

**UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

ALESSANDRA CARINE MARY SASAKI

**DESENVOLVIMENTO DE UM SISTEMA DE SEGURANÇA COM BASE
EM INTERNET DAS COISAS UTILIZANDO RASPBERRY PI E
SENSORES DE PRESENÇA**

BAURU - SP

2017

ALESSANDRA CARINE MARY SASAKI

**DESENVOLVIMENTO DE UM SISTEMA DE SEGURANÇA COM BASE
EM INTERNET DAS COISAS UTILIZANDO RASPBERRY PI E
SENSORES DE PRESENÇA**

Trabalho de Conclusão de Curso do Curso de Bacharelado em Ciência da Computação da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Faculdade de Ciências, campus Bauru.

Orientador: Prof. Dr. Eduardo Martins Morgado

BAURU
2017

Sasaki, Alessandra Carine Mary.
Desenvolvimento de um Sistema de Segurança
com base em Internet das Coisas utilizando
Raspberry Pi e Sensores de Presença. /
Alessandra Carine Mary Sasaki, 2017
55 f. : il.

Orientador: Eduardo Martins Morgado

Monografia (Graduação)-Universidade Estadual
Paulista. Faculdade de Ciências, Bauru, 2017

1. Sistema de Segurança. 2. Internet das Coisas.
3. *Raspberry Pi*. 4. Sensores de Presença.

ALESSANDRA CARINE MARY SASAKI

**DESENVOLVIMENTO DE UM SISTEMA DE SEGURANÇA COM BASE
EM INTERNET DAS COISAS UTILIZANDO RASPBERRY PI E
SENSORES DE PRESENÇA**

Trabalho de Conclusão de Curso do Curso de Bacharelado em Ciência da Computação da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Faculdade de Ciências, campus Bauru.

Aprovação: 13/02/2017.

Banca Examinadora

Prof. Dr. Eduardo Martins Morgado
Orientador
Faculdade de Ciências
Universidade Estadual Paulista - Bauru

Profa. Dra. Simone das Graças Domingues Prado
Faculdade de Ciências
Universidade Estadual Paulista - Bauru

Profa. Dra. Roberta Spolon
Faculdade de Ciências
Universidade Estadual Paulista - Bauru

Dedico este Trabalho de Conclusão de Curso em memória de Chiya Yoshida Baba (1923 - 2015), que partiu deixando um grande exemplo de ética, perseverança e amor.

AGRADECIMENTOS

Agradeço aos meus pais e à minha irmã, por acreditarem no meu potencial, me apoiarem ao longo de minha jornada acadêmica e tornarem possível a realização deste trabalho.

Agradeço ao meu orientador Eduardo Matins Morgado, por confiar em meu projeto, contribuir com minha formação e sempre incentivar a expansão do meu conhecimento.

Aos docentes que participaram do meu trajeto como discente na Universidade Estadual Paulista – Câmpus de Bauru, especialmente aos associados com o Departamento de Computação.

Aos meus amigos, por vivenciarmos juntos esta etapa da vida, compartilhando sabedoria e momentos inolvidáveis.

E ao Guilherme Mitsuo Saito, por me incentivar e auxiliar no processo de desenvolvimento.

RESUMO

Os altos índices de assaltos residenciais são um reflexo da falta de segurança no Brasil. Um sistema de segurança é um método normalmente utilizado para prevenir tais incidentes, porém seu custo geralmente é elevado. Este trabalho teve como objetivo a construção de um sistema de segurança simples, eficaz e barato.

Um computador portátil do tamanho de um cartão de banco, chamado *Raspberry Pi*, é o componente principal do projeto. Nele, o sistema operacional *Raspbian* foi instalado, e configurações adicionais foram efetuadas. O sensor de presença foi conectado ao computador para ser responsável pela captação da presença de um indivíduo no ambiente. A experiência de testar a sua funcionalidade em uma placa de testes com LEDs (*Light Emitting Diode* – Diodo Emissor de Luz, no português) gerou resultados positivos que motivaram a decisão de integrá-lo como forma de notificação visual no próprio sistema.

Utilizando o conceito de *Internet* das Coisas, que afirma ser possível a inclusão de objetos ordinários na rede de *internet*, o sistema faz uma conexão para enviar dados à um dispositivo móvel através de um aplicativo de comunicação. Um código em *Python* foi desenvolvido no *Raspberry Pi* para manipular os dados do sensor, gerar uma resposta visual através dos LEDs e enviar as mensagens corretas.

Após a criação e o desenvolvimento do *script*, com inúmeros testes, o sistema foi concluído com sucesso. Para explorar um pouco mais o projeto, tirando proveito de um adaptador e um dispositivo de som que possuem a tecnologia *Bluetooth*, foi feita a implementação de um efeito sonoro ao disparar o sensor. Por fim, uma caixa de madeira foi adaptada para armazenar o sistema de forma organizada e apropriada.

Palavras-Chave: Sistema de Segurança, *Internet* das Coisas, *Raspberry Pi*, Sensores de Presença.

ABSTRACT

The high rates of residential assaults are a reflection of the lack of security in Brazil. A security system is a method normally used to prevent such incidents, but its cost in general is high. This work aimed at building a simple, effective and cheap security system.

A portable computer the size of a credit card, called Raspberry Pi, is the main component of the project. In it, the Raspbian operating system was installed, and additional configurations were made. The motion detector has been connected to the computer to be responsible for capturing the presence of an individual in the environment. The testing of its functionality in a protoboard with LEDs (Light Emitting Diode) generated positive results, which motivated the decision of integrating it as a form of visual notification in the system itself.

Using the Internet of Things concept, which claims to be possible to connect ordinary objects to the internet, the system allows the Raspberry Pi to send data collected from the motion sensor to a mobile communication application. A code in Python was developed to the Raspberry Pi to manipulate the sensor data, generate a visual response through the LEDs and send the correct messages.

After the script creation and development, with numerous tests, the system was successfully completed. To further explore the project, a bluetooth adapter and a sound device were added to the system, triggering a sound effect when the motion detector is activated. Finally, a wooden box has been adapted to store the system in an organized and appropriate way.

Keywords: Security System, Internet of Things, Raspberry Pi, Motion Sensors.

LISTA DE ILUSTRAÇÕES

Imagem 1 - A wide variety of end devices will be connected to the Internet of Things	16
Imagem 2 - Raspberry Pi Model A	19
Imagem 3 - Raspberry Pi Model A+	19
Imagem 4 - Raspberry Pi Model B	20
Imagem 5 – Raspberry Pi Model B+.....	20
Imagem 6 - Raspberry Pi 2 Model B	22
Imagem 7 - Raspberry Pi 3.....	23
Imagem 8 - Raspberry Pi Zero	23
Imagem 9 -Comparação de um sinal analógico com um digital	25
Imagem 10 - Linear x Não linear	26
Imagem 11 -Chave Fim-de-Curso Aberta e Fechada.....	27
Imagem 12 - Funcionamento do Reed-Switch	28
Imagem 13 - Foto-Resistor LDR.....	28
Imagem 14 - Curva do LDR.....	29
Imagem 15 - Fotocélula.....	29
Imagem 16 - Fotodiodo	30
Imagem 17 - Fototransistor	31
Imagem 18 -Curvas do NTC e PTC	32
Imagem 19-Sensor piroelétrico	32
Imagem 20 - Sensores Capacitivos.....	33
Imagem 21 - Sensor Indutivo	34
Imagem 22 - Sensor ultrassônico.....	34
Imagem 23 - Ubuntu MATE 16.04 rodando no Raspberry Pi 3	35
Imagem 24 - OSMC	36
Imagem 25 - Conectores do Raspberry Pi 2 B+	39
Imagem 26 - Programa Win32DiskImager	41
Imagem 27 - Remote Desktop Connection	43
Imagem 28 - Sensor PIR.....	44
Imagem 29 - Placa de teste.....	45
Imagem 30 - GPIO do Raspberry Pi 2.....	47

Imagem 31 - getUpdates da API do Telegram	49
Imagem 32 - Sistema Final.....	52

LISTA DE TABELAS

Tabela 1 - Comparação dos modelos do Raspberry Pi 1.....	21
Tabela 2 - Comparação das gerações dos Raspberry Pi.....	24

SUMÁRIO

1	INTRODUÇÃO	12
1.1	PROBLEMA	13
1.2	JUSTIFICATIVA	14
1.3	OBJETIVOS	15
1.3.1	OBJETIVO GERAL	15
1.3.2	OBJETIVOS ESPECÍFICOS	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	INTERNET DAS COISAS	16
2.2	RASPBERRY PI	17
2.2.1	RASPBERRY PI 1	18
2.2.1.1	RaspberryPi Model A	18
2.2.1.2	Raspberry Pi Model A+	19
2.2.1.3	Raspberry Pi Model B	20
2.2.1.4	Raspberry Pi Model B+	20
2.2.2	RASPBERRY PI 2	22
2.2.3	RASPBERRY PI 3	22
2.2.4	RASPBERRY PI ZERO	23
2.3	SENSORES	25
2.3.1	CARACTERÍSTICAS FUNDAMENTAIS	25
2.3.1.1	Tipos de saídas	25
2.3.1.2	Linearidade	26
2.3.1.3	Alcance (<i>Range</i>)	26
2.3.1.4	Velocidade de resposta	26
2.3.2	SENSORES MECÂNICOS	26
2.3.2.1	Chaves Fim-de-Curso	27
2.3.2.2	<i>Reed-Switch</i>	27
2.3.3	SENSORES FOTOELÉTRICOS	28
2.3.3.1	Foto-Resistor (LDR – <i>Light Dependent Resistor</i>)	28
2.3.3.2	Fotocélula	29
2.3.3.3	Fotodiodo	30
2.3.3.4	Fototransistor	30

2.3.4	SENSORES TÉRMICOS.....	31
2.3.4.1	NTC e PTC	31
2.3.4.2	Sensor Piroelétrico	32
2.3.5	SENSOR CAPACITIVO.....	33
2.3.6	SENSOR INDUTIVO	33
2.3.6	SENSOR ULTRASSÔNICO	34
2.4	SISTEMAS OPERACIONAIS PARA RASPBERRY PI	35
2.4.1	UBUNTU MATE.....	35
2.4.2	SNAPPY UBUNTU CORE.....	36
2.4.3	WINDOWS IOT CORE.....	36
2.4.4	OSMC.....	36
2.4.5	OPENELEC.....	37
2.4.6	PINET.....	37
2.4.7	RISC OS.....	37
2.4.8	RASPBIAN	37
2.5	YOWSUP	38
3	MÉTODO DE PESQUISA	39
4	DESENVOLVIMENTO	41
4.1	CONFIGURANDO O RASPBERRY PI.....	41
4.2	CONFIGURANDO A REDE WI-FI	42
4.3	CONFIGURANDO O BLUETOOTH	42
4.4	ESTABELECENDO CONEXÃO REMOTA.....	42
4.5	CRIAÇÃO DE SCRIPT.....	44
4.6	CONFIGURAÇÃO DO SENSOR.....	44
4.7	MONTAGEM DA PLACA DE TESTE.....	45
4.8	PROBLEMAS COM O YOWSUP	48
4.9	INTEGRAÇÃO DO SISTEMA COM O TELEGRAM.....	48
4.10	IMPLEMENTAÇÃO DO BOTÃO RELATÓRIO	50
4.11	IMPLEMENTAÇÃO DO EFEITO SONORO	51
4.12	MONTAGEM DO SISTEMA	52
5	CONCLUSÃO	53
	REFERÊNCIAS.....	54

1 INTRODUÇÃO

Uma das maiores preocupações no Brasil é a falta de segurança. Segundo uma publicação do site Fiquem Sabendo, “A cada dia, 11 casas são invadidas por criminosos em São Paulo” (Ribeiro, 2015). Em outra notícia, o Jornal de Brasília afirma que “Crimes como roubo em residência aumentaram em fevereiro deste ano no Distrito Federal, em relação ao ano anterior — foram 57,9% a mais que no mesmo mês de 2015” (Jornal de Brasília, 2016). Com todos esses dados negativos, a procura por métodos que contribuam com a prevenção de roubos residenciais é grande. O custo de um sistema de segurança como alarmes que utilizam sensores de presença é bem elevado, um problema para quem não possui recursos financeiros. Além disso, alguns sistemas ainda utilizam o método de enviar mensagem de texto no celular como meio de comunicação com o usuário, gerando mais custo para ele ter ciência de quando o alarme foi acionado, uma vez que o próprio usuário irá pagar por essas mensagens. É possível criar um sistema de segurança barato que usufrui dos benefícios da conexão na *internet* para transferir as informações coletadas à um dispositivo móvel.

A *internet* nasceu das redes de computadores utilizadas para rastrear informações e eliminar pontos inimigos durante a Segunda Guerra Mundial. Antes apenas conectava laboratórios de pesquisa, mas com sua evolução atingiu o mundo inteiro.

Se a tecnologia da informação é hoje o que a eletricidade foi na Era Industrial, em nossa época a *Internet* poderia ser equiparada tanto a uma rede elétrica quanto ao motor elétrico, em razão de sua capacidade de distribuir a força da informação por todo o domínio da atividade humana. (Castells, 2001, p. 7)

Pode-se ver a dominância dos *Smartphones* e Redes Sociais recentemente, pois a escala atingida pela a *internet* permite que os usuários se conectem entre si, seja qual for o lugar onde se encontram. Porém, há um novo modo como a *internet* pode ser utilizada: além de conectar pessoas, conectará também objetos comuns, como a geladeira, o automóvel, o interruptor da luz, dentre outros. Esta é a ideia da *Internet* das Coisas (ou como conhecido em inglês, *IoT - Internet of Things*), que visa um futuro repleto de objetos com acesso à rede para facilitar a vida humana. O princípio de IoT pode ser captado na seguinte frase: “As tecnologias mais importantes

são aquelas que desaparecem. Elas se integram à vida no dia a dia até serem indistinguíveis dele” (Weiser, 1991, p. 1). A *Internet das Coisas* é um assunto recente e apontado como uma tendência, um campo promissor cheio de oportunidades. Ela promete também revolucionar na área industrial, trazendo melhorias na produção e, conseqüentemente, na economia mundial.

Um dispositivo chave para essa nova área é o sensor, que pode captar informações do ambiente como temperatura, umidade, presença, etc., e utilizá-las para algum propósito inovador. Para processar os dados coletados é necessário que haja um computador, mas os computadores atuais possuem uma potência muito elevada apenas para manipular dados de sensores, além de ocupar um bom espaço no local. A maneira inteligente de captar informações é em tempo real, exigindo uso exclusivo para essa função. Por esses motivos, uma boa opção para conectar os sensores é o *Raspberry Pi*, um pequeno computador que cabe na palma da mão, cujo custo não é alto (de 5 a 35 dólares) e as linguagens mais utilizadas para se programar nele são *Python* e *Scratch*. Sua criação foi para incentivar crianças a desenvolverem interesse na área da computação, e sua popularidade foi crescendo justamente por se adequar em projetos que envolvem *Internet das Coisas*.

1.1 PROBLEMA

A *Internet das Coisas* é um campo da computação novo e pouco conhecido. A implementação de uma rede de *internet* em coisas ordinárias é uma proposta que visa tornar qualquer coisa inteligente, o que implica que qualquer objeto existente no mundo pode possuir uma comunicação com a rede, e conseqüentemente com qualquer outro objeto. Por exemplo: se houvessem sistemas conectados à uma rede em automóveis, os carros localizados adiante poderiam avisar os anteriores sobre uma forte chuva que pode causar acidentes, ou até mesmo um congestionamento, a fim de alertar motoristas a realizar uma trajetória diferente e evitar engarrafamento. Além disso, abre espaço para que novos produtos e serviços possam ser oferecidos. Tudo isso está em um futuro próximo, e para que ele ocorra cada novo projeto que explora IoT é um passo em direção à essa nova realidade.

Na *internet*, informações são constantemente enviadas e recebidas de qualquer lugar para qualquer lugar em questão de segundos. O desenvolvimento do sistema de segurança conectado à uma rede pode torná-lo mais rápido e eficaz, uma vez que,

ao ser acionado o sensor, a informação coletada pelo *Raspberry Pi* é enviada para um dispositivo móvel e pode ser compartilhada para outros dispositivos no mesmo instante. Quando se trata de segurança, a rapidez é um fator importante e influencia diretamente na eficiência do sistema.

Deve-se ressaltar também que existem residências e comércios que não possuem recursos para arcar uma implementação de um sistema de segurança. Tendo um custo mais baixo, a acessibilidade destes dispositivos aumenta, podendo incluir os menos favorecidos financeiramente.

O projeto também visa incentivar o uso da *Internet* das Coisas, pois há poucos que têm ciência da grandiosidade dessa área e dos benefícios que ela promete oferecer futuramente.

1.2 JUSTIFICATIVA

Um sistema de segurança padrão que possui sensores normalmente gera custos de instalação e monitoramento, utilizando uma tecnologia antiga para a comunicação: a linha telefônica. A implementação da *internet* nesse sistema pode trazer benefícios significativos como acelerar a troca de informação e diminuir o custo.

A *Internet* das Coisas está relacionada com inovação, pretendendo conectar pessoas e coisas do cotidiano através da rede, otimizando nossas vidas. O que hoje são pequenos projetos com *Raspberry Pi*, futuramente serão em automóveis, casas e até em cidades. Muitas oportunidades irão surgir, assim como adaptações por parte dos usuários. Um exemplo foi a implementação do código de barras nos supermercados, que gerou inicialmente um choque até o indivíduo se familiarizar com o sistema, e posteriormente superou qualquer desavença devido a facilidade que ele proporcionou.

Desenvolvendo este projeto, a exploração do *Raspberry Pi* é um fator de destaque. É incrível saber o quão longe a tecnologia chegou, possibilitando a existência de um computador funcional do tamanho de um cartão de banco, cuja versatilidade é notável, possibilitando a criação de diversos projetos inovadores.

Uma das ótimas coisas sobre o *Raspberry Pi* é que não há um único jeito de usá-lo. Se você apenas quer ver vídeos e navegar na *Web*, ou você quer *hackear*, aprender, e criar com a placa, o *Raspberry Pi* é uma plataforma flexível para diversão, utilidade e experimentação. (Richardson & Wallace, 2012-2013, p. xii)

1.3 OBJETIVOS

Este trabalho visa desenvolver um sistema de segurança utilizando sensores de presença conectados ao *Raspberry Pi 2*. Quando acionado um sensor, um aviso será enviado para um dispositivo móvel conectado à *internet*.

1.3.1 OBJETIVO GERAL

Enviar a informação de disparo de um sensor para o *Raspberry Pi* e recebê-lo no dispositivo móvel conectado à *internet*.

1.3.2 OBJETIVOS ESPECÍFICOS

- a. Instalar o sistema operacional *Raspbian* no *Raspberry Pi*;
- b. Instalar e configurar o adaptador de rede sem fio Wi-Fi USB (*Universal Serial Bus*) no *Raspberry Pi*;
- c. Configurar o nome da rede (SSID – *Service Set Identification*, Conjunto de Serviço de Identidade, em português) no *Raspberry Pi*;
- d. Configurar acesso remoto;
- e. Instalar os sensores de movimento (PIR – *Passive Infrared*) no *Raspberry Pi*;
- f. Escrever o código em *python* para a detecção dos sensores, retornando *TRUE* (*verdadeiro*) ou *FALSE* (*falso*);
- g. Editar o código inicial, utilizando a biblioteca *Yowsup*, para implementar a notificação no dispositivo móvel;
- h. Configurar o dispositivo móvel para receber os dados;
- i. Montar um armazenamento para o sistema ficar visivelmente elegante.

2 FUNDAMENTAÇÃO TEÓRICA

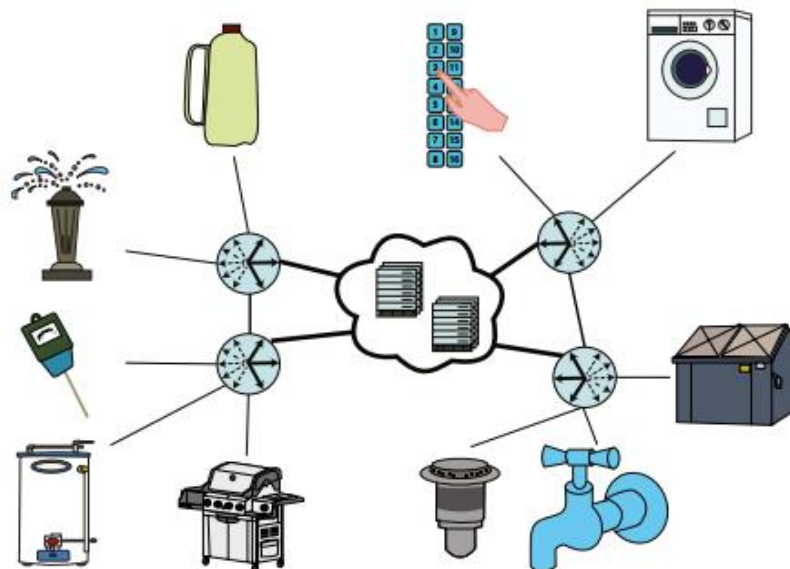
Neste capítulo serão apresentados alguns aspectos teóricos sobre *Internet das Coisas*, *Raspberry Pi*, sensores, sistemas operacionais para *Raspberry Pi* e o *Yowsup* com a finalidade de obter um maior entendimento sobre o tema deste trabalho.

2.1 INTERNET DAS COISAS

Tudo sobre a sociedade não está relacionado com ideias ou informações, mas sim com coisas comuns. Se cada coisa do mundo soubesse qual sua função, seria possível rastrear, diminuir custos e perdas, saber se há necessidade de reparos, dentre outras ações benéficas.

A *Internet das Coisas* nada mais é do que a onipresença da *internet* nos objetos, uma vez que estariam todos conectados em uma rede, realizando comunicações remotas com pessoas e outros objetos, como mostra a Imagem 1.

Imagem 1 - A wide variety of end devices will be connected to the Internet of Things



Fonte: Francis daCosta, 2013, p. 3.

É uma área bastante promissora, que irá gerar incontáveis oportunidades no mercado, prometendo revolucionar o cotidiano e melhorar a qualidade de vida de todos em um futuro próximo. A integração de uma rede pode transformar os objetos, tornando-os inteligentes para ajudar em diversas tarefas comuns que exigem a

interação humana e custam tempo, além de auxiliar em inúmeras profissões a fim de sanar pequenos ou grandes problemas ainda existentes, ou, quem sabe, criar soluções para problemas nunca resolvidos antes.

Sua existência ainda está na fase inicial, mas com o passar do tempo sua tecnologia estará avançada e o mundo será muito mais interessante, com inimagináveis mudanças. Assim como afirmam Vermasan e Friess, “A *Internet* das Coisas continua a afirmar sua posição importante no contexto de Tecnologias da Informação e Comunicação e no desenvolvimento da sociedade.” (Vermesan & Friess, 2013, p. 6).

2.2 RASPBERRY PI

O *Raspberry Pi* é um computador do tamanho de um cartão de crédito e que não custa mais do que 35 dólares, fatos que atraíram a atenção de muitos em seu lançamento.

A ideia inicial começou em 2006 por Eben Upton (s.d.), da Universidade de *Cambridge*, com o propósito de ensinar crianças a programar. Eben notou uma defasagem nas habilidades de estudantes de Ciência da Computação ao longo dos anos, então supôs que algo havia mudado na interação entre crianças e computadores.

A partir de 2008, processadores feitos para celulares tornaram-se melhores e mais baratos, sendo compatíveis com excelentes tipos de multimídia (como, por exemplo, o *blue-ray*). Com essa tecnologia agregada ao *Raspberry Pi*, crianças que inicialmente não demonstravam interesse na placa tiveram uma mudança de opinião.

Com todo o sucesso adquirido, criou-se o *Raspberry Pi Foundation*, fundação que desenvolve a placa e estimula o ensino básico de computação nas escolas.

Os computadores atuais, em geral, são de uso pessoal, desenvolvidos para exercer diversas tarefas comuns que exigem *hardwares* e *softwares* diferentes, como navegar na *Internet*, jogar jogos *online*, etc. Como são voltados para usuários que não precisam de nenhum conhecimento avançado na área da computação, mexer nas peças por diversão não é algo comum. Mesmo os que têm conhecimento avançado muitas vezes não arriscam tentar desenvolver ideias novas em *hardwares* cujo custo é alto. Assim, o *Raspberry Pi* ganha espaço por seu preço acessível e funcionalidades

diversas, sendo possível “banciar a falha” (Torvalds, 2012), como disse o fundador do *Linux*.

Não há uma funcionalidade única para o *Raspberry Pi*: é possível aprender, experimentar e criar inúmeros projetos interessantes. É uma plataforma tão flexível que garante diversão desde colocar uma ideia em prática, até testá-la e utilizá-la.

Existem inúmeras plataformas para a criação de projetos baseados em tecnologia, como por exemplo o *Arduino*, uma placa de desenvolvimento de microcontrolador, bem diferente do *Raspberry Pi*, que é um sistema em um *chip*.

Isso não significa que um é melhor do que o outro, apenas possuem características distintas que são levadas em consideração para sanar as exigências de determinado projeto. No caso do sistema de segurança que será desenvolvido neste trabalho, há o envolvimento de troca de dados via *internet*, o que torna o *Raspberry Pi* uma melhor opção do que o *Arduino*.

Existem quatro tipos de *Raspberry Pi* lançados até hoje, que serão discutidos a seguir.

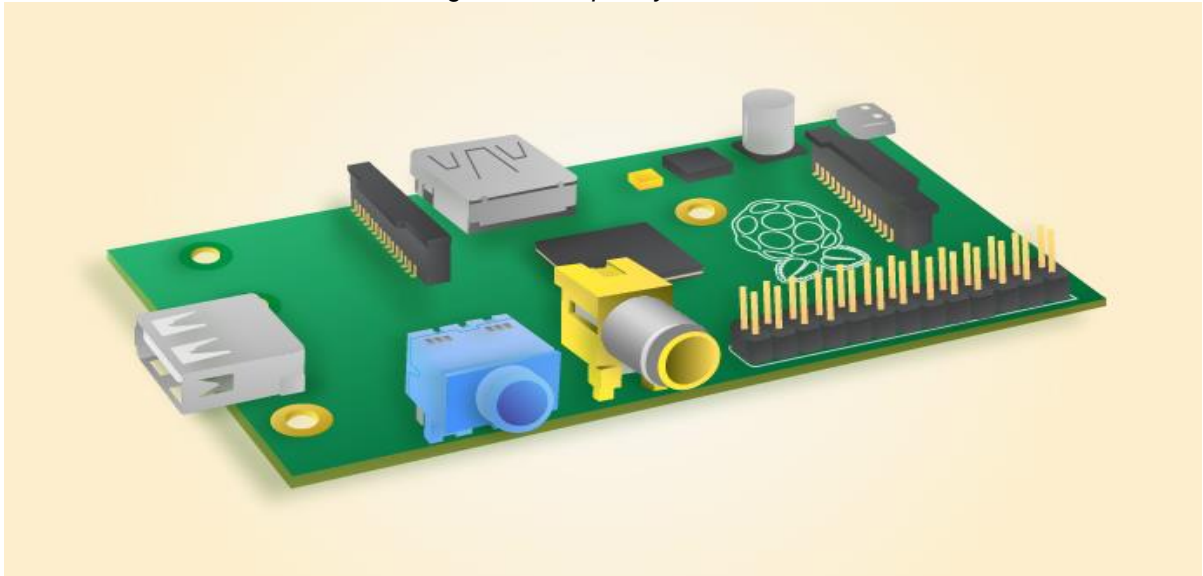
2.2.1 RASPBERRY PI 1

O *Raspberry Pi 1* possui quatro modelos: o A, A+, B e B+, e estes serão explicitados abaixo.

2.2.1.1 *RaspberryPi Model A*

O modelo A possui as mais baixas especificações, por isso é mais utilizado para projetos embutidos. Não possui entrada *Ethernet* e há algumas portas que não existem comparado aos modelos mais atuais, como pode ser observado na Imagem 2. Esse modelo é mais leve e consome menos energia, ideal para projetos de robótica.

Imagem 2 - Raspberry Pi Model A

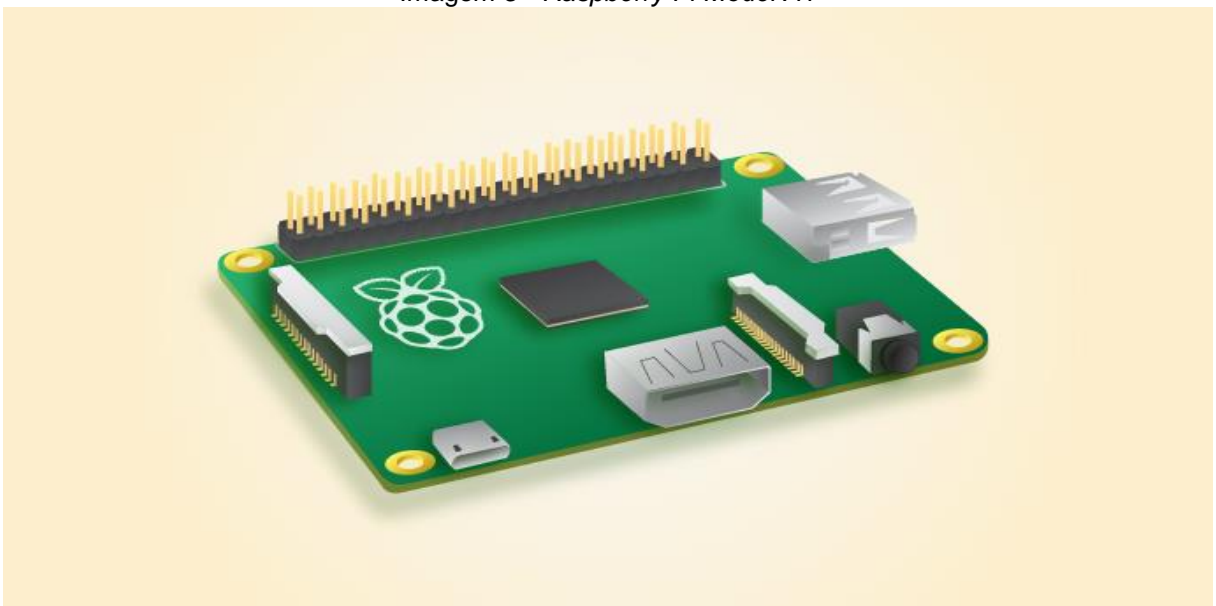


Fonte: <https://www.raspberrypi.org/wp-content/uploads/2014/09/model-a.png>, 2014.

2.2.1.2 Raspberry Pi Model A+

Em novembro de 2014, o modelo A+ substituiu o modelo A. Suas modificações quanto ao modelo A são: um GPIO (*General-Purpose input/Output* - Pinos de Entrada e Saída de Uso Geral, em português) com mais pinos, substituição do soquete antigo para um mais amigável de micro SD (*Secure Digital*), consumo mais baixo de energia, melhora na qualidade sonora e uma redução de 2 centímetros da placa ao todo, como visto na imagem 3.

Imagem 3 - Raspberry Pi Model A+

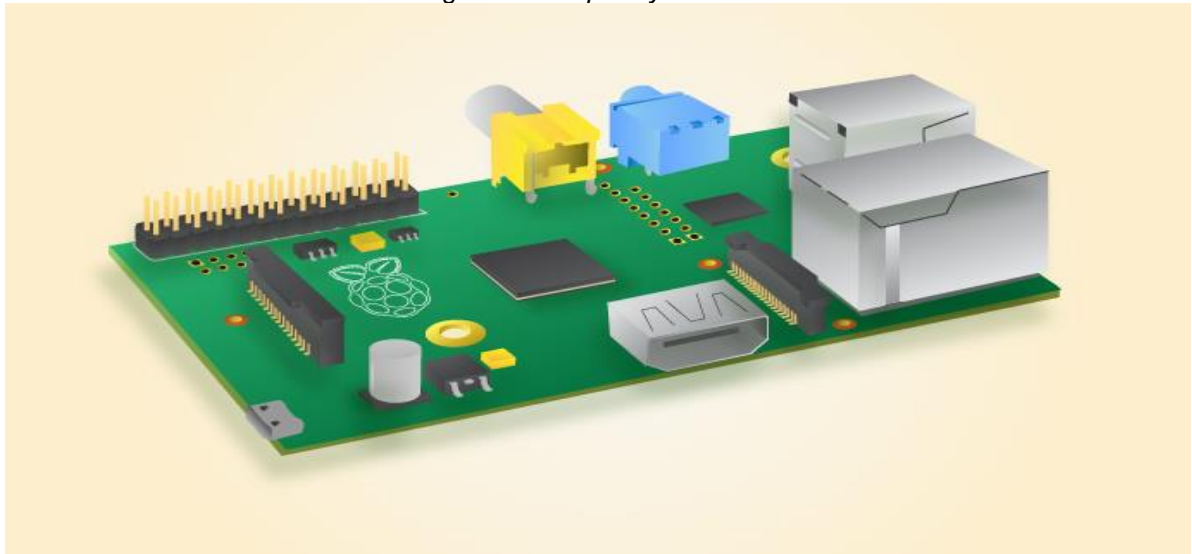


Fonte: https://www.raspberrypi.org/wp-content/uploads/2014/11/Model_A-.png, 2014.

2.2.1.3 Raspberry Pi Model B

O modelo B é o que possui melhores especificações dos *Raspberry Pi* 1, com 512MB de RAM (*Random Access Memory* – Memória de acesso Aleatório, em português), duas portas USB e entrada de *Ethernet* de 100MB. Sua arquitetura pode ser observada na Imagem 4.

Imagem 4 - *Raspberry Pi Model B*

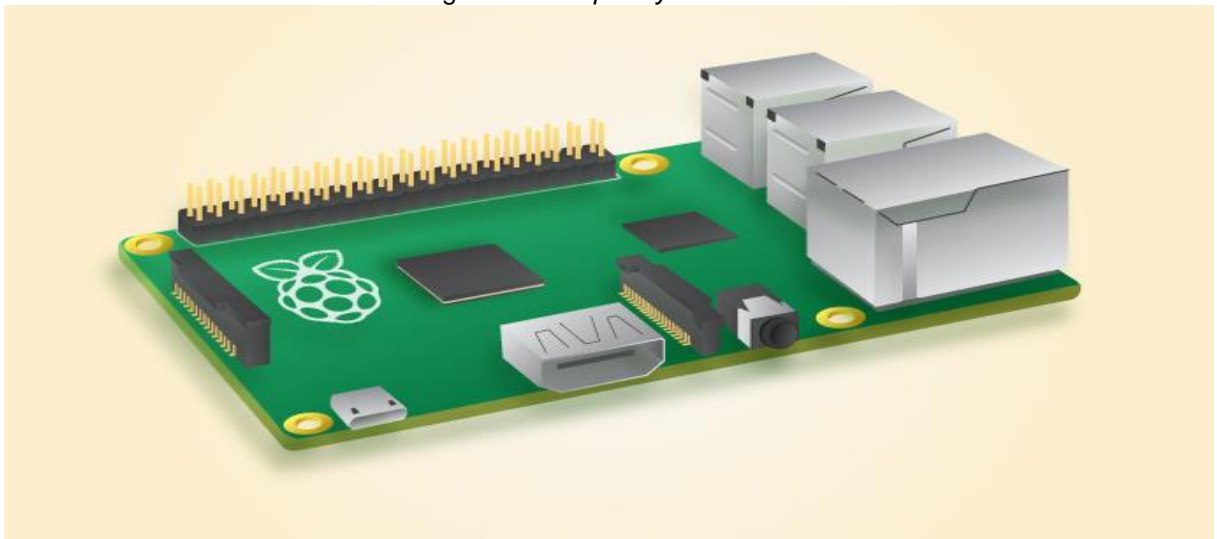


Fonte: <https://www.raspberrypi.org/wp-content/uploads/2014/09/model-b.png>, 2014.

2.2.1.4 *Raspberry Pi Model B+*

O modelo B+ é a versão revisada final do *Raspberry Pi* original. Ele substituiu o modelo B em julho de 2014 e em fevereiro de 2015 foi sucedido pelo *Raspberry Pi 2 Model B*. Sua estrutura final ficou mais transparente, como mostra a Imagem 5.

Imagem 5 – *Raspberry Pi Model B+*



Fonte: <https://www.raspberrypi.org/wp-content/uploads/2013/12/model-b-plus.png>, 2013.

Comparado com o modelo B, o modelo B+ apresenta: Mais pinos no GPIO (40 pinos, 14 a mais do que os modelos A e B), mais entradas USB (de 2 para 4), substituição do soquete antigo para um mais amigável de micro SD, menos consumo de energia, melhor qualidade sonora e uma distribuição mais limpa dos componentes na placa.

A Tabela 1 contém a comparação entre os modelos de Raspberry Pi apresentados quanto suas especificações.

Tabela 1 - Comparação dos modelos do *Raspberry Pi 1*

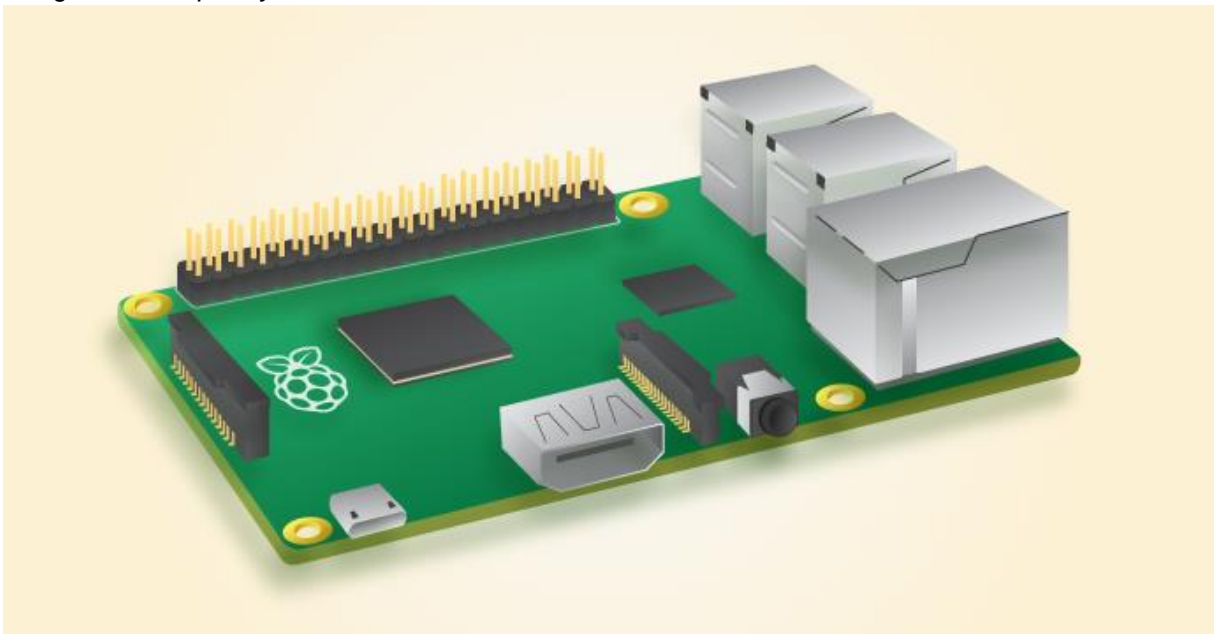
	Modelo A	Modelo A+	Modelo B	Modelo B+
Preço (Dólares)	U\$ 25	U\$ 20	U\$35	U\$ 35 U\$ 25
SoC (Sistema no Chip)	Broadcom BCM2835 (CPU + GPU. SDRAM é um chip separado em cima.)			
CPU	700 MHz ARM11 ARM1176JZF-S core			
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, OpenVG 1080p30 H.264 high-profile encode/decode, 250 MHz			
Memória (SDRAM)iB	128iB, aumentado para 256iB em fev/2012	256iB	256iB, aumentado para 512iB em Out/2015	512iB
Portas USB	1 USB 2.0 por BCM2835		2 USB 2.0 por hub USB integrado em LAN9512	4 USB 2.0 por hub USB integrado em LAN 9514
Saídas de vídeo	Vídeo composto RCA e HDMI (não ao mesmo tempo)	HDMI Vídeo composto requer adaptador de 4 polos	Vídeo composto RCA e HDMI (não ao mesmo tempo)	HDMI Vídeo composto requer adaptador de 4 pólos
Saídas de áudio	Conector TRS pinos de 3.5 mm , HDMI			
Entrada de áudio	Não há, porém é possível conectar um microfone USB			
Armazenamento integrado	Secure Digital SD / MMC / SDIO card slot	Micro Secure Digital / MicroSD slot	Secure Digital SD / MMC / SDIO card slot	Micro Secure Digital / MicroSD slot
Rede integrada	Não há		10/100 Ethernet à cabo RJ45	
Periféricos de baixo nível	26 pinos GPIO, SPI, I ² C, I ² S, UART	40 pinos GPIO, SPI, I ² C, I ² S, pinos I ² C IDC, UART	26 pinos GPIO, SPI, I ² C, I ² S, UART	40 pinos GPIO, SPI, I ² C, I ² S, pinos I ² C IDC, UART
Relógio em tempo real	Não há			
Taxa de alimentação	300 mA, (1.5 W)	600mA up to 1.2A @ 5V	700 mA, (3.5 W)	~650 mA, (3.0 W)
Fonte de energia	5 V (DC) via Micro USB tipo B ou cabeçalho do GPIO			
Tamanho	85.0 x 56.0 mm x 15mm	65.0 x 56.0 mm x 12mm	85.0 x 56.0 mm x 17mm	85.0 x 56.0 mm x 17mm
Peso	31g	23g	40g	40g

Fonte: Elaborado pelo autor.

2.2.2 RASPBERRY PI 2

O *Raspberry Pi 2 Model B* é a segunda geração da placa. Em fevereiro de 2015, substituiu o *Raspberry Pi 1 Model B+*. Possui uma CPU (*Central Processing Unit* – Unidade Central de Processamento, em português) *ARM Cortex-A7 quad-core* de 900MHz e 1GB de RAM. Mesmo sendo seu sucessor, possui compatibilidade completa com a geração anterior. Devido ao seu processador *ARMv7*, é possível fazer rodar todos as distribuições do *ARM GNU/Linux*, assim como *Windows 10* da *Microsoft*. A Imagem 6 contém a representação de sua estrutura, semelhante ao modelo *Raspberry Pi 1 Model B*.

Imagem 6 - *Raspberry Pi 2 Model B*

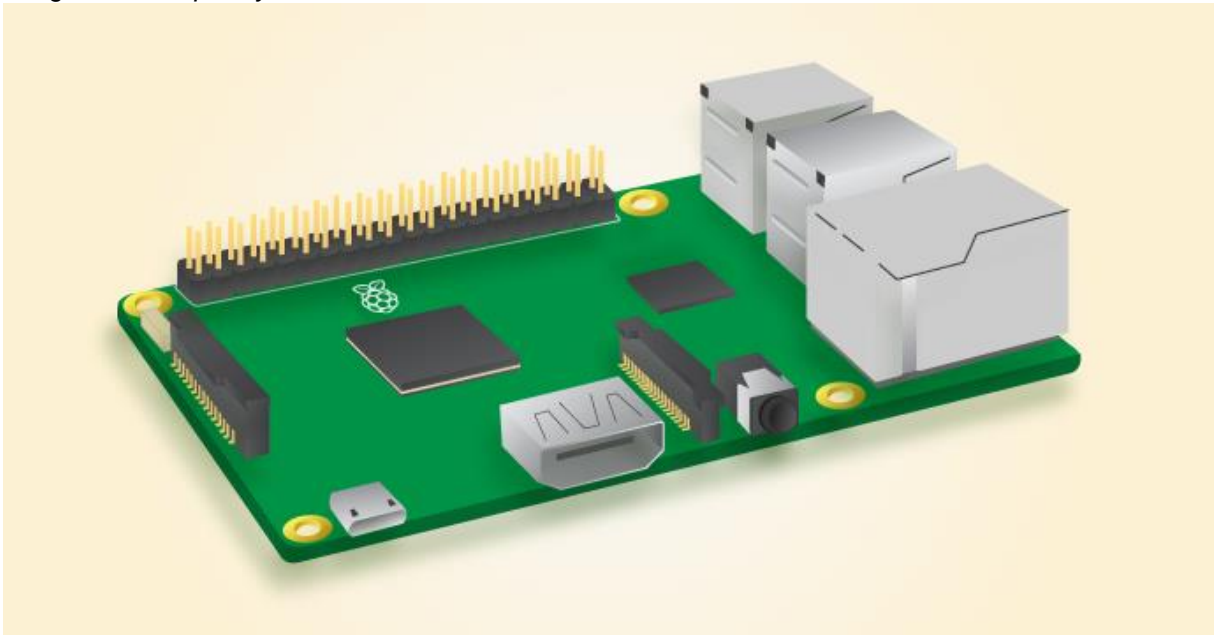


Fonte: https://www.raspberrypi.org/wp-content/uploads/2015/02/Pi_2_Model_B.png, 2015.

2.2.3 RASPBERRY PI 3

O *Raspberry Pi 3 Model B* é a terceira geração do *Raspberry Pi*. Ele substituiu o *Raspberry Pi 2 Model B* em fevereiro de 2016. Além das especificações básicas da segunda geração, apresenta também uma CPU *ARMv8 64 bits quad-core* de 1.2GHz, 802.11n LAN (*Local Area Network* – Rede de Área Local, em português) *Wireless*, *Bluetooth 4.1* e *Bluetooth Low Energy* (BLE). Sua distribuição de componentes na placa é semelhante ao *Raspberry Pi 2*, como observado na Imagem 7 abaixo, e é completamente compatível com ambas as gerações anteriores.

Imagem 7 - *Raspberry Pi 3*



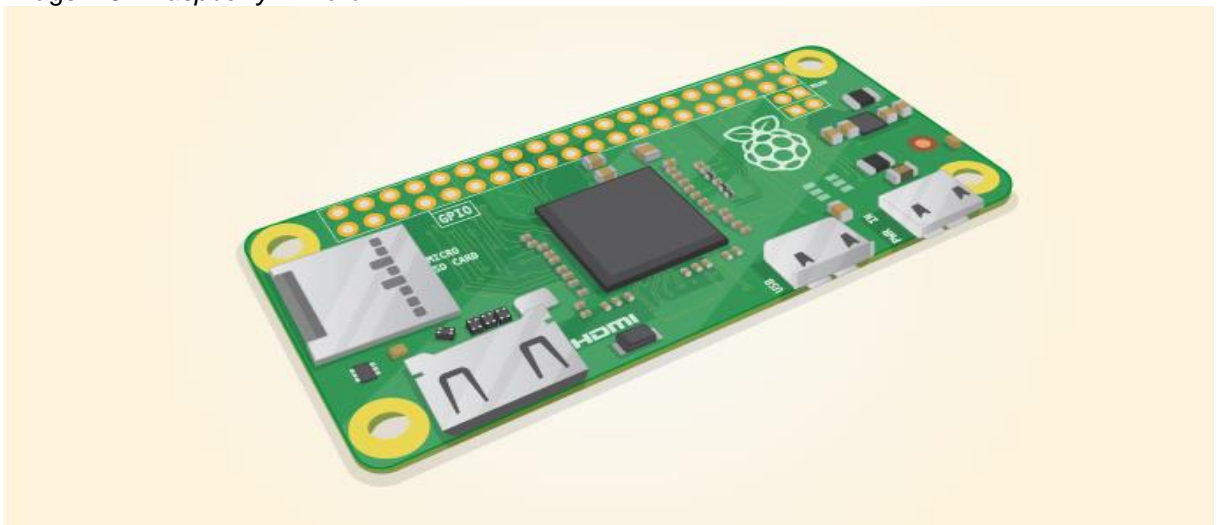
Fonte: https://www.raspberrypi.org/wp-content/uploads/2016/02/Pi_3_Model_B.png, 2016.

Essa versão é recomendada para utilização em escolas ou de uso geral, pois para projetos cuja placa será embutida, a melhor opção seria o Modelo A+ ou o *Raspberry Pi Zero*, descrito a seguir.

2.2.4 RASPBERRY PI ZERO

O *Raspberry Pi Zero* é metade do tamanho do Modelo A+, porém com o dobro de utilidade. Sua arquitetura muda bastante comparado aos outros modelos apresentados até agora, sendo bem mais compacto, como mostra a Imagem 8.

Imagem 8 - *Raspberry Pi Zero*



Fonte: https://www.raspberrypi.org/wp-content/uploads/2015/11/Pi_Zero_v1.2.png, 2015.

Na Tabela 2 em seguida, a comparação das gerações dos *Raspberry Pi*:

Tabela 2 - Comparação das gerações dos *Raspberry Pi*

	Raspberry Pi 2	Raspberry Pi 3	Raspberry Pi Zero
Preço (Dólares)	U\$ 35		U\$5
SoC (Sistema no Chip)	Broadcom BCM2836	Broadcom BCM2837	Broadcom BCM2835
CPU	900 MHz quad-core ARMv7 Costex-A7	1.2 GHz 64-bit quad-core ARMv8 Costex-A53	1000MHz Low Power ARM1176JZ-F
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, OpenVG 1080p30 H.264 high-profile encode/decode, 250 MHz	Broadcom VideoCore IV, OpenGL ES 2.0, OpenVG 1080p60 H.264 high-profile encode/decode, 400 MHz	Broadcom VideoCore IV
Memória (SDRAM)iB	1024MiB		512MiB
Portas USB	4 USB 2.0 por hub USB integrado em LAN9514		1 Micro USB OTG (On The Go)
Saídas de vídeo	HDMI Vídeo composto requer adaptador de 4 polos		HDMI, Vídeo composto via cabeçalho de 2 pinos não soldados
Saídas de áudio	Conector TRS pinos de 3.5 mm, HDMI		Audio HD multi-canal sobre HDMI
Entrada de áudio	Não há, porém é possível conectar um microfone USB		
Armazenamento integrado	Micro Secure Digital / MicroSD slot		
Rede integrada	10/100 Ethernet à cabo RJ45	10/100 wired Ethernet RJ45, 802.11n Wi-Fi & Bluetooth 4.1 integrados	Não Há
Periféricos de baixo nível	40 pinos GPIO, SPI, I ² C, I ² S, pinos I ² C IDC, UART		40 pinos GPIO, SPI (não populado)
Relógio em tempo real	Não há		
Taxa de alimentação	~650 mA, (3.0 W)	800mA (4.0 W), até 2.5A	160mA
Fonte de energia	5 V (DC) via Micro USB tipo B ou cabeçalho do GPIO		
Tamanho	85.0 x 56.0 mm x 17mm	85.6 x 56.5 mm x 17mm	65.0 x 30.0 mm x 5mm
Peso	40g	45g	9g

Fonte: Elaborado pelo autor.

A versão que será utilizada neste trabalho é o *Raspberry Pi 2 Model B*. O ideal seria o da terceira geração, uma vez que possui *Wireless* embutido, porém a compra do aparelho foi feita antes do seu lançamento. Para sanar essa necessidade, será instalado um adaptador de rede sem fio Wi-Fi USB na placa.

2.3 SENSORES

Atualmente, a computação e a *internet* ainda são dependentes de ações humanas para exercerem funções: seja para ligá-los, configurá-los, apertar botões, etc. O ser humano, por sua vez, possui seu tempo, atenção e precisão escassos, o que significa que são péssimos em coletar dados do ambiente. Um exemplo disso é a temperatura, não somos capazes de dizer exatamente qual a temperatura que estamos sentindo no momento, já um sensor de temperatura consegue.

Em geral, sensores são dispositivos que captam informações do ambiente e os transformam em dados para um sistema. Podem recolher inúmeras variáveis externas cujo ser humano não é capaz de captar precisamente.

Normalmente os sensores possuem um circuito de interface que manipula os dados captados para que o controlador consiga ler.

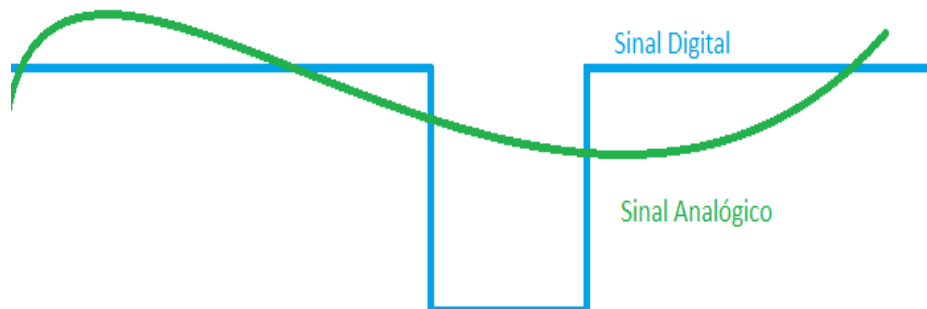
2.3.1 CARACTERÍSTICAS FUNDAMENTAIS

Os sensores possuem características importantes que devem ser observadas para a escolha ideal, dependendo da aplicação que se deseja desenvolver.

2.3.1.1 Tipos de saídas

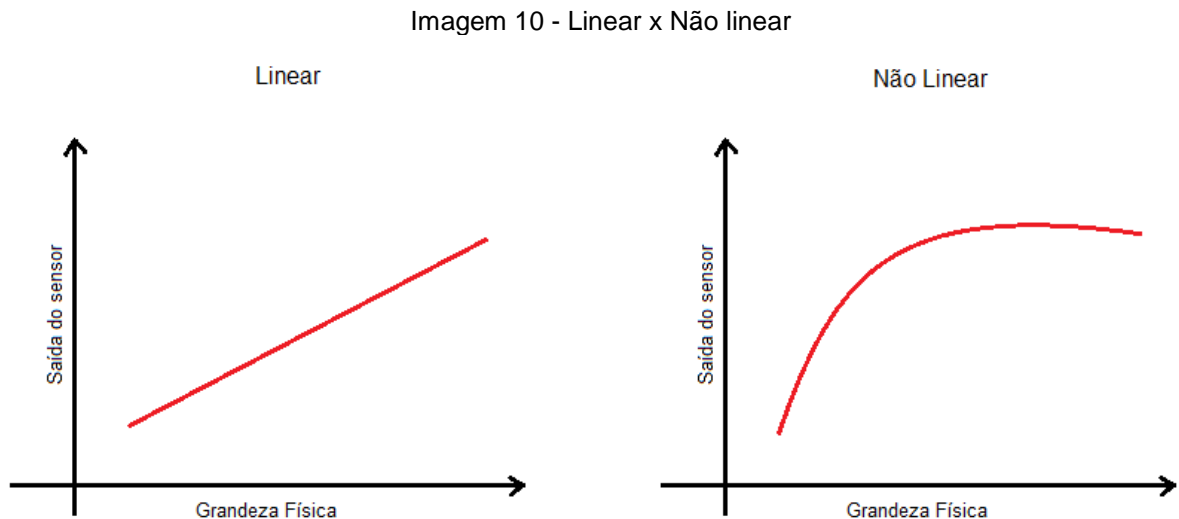
Existem dois tipos de saída de sensores: analógicas ou digitais. As analógicas podem retornar qualquer valor, enquanto as digitais retornam apenas zero ou um (saída binária), conforme ilustra a Imagem 9. Como o sensor que será utilizado neste trabalho é o de presença, será um sensor digital: o sinal um significa que o sensor detectou presença, e o zero indica que não há movimento no ambiente.

Imagem 9 - Comparação de um sinal analógico com um digital



2.3.1.2 Linearidade

A linearidade é a curva que se forma entre a saída do sensor com a grandeza física medida, podendo ser analisada na Imagem 10. Existem alguns sensores que não são lineares.



Fonte: Elaborado pelo autor.

2.3.1.3 Alcance (*Range*)

O alcance (ou *range*) é o intervalo que contém todos os valores de entrada de um sensor.

2.3.1.4 Velocidade de resposta

A velocidade de resposta é o tempo que demora para a obtenção dos valores de saída do sensor. É recomendado que o sensor possua uma resposta imediata para não reduzir a eficiência do sistema e não comprometer a qualidade do projeto.

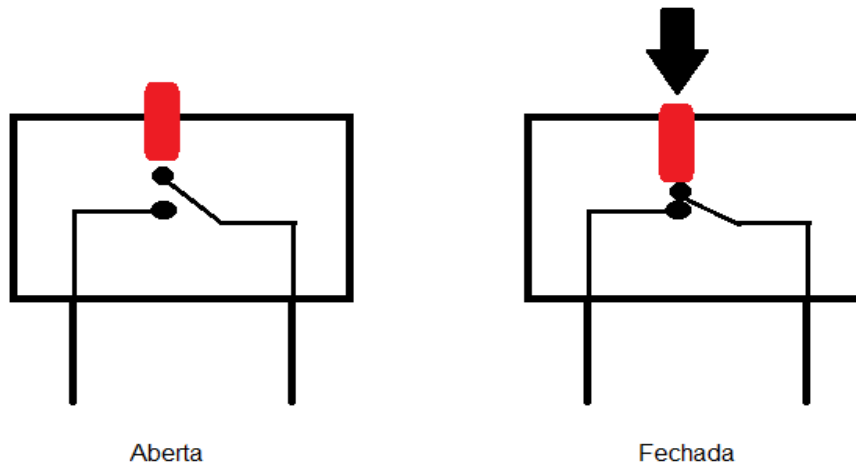
2.3.2 SENSORES MECÂNICOS

Sensores mecânicos detectam movimentos, presença ou posições com o auxílio de recursos mecânicos (como chaves). Como possuem peças móveis, estão sujeitos à quebra ou desgaste. A inércia natural é outro fator que compromete a velocidade de ação, além do repique, que ao acionado, pode enviar um sinal falso.

2.3.2.1 Chaves Fim-de-Curso

Atuam como interruptor, são chaves que exercem a função de liga e desliga, ao interromper ou não um circuito. Na Imagem 11, o desenho do lado esquerdo mostra o circuito aberto devido ao botão que não está acionado. Já na imagem da direita, o botão é pressionado, fechando o circuito.

Imagem 11 -Chave Fim-de-Curso Aberta e Fechada

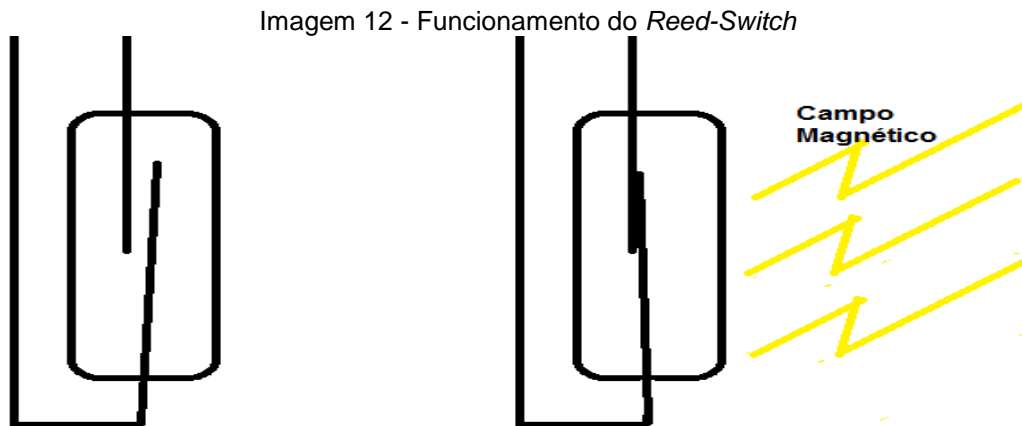


Fonte: Elaborado pelo autor.

Sua finalidade principal é descobrir se um dispositivo atingiu ou não seu deslocamento máximo, mas também pode servir para detectar a abertura ou fechamento de portas, deslocamentos de posições e a presença de determinado objeto.

2.3.2.2 Reed-Switch

Sua principal característica é funcionar com influência de um campo magnético. A Imagem 12 ilustra o seu funcionamento: supondo que existam dois contatos separados, ao entrar no campo magnético, ambos se juntam e fecham o circuito.



Fonte: Elaborado pelo autor

2.3.3 SENSORES FOTOELÉTRICOS

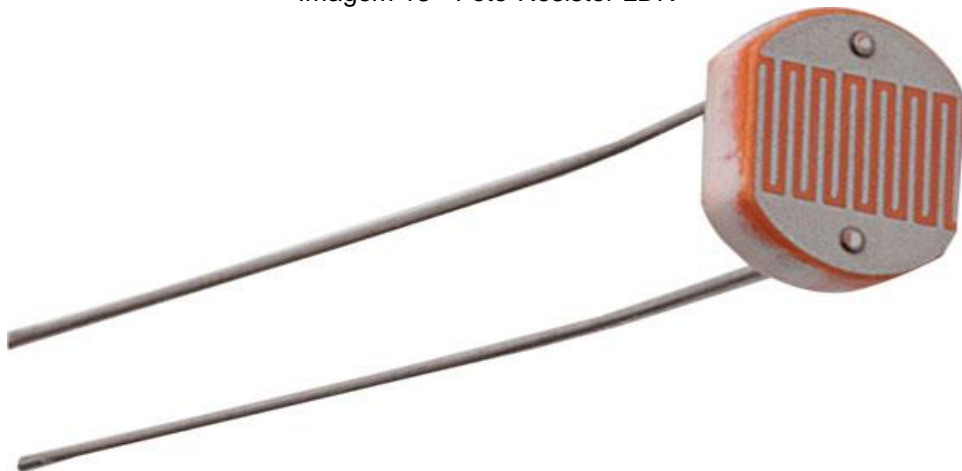
São sensores que reagem à luz. Sua velocidade de resposta é bem maior do que os sensores mecânicos, além de não possuir o fator da inércia e não quebrarem ou se desgastarem, pois não possuem peças mecânicas.

2.3.3.1 Foto-Resistor (LDR – *Light Dependent Resistor*)

De acordo com a Imagem 13, sua superfície possui Sulfeto de Cádmio (CdS) e sua resistência elétrica depende da quantidade de luz que o atinge.

A vantagem da utilização desse sensor está em sua grande sensibilidade e na possibilidade de manusear correntes relativamente altas. Porém, sua velocidade está em torno de dezenas de quiloherztz, sendo bastante lentos em suas operações.

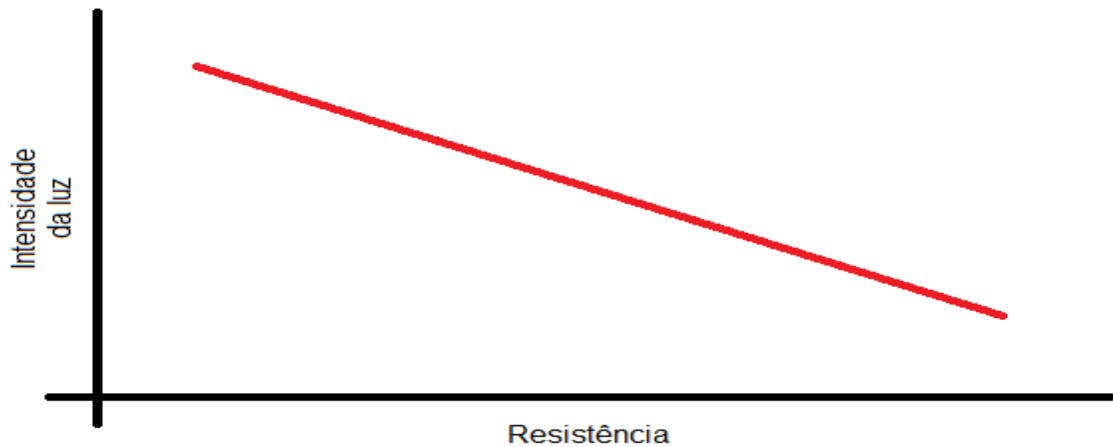
Imagem 13 - Foto-Resistor LDR



Fonte: <https://meetarduino.files.wordpress.com/2012/05/ldr.jpg>, 2012.

A sua resistência cai na medida em que a incidência de luz aumenta, segundo a Imagem 14. Essa curva de resposta é semelhante ao dos olhos humanos, o que permite a captação de luzes convencionais de ambiente e lâmpadas.

Imagem 14 - Curva do LDR



Fonte: Elaborado pelo autor.

2.3.3.2 Fotocélula

As fotocélulas (ou células fotoelétricas) parecem com uma placa, de acordo com a Imagem 15. Ao captarem luminosidade em sua superfície, geram tensões elétricas. Além de servirem como sensores, também podem gerar energia a partir da luz solar.

Imagem 15 - Fotocélula



Fonte: <http://www2.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWending/4---sensores-v2.0.pdf>,

2010.

Em geral, são melhores que os sensores LDR, pois são mais sensíveis e rápidos, abrindo mais oportunidades para o desenvolvimento de projetos.

A fotocélula age como um gerador, sendo baseado em transistores e amplificadores para gerar uma tensão de saída.

2.3.3.3 Fotodiodo

É um dispositivo pequeno, como na Imagem 16, no entanto bastante eficaz. Sua atuação ocorre com base no princípio de que, ao incidir fótons na sua junção semicondutora, haja a liberação de portadores de carga. É possível que esses portadores gerem uma tensão entre os terminais do diodo ou afetem sua resistência à passagem de luz.

Sua velocidade de atuação é rápida, tendo em vista sua necessidade de circuitos bons de amplificação por serem demasiadamente sensíveis.

Imagem 16 - Fotodiodo



Fonte: http://es.farnell.com/productimages/standard/en_GB/42251003.jpg, sem data.

É possível utilizá-los de duas formas distintas: Como gerador, onde há uma tensão produzida, e no modo resistivo, fazendo com que a corrente reversa seja alterada quando há captação de luz.

2.3.3.4 Fototransistor

Assim como os fotodiodos, os fotosensores também liberam cargas quando há ocorrência de luz nas junções. A única diferença está na capacidade do fototransistor de ampliar as correntes produzidas no processo.

Sua curva é bastante semelhante com a do fotodiodo e das fotocélulas, possuindo grande sensibilidade. Além disso, é possível aumentar a sensibilidade utilizando o terminal de base (se houver um). A Imagem 17 ilustra sua arquitetura.

Imagem 17 - Fototransistor



Fonte:

http://proesi.com.br/media/catalog/product/cache/1/image/403x303/9df78eab33525d08d6e5fb8d27136e95/f/o/foto_transistor_4ec51fa97f764.jpg, sem data.

2.3.4 SENSORES TÉRMICOS

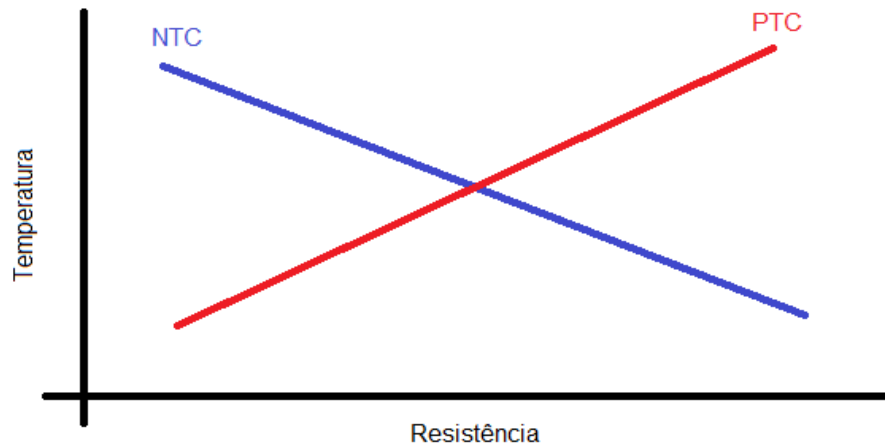
São sensores que atuam no meio para a captura da variação de temperatura.

2.3.4.1 NTC e PTC

Os NTC (*Negative Temperature Coefficient* – Coeficiente de Temperatura Negativo) “São resistores não lineares cuja resistência varia sob a ação da temperatura. Em outras palavras, são semicondutores que conduzem melhor a corrente elétrica no estado quente do que frio.” (Zuim, s. d.). Portanto, quando a temperatura aumenta, a resistência elétrica diminui e seu coeficiente de temperatura é considerado negativo.

O PTC (*Positive Temperature Coefficient* – Coeficiente de Temperatura Positivo) é uma resistência que atua de forma oposta ao NTC: conforme a temperatura aumenta, o PTC aumenta. A Imagem 18 demonstra o comportamento antagônico dos dois:

Imagem 18 -Curvas do NTC e PTC



Fonte: Elaborado pelo autor.

São bastante utilizados, pois seu custo é baixo e seu manuseio é fácil, além da sua faixa de variação de temperatura ser grande.

2.3.4.2 Sensor Piroelétrico

Estão presentes nos sensores de presença (apresentado na imagem 19) e nos alarmes de incêndio. Seu funcionamento se deve à uma substância que se polariza na presença de radiação infravermelha e gera tensão, que pode ser amplificada. A pessoa consegue ser detectada pois o corpo gera calor suficiente para que o infravermelho capte.

Este é o tipo de sensor que será utilizado no projeto.

Imagem 19-Sensor piroelétrico



Fonte: http://www.mastertec.ind.br/2010/imagens/sensor_sip4.jpg, 2010.

2.3.5 SENSOR CAPACITIVO

Gera um campo eletrostático e monitora-o, para que qualquer sinal de mudança nele seja detectado. Possui um oscilador, um retificador de sinal, um circuito de filtragem e um de saída, que podem ser notados na Imagem 20.

Existem dois tipos de sensores capacitivos: os analógicos, que depende da distância entre duas placas, do material delas e o dielétrico entre elas, e os digitais, que possuem o oscilador inativo até o alvo se aproximar e modificar a capacitância do circuito, ativando o oscilador para ativar o circuito de saída quando atingir determinado valor.

Imagem 20 - Sensores Capacitivos



Fonte: <http://ecatalog.weg.net/files/produtos/sensores-capacitivos-G.jpg>, sem data.

2.3.6 SENSOR INDUTIVO

São sensores que detectam a presença de elementos metálicos que atravessam seu campo magnético, sem o contato direto. Sua composição é um núcleo envolvido por uma bobina, tal como a Imagem 21.

Ao detectar objetos como ímãs, compostos por ferro, e mesmo os diamagnéticos (dispensam as linhas de força de um campo magnético) a bobina se altera, pois interferem o campo gerado por um oscilador que está conectado na bobina. Ao sentir a variação de uma determinada distância, o sensor comuta.

Eles não necessitam de energia mecânica, a comutação é em alta velocidade, não sentem impactos de vibrações e choques e não necessitam de contato direto com a peça.

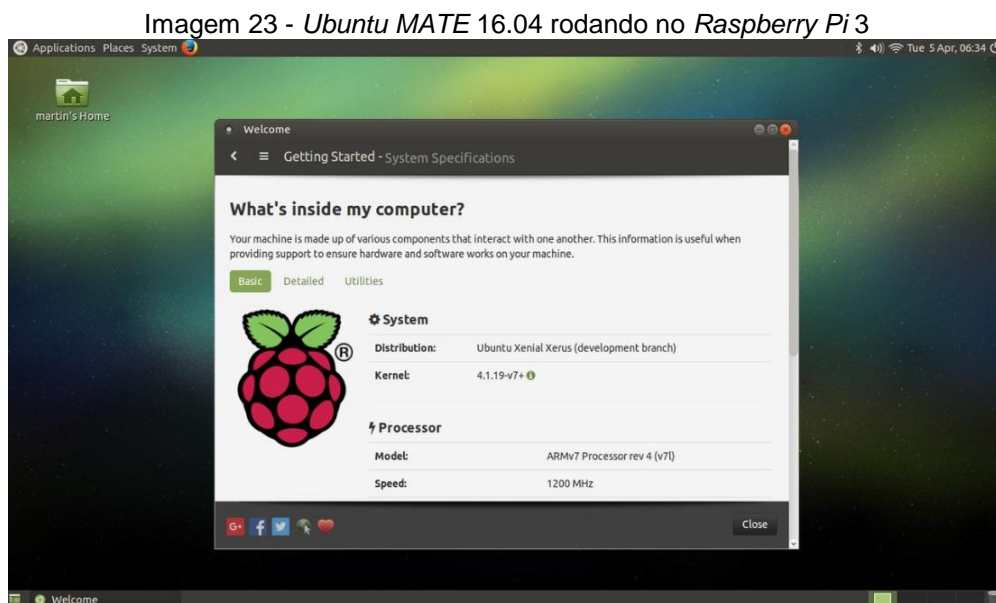
2.4 SISTEMAS OPERACIONAIS PARA *RASPBERRY PI*

O sistema operacional oficial da Fundação *Raspberry Pi* é o *Raspbian*, apesar de não serem afiliadas. Existem, porém, inúmeras imagens de sistemas operacionais feitos por terceiros que rodam na placa. Nesta seção, algumas delas serão analisadas.

2.4.1 *UBUNTU MATE*

O *Ubuntu MATE* é um sistema operacional semelhante ao *Ubuntu* dos computadores convencionais, assim como ilustrado na Imagem 23. É designado para rodar em *Raspberry Pi 2* e 3.

Desenvolvido por Martin Wimpress e Rohithv Madhavan, a imagem foi fundamentada na base *armhf* (*ARM Hard Float Port*) do *Ubuntu* convencional, o que significa que as aplicações podem ser instaladas da mesma forma que no *desktop* (via *apt-get*) e é possível utilizá-las normalmente, como, por exemplo, o *Libre Office*. A sua versão 16.04 funciona perfeitamente com os componentes extras do *Raspberry Pi 3*, o *Bluetooth* e o *Wi-Fi*.



Fonte: https://ubuntu-mate.org/gallery/Screenshots/09_RASPBERRY.png, sem data.

2.4.2 SNAPPY UBUNTU CORE

O *Snappy Ubuntu Core* é uma versão reduzida do *Ubuntu* com o intuito de rodar em dispositivos conectados na *internet* e em máquinas autônomas (desde objetos dentro da própria casa até *drones*) com segurança. É possível usar não só em placas de desenvolvimento e em dispositivos com *internet*, mas também em instâncias na nuvem ou em qualquer lugar de uma máquina virtual.

2.4.3 WINDOWS IOT CORE

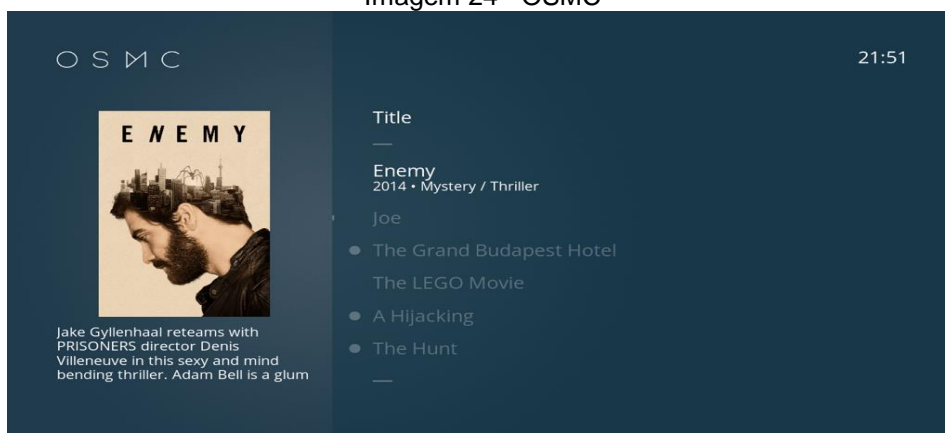
Essa é a versão reduzida do *Windows* para dispositivos pequenos com ou sem tela. Ela utiliza a API *Universal Windows Platform* (UWP), que é rica e extensível.

Seus requisitos são: 256MB (128MB para o sistema operacional) de memória RAM para sistemas sem *display* de vídeo, 512MB (256MB para o sistema operacional) de memória RAM para sistemas com *display* de vídeo e um processador de 400MHz ou superior.

2.4.4 OSMC

Baseado no *Kodi*, um projeto já existente, e em *Linux*, o OSMC (*Open Source Media Center*) é um reprodutor de mídia grátis cujo código fonte é aberto. Por ser fácil de usar, gratuito, rápido, com atualizações regulares, o OSMC possui características que atraem a comunidade. Sua aparência é simples e agradável, como na Imagem 24.

Imagem 24 - OSMC



Fonte: <https://osmc.tv/assets/img/skin/skin4.png?v=94e0c6a9a3>, sem data.

2.4.5 OPENELEC

O *OpenELEC*, também baseado no *Kodi*, é um sistema operacional integrado. Seu projeto foi pensado para ser leve em tamanho e complexidade, sendo ideal para sistemas pequenos, além de apresentar facilidades com relação às configurações.

2.4.6 PINET

PiNet é um projeto desenvolvido para ajudar escolas a configurar e manusear o *Raspberry Pi* nas aulas. Seu código é aberto e sua construção contou com professores que coletaram a experiência adquirida em 15 países.

É instalado um *software* de servidor em um computador com *Linux* 14.04, para assim conectar o servidor e os *Raspberry Pis* juntos, utilizando uma *internet* via cabo.

2.4.7 RISC OS

O sistema operacional *RISC OS* foi designado pela *Acorn*, em *Cambridge*, na Inglaterra. Lançado em 1987, sua origem nos leva ao time original que desenvolveu os microprocessadores ARM.

Testado por uma comunidade fiel de desenvolvedores e usuários, ele é rápido, compacto e eficiente. Como não é uma versão de *Linux* nem *Windows*, possui suas próprias funções e aparência.

2.4.8 RASPBIAN

Raspbian é o sistema operacional grátis baseado em *Debian* (sistema operacional criado por indivíduos que defendem a criação de sistemas operacionais livres), otimizado para o *Raspberry Pi*. É possível executar funções básicas na placa, porém inclui também aproximadamente 35 mil pacotes com *softwares* de fácil instalação.

Foi escolhido esse sistema operacional devido ao seu fator de otimização, pois as funções no *Raspberry Pi* são executadas de forma mais leve, além de uma variedade de pacotes que podem ser utilizados.

2.5 YOWSUP

Yowsup é uma biblioteca criada na linguagem *python*, que permite a criação de aplicações que utilizam o serviço do *WhatsApp*.

O *WhatsApp Messenger* é um aplicativo que permite trocar mensagens sem haver a imposição de pagar, caso haja uma conexão com a *internet*. Inicialmente foi desenvolvido para celulares, mas atualmente é multiplataforma.

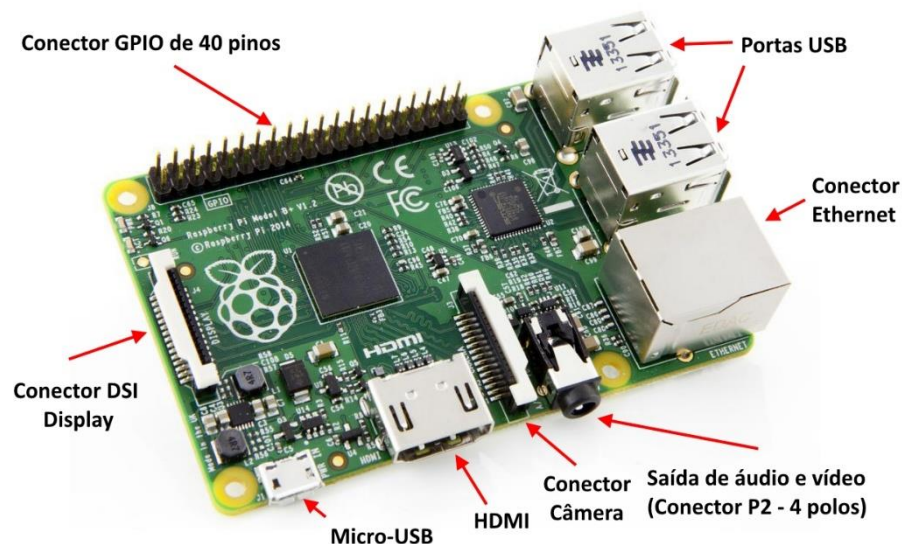
3 MÉTODO DE PESQUISA

Os passos iniciais do trabalho se concentraram no *Raspberry Pi*. Primeiro houve a instalação do sistema operacional *Raspbian*. Como a versão da placa é o *Raspberry Pi 2*, não há Wi-Fi embutido, exigindo a instalação e configuração de um adaptador de Wi-Fi em algumas das portas USB indicadas na Imagem 25. Após sua devida instalação, sua rede foi localizada na lista para que fosse estabelecida a conexão.

Para facilitar a interação com a placa, as configurações da conexão remota foram aplicadas, com o objetivo de acessar o *Raspberry Pi* de um computador.

Os sensores tiveram sua devida instalação, e um *script* inicial de teste foi criado. A biblioteca *Yowsup* seria utilizada no código como responsável pelo auxílio na implementação do envio de alertas para o dispositivo móvel, porém o aplicativo *WhatsApp* possui um sistema de detecção e bloqueio de mensagens constantes, por considera-las *spam*. Em seu lugar, o aplicativo *Telegram* foi adotado.

Imagem 25 - Conectores do *Raspberry Pi 2 B+*



Fonte: <http://blog.filipeflop.com/wp-content/uploads/2014/08/raspberry-pi-b-plus-conectores1.jpg>, 2014.

O dispositivo móvel foi configurado com o auxílio da API (*Application Programming Interface* – Interface de Programação de Aplicativos) *Telegram BOT* para receber as notificações, e ao mesmo tempo o *script* foi sendo ampliado para

existir a conexão via *web*. Ao terminá-lo, uma sessão de testes garantiu que o programa funcionasse corretamente.

A última fase se limitou em construir uma identidade visual com o objetivo de torna-lo organizado e apresentável.

Por fim, testes finais e correções de possíveis falhas foram analisadas e executadas para que o trabalho atingisse o seu objetivo.

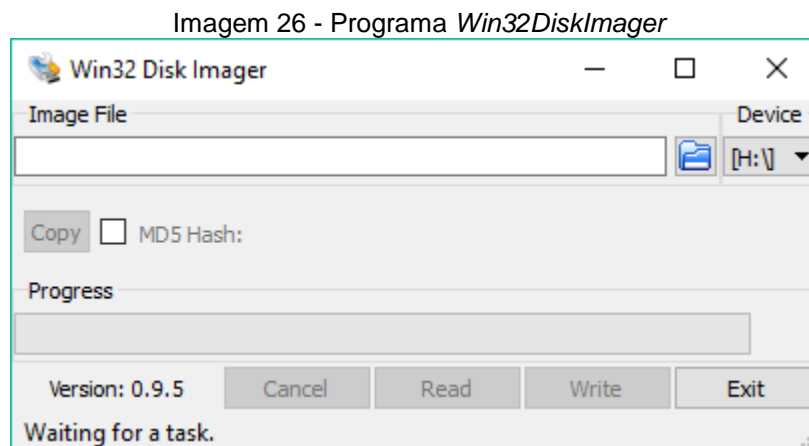
4 DESENVOLVIMENTO

Durante o desenvolvimento do projeto, algumas modificações tiveram que ser aplicadas. Uma delas foi a substituição da biblioteca *Yowsup* pelo *Telegram BOT*. Além disso, alguns detalhes foram incorporados para que o produto final se tornasse mais completo, como a utilização de um adaptador *bluetooth* que se conecta à um dispositivo de áudio, a fim de reproduzir um alerta sonoro no momento em que o sensor é disparado.

É importante ressaltar que foi utilizado o sistema operacional *Windows* no computador cujo trabalho foi realizado, e alguns procedimentos são executados com êxito apenas neste sistema.

4.1 CONFIGURANDO O RASPBERRY PI

Em <https://www.raspberrypi.org/downloads/raspbian/> encontra-se a imagem mais atual do sistema operacional *Raspbian*, com o nome *RASPBIAN JESSIE WITH PIXEL*. Após baixa-la, utiliza-se o programa *Win32DiskImager*, que irá auxiliar na gravação da imagem, como mostrado na Imagem 26.



Fonte: Elaborado pelo autor.

No espaço em branco, coloca-se a localização da imagem, que pode ser facilmente encontrada clicando na pasta azul. É necessário prestar atenção em qual letra o computador atribuiu para o cartão SD (encontrado em Meu Computador, no menu Iniciar) e selecioná-lo ao lado direito da pasta azul, em *Device*. Feito isso, inicia-se a gravação clicando no botão *write* e em alguns instantes estará completa. O cartão

SD já pode ser inserido no *Raspberry Pi* e, ao ligar a placa, o sistema irá efetuar a primeira inicialização com todas as instalações. É recomendável conectar um monitor na entrada HDMI (*High Definition Multimedia Interface* – Interface Multimídia de Alta Definição) para verificar o processo de instalação e inserir teclado e mouse nas portas USB, pois durante o processo pede-se a confirmação de instalações e no final exige um usuário e senha para acesso futuro. Feito isso, a placa já está com sistema operacional e pronta para ser utilizada.

4.2 CONFIGURANDO A REDE WI-FI

O adaptador de Wi-Fi é *USB*, portanto basta inserí-lo em uma das quatro portas do *Raspberry Pi* que sua instalação é automática. No canto superior direito do monitor encontra-se o ícone de Wi-Fi caso a instalação tenha sido concluída com sucesso. Clicando nele, já é possível escolher uma rede e conectar-se nela.

Caso aconteça de não ocorrer a instalação automaticamente, existe a possibilidade de o adaptador não ser compatível com a placa. Em http://elinux.org/RPi_USB_Wi-Fi_Adapters há uma lista de adaptadores compatíveis para consulta.

4.3 CONFIGURANDO O BLUETOOTH

Assim como o Wi-Fi, o adaptador *bluetooth* também é *USB* e sua instalação se faz automática, só é preciso inserí-lo na porta. Deve-se clicar no símbolo do *bluetooth* que irá aparecer no canto superior direito da tela e selecionar o dispositivo de áudio para parear.

4.4 ESTABELECENDO CONEXÃO REMOTA

O protocolo RDP (*Remote Desktop Protocol*) é utilizado para estabelecer o acesso do computador ao *Raspberry Pi*. Não é preciso instalar programas novos no computador, pois o cliente já vem pré-instalado na maioria das versões do *Windows*. Para utilizá-lo no *Raspberry Pi*, deve-se instalar o cliente *xrdp* para que haja o acesso

do *desktop Linux*. No *menu*, selecionar o *terminal* encontrado em *accessories*. Ao executá-lo, digita-se o comando abaixo:

```
sudo apt-get install xrdp
```

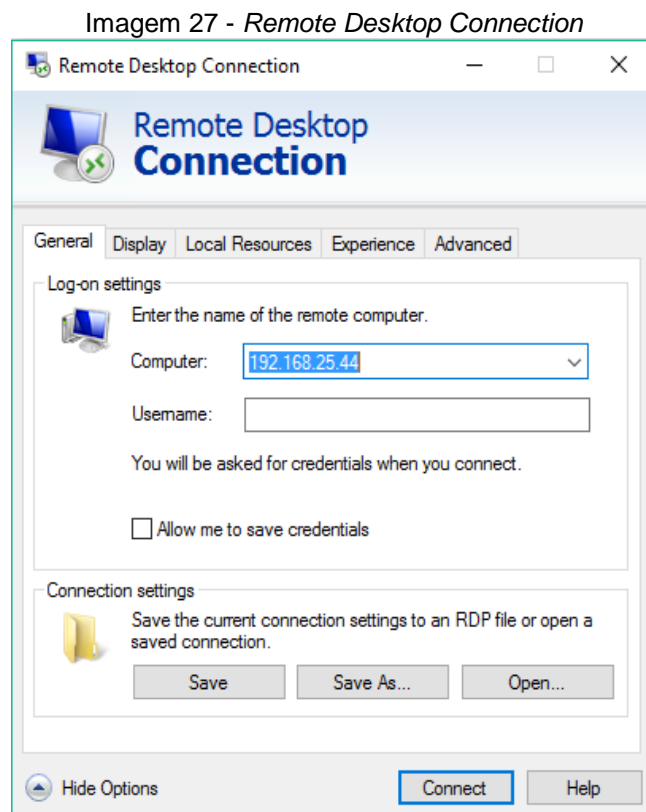
A instalação começará e terminará automaticamente.

O próximo passo se concentra na obtenção do dado fundamental para que haja a conexão: o endereço IP do *Raspberry Pi* na rede Wi-Fi. É indispensável que a placa esteja conectada na rede, como já configurado no passo 6.2. Há duas formas de encontrar o IP. A primeira é simples: digita-se no próprio terminal o seguinte comando:

```
hostname -i
```

Ou pode-se acessar as configurações do roteador no computador, que deve estar conectado no mesmo Wi-Fi ou por um cabo de rede. Abre-se qualquer navegador e digita-se o endereço do roteador, que pode variar de acordo com o modelo. Normalmente o IP estará localizado em uma área que lista os dispositivos conectados na rede.

Sabendo o endereço IP do *Raspberry Pi*, executa-se o programa “*Remote Desktop Connection*” no computador. Uma janela irá abrir solicitando alguns dados necessárias para acessar o dispositivo, como na Imagem 27:



Fonte: Elaborado pelo autor.

No campo *Computer* coloca-se o IP do *Raspberry*, e logo abaixo o usuário definido nas configurações iniciais da placa. A senha será exigida quando houver conexão. Em *Connection settings* é possível salvar as configurações, o que faz o programa criar um atalho no *desktop* do computador. Esta etapa permite um acesso rápido que dispensa o uso de monitor, mouse e teclado, facilitando o manuseamento e deixando o *Raspberry* com mais entradas livres.

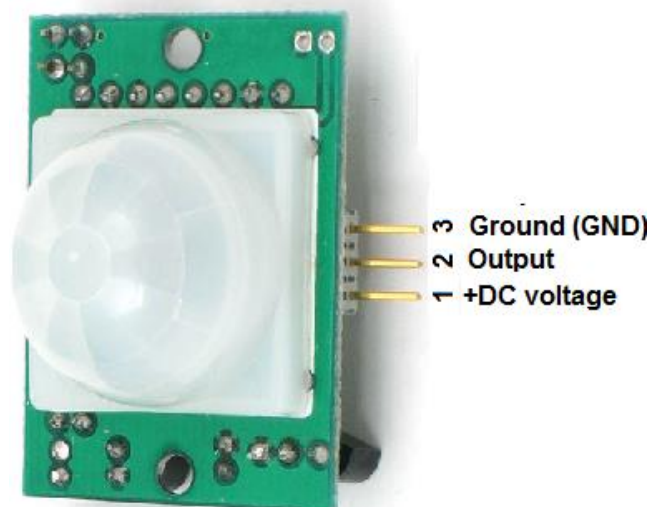
4.5 CRIAÇÃO DE SCRIPT

O próprio sistema operacional fornece uma área reservada para a programação em *python*. Em *menu*, há uma área chamada *programming*, onde pode se escolher entre *Python 2* ou *3*. Foi escolhido o *Python 2* para os *scripts* desse projeto pois o *Python 3* é mais recente, possuindo menos bibliotecas que a versão 2. Ao selecioná-lo, um *prompt* irá abrir, permitindo o usuário digitar seu programa.

4.6 CONFIGURAÇÃO DO SENSOR

O sensor possui três pinos que devem ser conectadas no GPIO do *Raspberry Pi*. Como na Imagem 28, o pino central 2 deve ser conectada em alguma das portas de entrada e saída, enquanto o pino 1 deve ser ligado na entrada de 5 volts e o pino 3 no terra. A porta GPIO escolhida foi a 4.

Imagem 28 - Sensor PIR

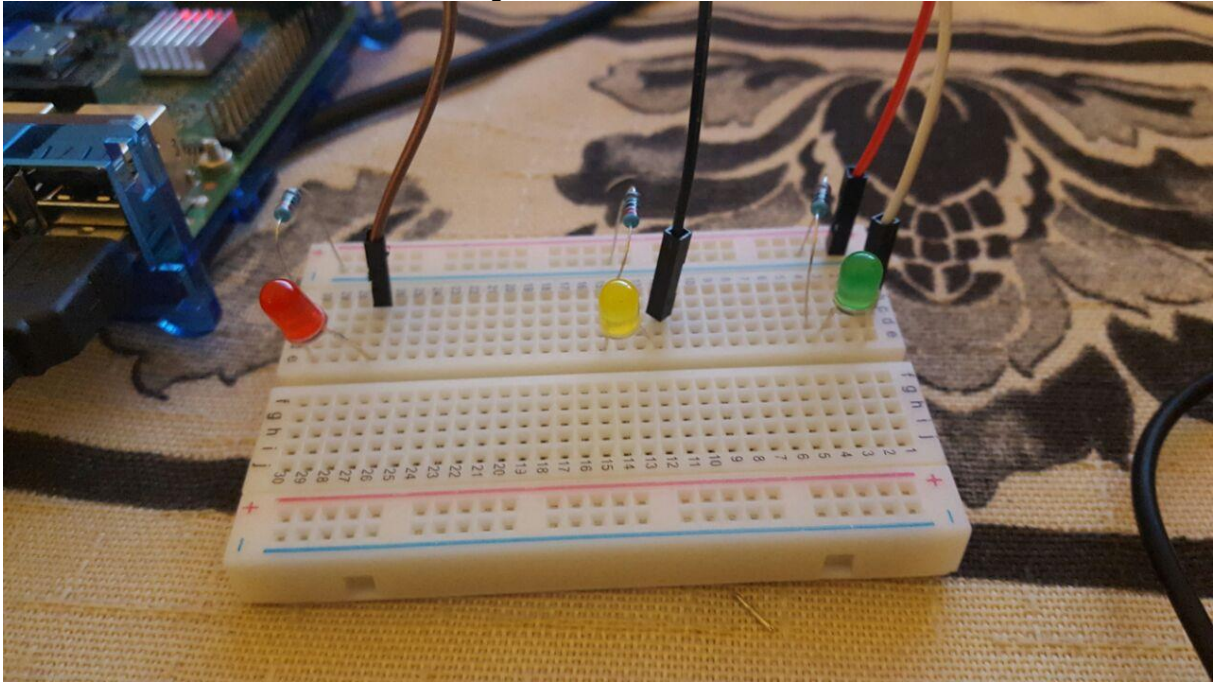


Fonte: <http://www.learningaboutelectronics.com/images/PIR-motion-sensor-pinout.png>, sem data.

4.7 MONTAGEM DA PLACA DE TESTE

A placa de teste inicialmente foi montada com o intuito de testar o funcionamento do sensor de presença. Três LEDs foram conectados como mostra a Imagem 29:

Imagem 29 - Placa de teste



Fonte: Elaborado pelo autor.

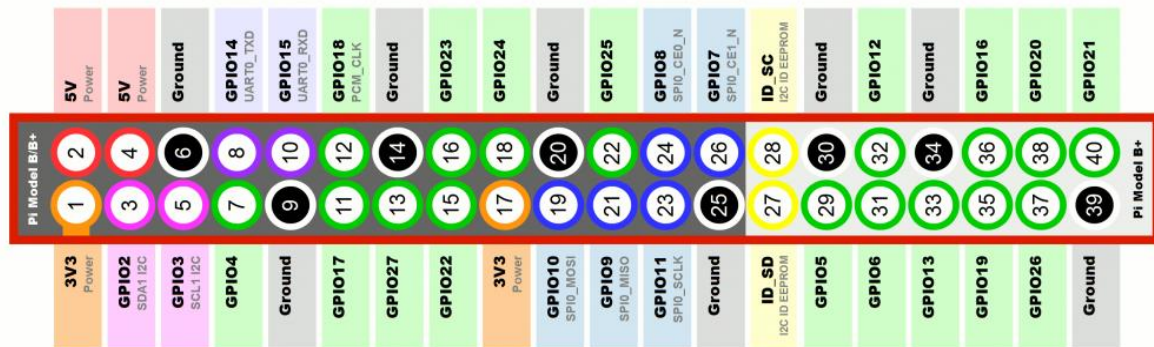
Os LEDs possuem um terminal menor chamado cátodo, que se conecta no GND (terra), e um maior chamado ânodo, que foi conectado em uma entrada GPIO configurada para saída (*Output*). Se a saída for *True* (ou *High*) o LED acende, se for *False* (ou *Low*) ele apaga. Entre o GND e o cátodo, foi colocado um resistor de 220 *Ohms* para não queimar os LEDs, já que a voltagem do *Raspberry* é 5 volts. A seguir, o *script* de teste que foi rodado:


```

1  import RPi.GPIO as GPIO
2  import time
3
4  GPIO.setmode(GPIO.BCM)
5
6  GPIO_PIR = 4
7  GPIO_GREEN = 17
8  GPIO_RED = 18
9  GPIO_YELLOW = 22
10
11
12 GPIO.setup(GPIO_PIR,GPIO.IN)
13 GPIO.setup(GPIO_RED, GPIO.OUT, initial = False)
14 GPIO.setup(GPIO_GREEN, GPIO.OUT, initial = False)
15 GPIO.setup(GPIO_YELLOW, GPIO.OUT, initial = False)
16
17 atual = 0
18 anterior = 0
19
20 try:
21
22     while GPIO.input(GPIO_PIR)==1:
23         GPIO.output(GPIO_YELLOW, GPIO.HIGH)
24         atual = 0
25         GPIO.output(GPIO_RED, GPIO.HIGH)
26         GPIO.output(GPIO_GREEN, GPIO.LOW)
27         GPIO.output(GPIO_YELLOW, GPIO.LOW)
28         while True :
29             atual = GPIO.input(GPIO_PIR)
30
31             if atual==1 and anterior==0:
32                 anterior=1
33                 GPIO.output(GPIO_RED, GPIO.LOW)
34                 GPIO.output(GPIO_GREEN, GPIO.HIGH)
35
36             elif atual==0 and anterior==1:
37                 GPIO.output(GPIO_YELLOW, GPIO.HIGH)
38                 anterior=0
39                 GPIO.output(GPIO_RED, GPIO.HIGH)
40                 GPIO.output(GPIO_GREEN, GPIO.LOW)
41                 time.sleep(1)
42                 GPIO.output(GPIO_YELLOW, GPIO.LOW)
43
44             time.sleep(0.01)
45
46 except KeyboardInterrupt:
47     GPIO.cleanup()

```

Primeiramente, nas linhas 1 e 2, a biblioteca *Rpi.GPIO* foi utilizada para o programa detectar quais portas do *Raspberry Pi* serão utilizadas. A *time* serviu apenas para dar um comando *time.sleep* (linha 44), que é um intervalo de tempo em que o sistema interrompe a execução do programa por um determinado tempo em segundos. A quarta linha é uma referência para os números dos pinos da *Broadcom SOC channel*, os que estão presentes no *Raspberry Pi*, como mostra a Imagem 30:

Imagem 30 - GPIO do *Raspberry Pi 2*

Fonte: <http://www.raspberrypi-spy.co.uk/wp-content/uploads/2012/06/Raspberry-Pi-GPIO-Layout-Model-B-Plus-rotated-2700x900-1024x341.png>, 2012.

O pino de entrada do sensor foi ligado no GPIO 4 (representado pelo círculo verde com o número 7 no diagrama), seu pino de VCC (alimentação do *proto-board*) pode ser conectado no círculo de número 2 ou 4, e o seu terra pode ser ligado em qualquer círculo preto determinado por *ground*. O LED vermelho foi ligado no GPIO 18, o amarelo no GPIO 22 e o verde no GPIO 17. No *script*, deve-se estabelecer as entradas escolhidas de acordo com as linhas 6 a 9.

As configurações de entrada (*input*) e saída (*output*) são declaradas nas linhas 12 a 15 com o comando *GPIO.setup*. Os parâmetros são: primeiro o dispositivo (LED ou sensor), depois se é *input* ou *output* e o último só funciona caso seja *output*, para determinar se começa com *True* ou *False*.

Nas próximas linhas, as variáveis “atual” e “anterior” servem para controlar o disparo do sensor. Ambas começam com zero. A combinação do *Try* e *Except* funciona como um método de parada do *script*. Tudo o que estiver depois do *Try* será rodado até que o comando CTRL + C encerra a execução devido ao *KeyboardInterrupt*. O *GPIO.cleanup()* da última linha limpa todas as portas utilizadas no *script* rodado.

No primeiro *while*, enquanto o sensor estiver captando presença, a variável “atual” fica zero e a luz amarela fica acesa. Assim, ao ligar o sistema, ele precisa de um tempo para não detectar mais movimento e começar. Quando o sensor não detectar nenhum movimento, a luz vermelha acende e o sistema prossegue.

O *While True* na linha 28 fica constantemente checando a entrada do sensor e armazena no “atual”. Se o “atual” for 1 (ou seja, o sensor captou movimento) e o anterior for zero, “anterior” recebe 1 e a luz verde acende indicando que o sensor foi acionado. Se o atual do sensor for zero (sem movimento) e o anterior for 1, o amarelo

acende para indicar uma transição de fase, o anterior recebe zero para poder entrar no *if* da linha 31 e logo em seguida a luz vermelha acende indicando que não há movimento. O *time.sleep(1)* da linha 44 serve apenas para dar um pequeno intervalo de tempo até que a luz amarela se apague.

O “atual” e o “anterior” garantem que um ciclo seja estabelecido, sendo que para detectar movimento é preciso que ele não tenha detectado antes, e vice-versa. Eles também, futuramente, evitam o envio de inúmeras mensagens repetidas no *Telegram*.

A ideia inicial do projeto não incluía os LEDs. Conforme o sucesso dos testes, foi decidido integrá-los como uma forma de notificar o usuário visualmente sobre o estado do sistema.

4.8 PROBLEMAS COM O YOWSUP

O *Yowsup* é uma biblioteca em *python* para o aplicativo *Whatsapp*. Ele seria usado para enviar mensagens ao usuário alertando sobre disparos no sensor de presença, porém o próprio *Whatsapp* bloqueia mensagens repetidas e enviadas constantemente, considerando-as como *spam*. A saída encontrada foi utilizar o *Telegram*, um aplicativo que possui ferramentas que permitem o desenvolvimento de BOTs.

4.9 INTEGRAÇÃO DO SISTEMA COM O TELEGRAM

O *BotFather* é um BOT (diminutivo de *Robot*, Robô em português) que auxilia na criação de outros BOTs. Para começar, deve-se acessar o site <https://telegram.me/BotFather>. Existem duas opções para começar a utilizá-lo: enviando uma mensagem no seu *Telegram* pelo celular ou através do *Telegram Web* (acesso do *Telegram* no navegador do computador). Ao escolher a plataforma que irá acessar, abrirá uma mensagem informando sobre as utilidades do BOT com um botão “começar”. Clicando nele, um *chat* com o nome “*BotFather*” irá abrir em suas mensagens com diversas instruções. Para criar um novo BOT, basta digitar no *chat* “/newbot”, e ele então pedirá um nome que deve terminar com “bot”. Feito isso, o *BotFather* irá mandar um *token* de acesso muito importante para utilizar na API do

Telegram, além de um *link*: “*telegram.me/(nome)*”. Clicando nele, o *chat* do seu *bot* abrirá, e pressionando o botão *começar*, o BOT inicia.

É preciso descobrir o número de identificação do *chat* (*chat_id*) da pessoa que irá receber as mensagens do BOT. Para encontrá-lo, primeiro deve-se enviar uma mensagem qualquer para o BOT. Depois, utiliza-se o seguinte endereço: “*https://api.telegram.org/bot<TOKEN>/getUpdates*” em qualquer navegador, substituindo o *<TOKEN>* pelo descoberto no *BotFather*. O retorno será algumas linhas de código como o descrito na Imagem 31.

Imagem 31 - *getUpdates* da API do *Telegram*

```
{"ok":true,"result":[{"update id":355550588,
"message":{"message_id":3,"from":{"id":<NUMERO_DO_CHAT_ID>,"first_name":<SEU_NOME>,"username":<SEU_USUÁRIO>},"chat":
{"id":<NUMERO_DO_CHAT_ID>,"first_name":<SEU_NOME>,"username":<SEU_USUÁRIO>,"type":"private"},"date":1481699616,"text":"dgdg"}]}}
```

Fonte: Elaborado pelo autor.

O *<NUMERO_DO_CHAT_ID>* será o identificador para que a mensagem seja entregue no número do seu *Telegram*. O “*text*” no final é a mensagem que foi enviada para o Bot.

Agora que as informações necessárias já foram coletadas, basta utilizar no código em *python* o trecho abaixo que envia a mensagem desejada para o *Telegram*:

```
import urllib
import urllib2

bot_id = "<TOKEN>"
url_telegram = "https://api.telegram.org/bot"

result = urllib2.urlopen(url_telegram + bot_id + "/sendMessage",
urllib.urlencode({ "chat_id": <NUMERO_DO_CHAT_ID>,
"text": '<TEXTO>' })).read()
```

Utilizando as bibliotecas *urllib* e *urllib2*, substituindo o *<TOKEN>* pelo gerado no *BotFather* e o *<NUMERO_DO_CHAT_ID>* pelo número identificador, basta escrever a mensagem que irá enviar no *<TEXTO>*.

4.10 IMPLEMENTAÇÃO DO BOTÃO RELATÓRIO

Na placa de teste, além dos LEDs, um botão foi implementado com o intuito de gerar um relatório e enviar para o *Telegram* BOT. O relatório informa a quantidade de vezes que o sensor foi acionado e o tempo decorrido desde a inicialização do sistema. O trecho de código abaixo faz os cálculos necessários para o tempo decorrido e soma o total de vezes que o sensor detectou movimento:

```

1  import math
2  import datetime
3
4  GPIO_BUTTON = 23
5  DisparoSensor = 0
6  now = time.time()
7
8  GPIO.setup(GPIO_BUTTON, GPIO.IN)
9
10 try:
11     [...]
12
13     while True :
14         [...]
15         if Current_State==1 and Previous_State==0:
16             [...]
17             DisparoSensor = DisparoSensor + 1
18             [...]
19
20         if GPIO.input(GPIO_BUTTON)== 0:
21             end = time.time() - now
22
23             s = end
24             s = int(s)
25             m,s = divmod(s,60)
26             h,m = divmod(m,60)
27
28
29             result = urllib2.urlopen("https://api.telegram.org/bot" +
"<TOKEN>" + "/sendMessage", urllib.urlencode({ "chat_id":
<NUMERO_DO_CHAT_ID>, "text": '\xF0\x9F\x94\xB2 Relatório \n
\xE2\x97\xBE Tempo decorrido: ' + str("%02d:%02d:%02d" % (h, m, s)) +
'\n \xE2\x97\xBE O sensor foi acionado ' + str(DisparoSensor) + '
vez(es).' })).read()
30
31         time.sleep(0.01)
32
33 except KeyboardInterrupt:
34     [...]
```

A biblioteca *math* é utilizada para a execução da operação *divmod* nas linhas 25 e 26, enquanto a *datetime* inclui todos os manuseamentos que envolvam tempo, como nas linhas 6 e 21. A entrada do botão no GPIO é definido na linha 4. A variável “DisparaSensor” armazena o número de vezes que o sensor foi acionado, como mostra a soma na linha 17, dentro da condição de disparo. A variável “now” recebe o

`time.time()`, que é um número *float* grande que representa os anos, os dias, as horas, os minutos, os segundos e milissegundos que passaram desde uma data pré-definida do computador (*Epoch*). No *Unix*, a data é primeiro de janeiro de 1970, com zero horas, zero minutos, zero segundos e zero milissegundos. O “*now*” marca o momento em que a aplicação começou a ser executada. Quando o botão é pressionado, faz-se rodar as condições do *if* na linha 20. A variável “*end*” invoca novamente o `time.time()`, que será o número representando o momento em que foi pressionado o botão, e o subtrai pelo “*now*”, que foi o tempo determinado logo que o sistema iniciou. O resultado é um número em segundos do tempo que decorreu desde o início do programa até o apertar do botão. Nas linhas subsequentes, primeiro o número é convertido para inteiro, excluindo assim as casas decimais a fim de utilizar o `divmod`, que divide o tempo em horas, minutos e segundos, e depois são armazenadas nas variáveis “*h*”, “*m*” e “*s*”, respectivamente. Ambos os dados são enviados para o *Telegram* através da linha 29.

4.11 IMPLEMENTAÇÃO DO EFEITO SONORO

O efeito sonoro ao disparar o sensor foi implementado com a biblioteca *pygame*. O som escolhido é de domínio público e foi obtido em um site que disponibiliza inúmeros efeitos sonoros. Pequenos trechos de código foram adicionados para que o som pudesse ser executado:

```

1  import pygame
2  [...]
3  try:
4  [...]
5
6  while True :
7
8      if Current_State==1 and Previous_State==0:
9          [...]
10
11         pygame.mixer.init()
12         pygame.mixer.music.load("sonzinho.mp3")
13         pygame.mixer.music.play()
14         while pygame.mixer.music.get_busy() == True:
15             continue
16
17     [...]
```

A linha 11 inicia o módulo do *mixer*, possibilitando o carregamento e a reprodução de sons. Logo abaixo, o som é carregado e preparado para execução. Dentro do parênteses deve-se colocar o nome do arquivo de som, e ele deve estar na mesma

pasta que o *script*. Na linha seguinte, o arquivo é reproduzido. A condição da linha 14 garante que a música será reproduzida inteira e ininterruptamente.

4.12 MONTAGEM DO SISTEMA

A aparência do projeto tornou-se rústica ao longo do processo. Com o intuito de aprimorar sua apresentação, uma caixa de madeira foi adotada para conter todas as partes do trabalho. Em sua superfície, duas aberturas iguais conseguem exibir o conteúdo da caixa. Em uma delas, foi colocado um vidro de forma que o *Raspberry Pi* pudesse ser exibido. O outro orifício permanece aberto com a finalidade de haver interação com o botão. Um pedaço de madeira pequeno foi cortado em medidas precisas para alocar os três LEDs e o botão, formando uma espécie de painel. Inicialmente o sensor ficaria junto com a placa, porém seu funcionamento com base em infravermelho foi interferido pelo vidro. Assim, com as sobras do pedaço de madeira, foi construído um pequeno envoltório para o sensor se manter fixado acima da caixa, entre os orifícios. O *Raspberry Pi* foi preso com quatro parafusos. Dois furos foram abertos para que os cabos da fonte e do HDMI pudessem ser conectados na placa. Por fim, a caixa recebeu um acabamento de tinta, e o sistema final pode ser observado na Imagem 32, logo abaixo.

Imagem 32 - Sistema Final



Fonte: Elaborado pelo autor.

5 CONCLUSÃO

O projeto atingiu os seus objetivos, mesmo adotando métodos diferente do planejado inicialmente.

O sensor utilizado não possui um alcance grande, porém o suficiente para o protótipo desenvolvido. Para captar movimentos em um ambiente fechado, como em uma sala, deve-se utilizar um sensor mais potente.

O *Telegram* BOT foi uma alternativa muito eficaz e pertinente para substituir o *Yowsup*, pois programas extras não precisaram ser instalados, operando somente por recursos *Web* no próprio código em *python*. Além disso, este impasse permitiu que meus conhecimentos fossem ampliados.

A inclusão da placa de teste deu um efeito positivo em questão de o usuário saber o que está acontecendo no processo de execução do projeto. Sem essa notificação visual, o trabalho poderia ter ficado mais técnico e menos acessível para a comunidade em geral.

O efeito sonoro também foi um detalhe que contribuiu para o aprimoramento do trabalho, dado que quanto mais notificações externas, melhor é a impressão do sistema final com relação à sociedade.

Posteriormente, pode-se incluir uma câmera no sistema de segurança com o auxílio de um monitor para mostrar as imagens gravadas, tornando-o mais completo. Além disso, é possível criar um executável para fazer o código rodar logo que o *Raspberry Pi* é ligado, evitando a conexão remota, deixando o sistema mais direto e focado em sua funcionalidade.

REFERÊNCIAS

ASHTON, K. That 'Internet of Things' Thing. **RFID Journal**, p. 1, 2009.

BURRUS, D. The Internet of Things Is Far Bigger Than Anyone Realizes. **Wired**, Novembro 2014. Disponível em: <<https://www.wired.com/insights/2014/11/the-internet-of-things-bigger>>. Acesso em: 19 Maio 2016.

CASTELLS, M. **A Galáxia Internet**: reflexões sobre a Internet, negócios e a sociedade. Oxford: Jorje Zaha Editor, 2001.

DACOSTA, F. **Rethinking The Internet Of Things**. Santa Clara: Apress Media, 2013.

DEBIAN. Sobre o Debian. **Debian**, 05 Julho 2016. Disponível em: <<https://www.debian.org/intro/about>>. Acesso em: 19 Maio 2016.

ENDEAVOR BRASIL. Tudo o que você precisa saber sobre internet das coisas. **Endeavor Brasil**, 18 Junho 2015. Disponível em: <<https://endeavor.org.br/internet-das-coisas/>>. Acesso em: 2016 Março 2016.

GALAL, T. Yowsup 2. **GitHub**, 22 Maio 2016. Disponível em: <<https://github.com/tgalal/yowsup/blob/master/README.md>>. Acesso em: 19 Maio 2016.

JORNAL DE BRASÍLIA. Assaltos em residências do DF aumentam 58% em fevereiro. **Jornal de Brasília**, 08 Março 2016. Disponível em: <<http://www.jornaldebrasil.com.br/CIDADES/ASSALTOS-EM-RESIDENCIAS-DO-DF-AUMENTAM-58-EM-FEVEREIRO/>>. Acesso em: 21 Agosto 2016.

MICROSOFT. Learn about Windows 10 IoT Core. **Microsoft**. Disponível em: <<https://developer.microsoft.com/pt-br/windows/iot/Explore/IoTCore>>. Acesso em: 21 Maio 2016.

OPENELEC. What is OpenELEC? **OpenELEC**. Disponível em: <<http://openelec.tv/home/what-is-openelec>>. Acesso em: 21 Maio 2016.

OSMC. About. **OSMC**. Disponível em: <<https://osmc.tv/about/>>. Acesso em: 21 Maio 2016.

PINET TEAM. Getting started with PiNet. **PiNet**. Disponível em: <http://pinet.org.uk/articles/installation/getting_started.html>. Acesso em: 21 Maio 2016.

RASPBERRY PI FOUNDATION. GETTING STARTED WITH YOUR RASPBERRY PI, Caldecote. Disponível em: <<https://www.raspberrypi.org/help/videos/>>. Acesso em: 18 Maio 2016.

RASPBERRY PI. RASPBERRY PI 1 MODEL A. **Raspberry Pi**. Disponível em: <<https://www.raspberrypi.org/products/model-a/>>. Acesso em: 18 Maio 2016.

RASPBERRY PI. RASPBERRY PI 1 MODEL A+. **Raspberry Pi**. Disponível em: <<https://www.raspberrypi.org/products/model-a-plus/>>. Acesso em: 18 Maio 2016.

RASPBERRY PI. RASPBERRY PI 1 MODEL B. **Raspberry Pi**. Disponível em: <<https://www.raspberrypi.org/products/model-b/>>. Acesso em: 18 Maio 2016.

RASPBERRY PI. RASPBERRY PI 1 MODEL B+. **Raspberry Pi**. Disponível em: <<https://www.raspberrypi.org/products/model-b-plus/>>. Acesso em: 18 Maio 2016.

RASPBERRY PI. RASPBERRY PI 2 MODEL B. **Raspberry Pi**. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>>. Acesso em: 18 Maio 2016.

RASPBERRY PI. RASPBERRY PI 3 MODEL B. **Raspberry Pi**. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>>. Acesso em: 18 Maio 2016.

RASPBERRY PI. RASPBERRY PI ZERO. **Raspberry Pi**. Disponível em: <<https://www.raspberrypi.org/products/pi-zero/>>. Acesso em: 18 Maio 2016.

RASPBIAN. Welcome to Raspbian. **Raspbian**. Disponível em: <<https://www.raspbian.org/>>. Acesso em: 19 Maio 2016.

RIBEIRO, L. Morumbi registra 272 assaltos a residências entre janeiro e outubro. **Fiquem Sabendo**, 28 Dezembro 2015. Disponível em: <<http://www.fiquemsabendo.com.br/2015/12/morumbi-registra-272-assaltos-a-residencias-entre-janeiro-e-outubro/>>. Acesso em: 18 Agosto 2016.

RICHARDSON, M.; WALLACE, S. **Getting Started with Raspberry Pi**. Sebastopol: Maker Media, 2012-2013.

RISC OS. Welcome to RISC OS Pi. **RISC OS**, 05 Setembro 2015. Disponível em: <<https://www.riscosopen.org/wiki/documentation/show/Welcome%20to%20RISC%20OS%20Pi>>. Acesso em: 21 Maio 2016.

TAIT, T. F. C. Evolução da Internet: do início secreto à explosão mundial. **Informativo PET Informática**, p. 2, 2007.

THE CENTRE FOR COMPUTING HISTORY. One of the First Raspberry Pi Computers Donated to Museum. **The Centre for Computing History**, 01 Setembro 2012. Disponível em: <<http://www.computinghistory.org.uk/news/16944/>>. Acesso em: 18 Maio 2016.

TORVALDS, L. Linus Torvalds: Linux succeeded thanks to selfishness and trust. **BBC News**, 13 Junho 2012. Disponível em: <<http://www.bbc.com/news/technology-18419231>>. Acesso em: 15 Julho 2016.

UBUNTU MATE TEAM. Ubuntu MATE for the Raspberry Pi 2 and Raspberry Pi 3. **Ubuntu MATE**. Disponível em: <<https://ubuntu-mate.org/raspberry-pi/>>. Acesso em: 21 Maio 2016.

UBUNTU. Ubuntu for the Internet of Things. **ubuntu**. Disponível em: <<https://www.ubuntu.com/internet-of-things>>. Acesso em: 21 Maio 2016.

UPTON, E. PRICE CUT! RASPBERRY PI MODEL B+ NOW ONLY \$25. **Raspberry Pi**, 14 Maio 2015. Disponível em: <<https://www.raspberrypi.org/blog/price-cut-raspberry-pi-model-b-now-only-25/>>. Acesso em: 18 Maio 2016.

UPTON, E. Blog Raspberry Pi. **Raspberry Pi**. Disponível em: <<https://www.raspberrypi.org/blog/author/eben/>>. Acesso em: 18 Maio 2016.

VERMESAN, O.; FRIESS, P. **Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems**. Aalborg: River Publishers, 2013.

WEISER, M. Scientific American Ubicomp Paper after Sci Am editing. **Ubiq**, Setembro 1991. Disponível em: <<http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>>. Acesso em: 2016 Julho 16.

WENDLING, M. Sensores. **UNESP - Faculdade de Engenharia de Guaratinguetá**, Guaratinguetá, 2010. Disponível em: <<http://www2.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4---sensores-v2.0.pdf>>. Acesso em: 12 Maio 2016.

WHATSAPP. Sobre o WhatsApp. **WhatsApp**. Disponível em: <<https://www.whatsapp.com/about/>>. Acesso em: 19 Maio 2016.

XIA, F. et al. Internet of Things. **Wiley Online Library**, p. 3, 2012.

ZUIM, E. Resistores Não Lineares. **Eze Website**. Disponível em: <http://www.ezuim.com/pdf/res_nl.pdf>. Acesso em: 03 Julho 2016.