
2	import RPi.GPIO as GPIO #Define biblioteca da GPIO
3	import time #Define biblioteca de cálculo do tempo
4	GPIO.setmode(GPIO.BOARD)
5	#Define os pinos 12, 13, 14 e 15 da placa como saída
6	GPIO.setup(12, GPIO.OUT)
7	GPIO.setup(13, GPIO.OUT)
8	GPIO.setup(14, GPIO.OUT)
9	GPIO.setup(15, GPIO.OUT)
10	
11	#rotina para acender a lâmpada 1
12	def acende(pino):
13	GPIO.output(pino, 1)
14	Return
15	
16	#rotina para acender a lâmpada 2
17	def acende(pino):
18	GPIO.output(pino, 1)
19	return
20	
21	#rotina para acender o ventilador
22	def acende(pino):
23	GPIO.output(pino, 1)
24	return
25	
26	#rotina para acender a cafeteira
27	def acende(pino):
28	GPIO.output(pino, 1)
29	return
30	
31	#rotina para apagar a lâmpada 1
32	def apaga(pino):
33	GPIO.output(pino, 0)
34	Return
35	#rotina para apagar a lâmpada 2
36	def apaga(pino):

```

37     GPIO.output(pino, 0)
38     return
39     #rotina para apagar o ventilador
40     def apaga(pino):
41         GPIO.output(pino, 0)
42         return
43     #rotina para apagar a cafeteira
44     def apaga(pino):
45         GPIO.output(pino, 0)
46         return
  
```

Fonte: Os autores.

Com o algoritmo das rotinas acima, é possível adaptá-los para a ativação dos componentes da residência, seja pela utilização de botões ou por controle de tempo.

**Tabela 2** – Código para criar a interface com botões para ativar os Módulos Relé.

Linha	Código
1	def __init__(self, master=None):
2	#cria o widget da primeira coluna
3	self.fontePadrao = ("Arial", "10") #define o tipo e
4	o tamanho da letra
5	self.primeirocontainer = Frame(master)
6	self.primeirocontainer["pady"] = #define as
7	dimensões do eixo y
8	self.primeirocontainer.pack() #imprime
9	#cria o titulo da primeira coluna
10	self.titulo = Label(self.primeirocontainer,
11	text="Estado da Lâmpada")
12	self.titulo["font"] = ("Arial", "10", "bold")
13	#define o tipo e o tamanho da letra
14	self.titulo["width"] = 17 # define a largura do
15	titulo
16	self.titulo.pack()
17	#cria o widget da segunda coluna
18	self.fontePadrao = ("Arial", "10") #define o tipo e
19	o tamanho da letra
20	self.segundocontainer = Frame(master)
21	self.segundocontainer["pady"] = 10 #define as
22	dimensões do eixo y
23	self.segundocontainer.pack()#imprime
24	#cria o botão da segunda coluna
25	self.botao = Button(self.primeirocontainer)
26	self.botao ["text"] = "Lâmpada Desligada" #digita
27	‘Lâmpada Desligada’ no botão
28	self.botao ["font"] = ("Calibri", "10") #define o
29	tipo e o tamanho da letra
30	self.botao ["width"] = 15 # define a largura do
31	botão
32	self.botao ["background"] = "red" #define a cor do
33	botão
34	

```
35         self.botao.bind("<Button-1>", self.mudartexto)
36     #seleciona quando o botão é apertado
37         self.botao.pack()
38     def mudartexto(self, event): #evento em que o botão é
39     precionado
40         if self.botao ["text"] == "Lâmpada Ligada":
41             self.botao ["text"] = "Lâmpada Desligada"
42             self.botao ["background"] = "red" #muda a cor
43     do botão para vermelho
44         else:
45             self.botao ["text"] = "Lâmpada Ligada"
46             self.botao ["background"] = "green" #muda a cor
47     do botão para verde
48
49     root = Tk()
50     Frame(root)
51     root.title("janela de testes") #titulo da interface
52     Application(root)
53     root.mainloop() #loop para a tela ficar ativa
54
55
56
```

Fonte: Os autores.

O algoritmo acima tem como função criar a interface gráfica com botões virtuais para ser utilizada na tela multi-toque LCD do Raspberry e assim poder controlar o sistema.

#### **4. RESULTADOS E DISCUSSÃO**

Ao final, colocando o projeto em prática, pudemos fazer o uso deste sistema doméstico facilitando em muito as atividades domésticas.

O sistema, apesar de simples, mostrou-se funcional e pratico, onde todos os objetivos propostos foram concluídos com êxito, dando a possibilidade de uma pessoa com restrição de conhecimentos na área poder controlar sua residência remotamente pelo seu smartphone, assim abrindo uma nova janela para o uso da tecnologia residencial.

#### **5. CONSIDERAÇÕES FINAIS**

Ao utilizarmos a área de automação em nosso sistema e realizar uma breve consulta sobre domótica, pudemos observar em que pé estamos na nossa atualidade e oque ainda esta por vir. Foi possível demonstrar que algo de simples montagem e manutenção pode facilitar a vida de muitas pessoas dentro de seus lares.

## AGRADECIMENTOS

Agradecemos ao Instituto Federal Catarinense pela oportunidade em nos oferecer a possibilidade de realizar o projeto de pesquisa do edital 162/16 em conjunto com o curso de Engenharia de Controle e Automação nos dando possibilidade para realizar este projeto com sucesso.

## REFERÊNCIAS BIBLIOGRÁFICAS

ANDROID. **O Sistema Operacional móvel mais conhecido do mundo.** Disponível em:

<<https://www.android.com/>>. Acesso em: 20 de jul. 2017.

ARDUINO. **O que é Arduino.** Disponível em: <<http://blog.filipeflop.com/>>. Acesso em: 11 de jul. 2017.

BLUETOOTH. **Como funciona o Bluetooth.** Disponível em:

<<https://www.tecmundo.com.br/bluetooth/161-o-que-e-bluetooth-.htm>>. Acesso em: 20 de jul. 2017.

CNET. **Android Update.** Disponível em: <<https://www.cnet.com/android-update/>>. Acesso em: 20 de jul. 2017.

DISPLAY LCD. **Image.** Disponível em: <<https://www.sainsmart.com/sainsmart-3-5-inch-tft-lcd-320-480-touch-screen-display-for-raspberry-pi-2-b-b.html>>. Acesso em: 20 de jul. 2017.

DISPLAY LCD. **Raspberry pi.** Disponível em: <<http://www.filipeflop.com/pd-24ca21-display-lcd-tft-touch-3-5-raspberry-pi.html>>. Acesso em: 20 de jul. 2017.

DOMÓTICA. **Interação natural na domótica.** Disponível em:

<<https://prezi.com/2kym0rjkmw1x/interacao-natural-na-domotica/>>. Acesso em: 11 de jul. 2017.

HDMI. **O que é HDMI?.** Disponível em: <<https://canaltech.com.br/produtos/O-que-e-HDMI/>>.

Acesso em: 20 de jul. 2017.

MÓDULO RELÉ. **Arduino.** Disponível em: <<https://www.robocore.net/>>. Acesso em: 10 de jul. 2017.

RASPBERRY PI. **O que é Raspberry Pi.** Disponível em: <<http://www.raspberrypi.org>>. Acesso em: 10 de jul. 2017.

SUEIRO, Diego. **Arduino vs Raspberry: Entenda as diferenças.** Disponível em:

<<https://www.embarcados.com.br/arduino-vs-raspberry-pi>>. Acesso em: 11 de jul. 2017.