



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

VINÍCIUS DE LUCA BUENO

**IOT COM MICROCOMPUTADORES: CONTROLE E SIMPLICIDADE PARA
CASAS INTELIGENTES**

**Assis/SP
2018**



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

VINÍCIUS DE LUCA BUENO

**IOT COM MICROCOMPUTADORES: CONTROLE E SIMPLICIDADE PARA
CASAS INTELIGENTES**

Projeto de pesquisa apresentado ao curso de Ciências da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

**Orientando(a): Vinícius de Luca Bueno
Orientador(a): Douglas Sanches da Cunha**

**Assis/SP
2018**

FICHA CATALOGRÁFICA

BUENO, Vinícius de Luca.

IOT COM MICROCOMPUTADORES: CONTROLE E SIMPLICIDADE PARA CASAS INTELIGENTES / Vinícius de Luca Bueno. Fundação Educacional do Município de Assis –FEMA – Assis, 2018.

53.

1. Internet das coisas. 2. Raspberry Pi. 3. Casas inteligentes.

CDD: 005.304
Biblioteca da FEMA

IOT COM MICROCOMPUTADORES: CONTROLE E SIMPLICIDADE PARA CASAS INTELIGENTES

VINÍCIUS DE LUCA BUENO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____
Prof. Douglas Sanches da Cunha

Examinador: _____
Prof. Fernando Cesar de Lima

DEDICATÓRIA

Dedico este trabalho à minha família que me estimulou e incentivou durante toda a criação do projeto, aos meus amigos que participaram de maneira direta ou indireta à na criação e progresso do projeto, e ao meu orientador por abrir alas para as escolhas dos caminhos em momentos difíceis.

Resumo: Este artigo tem como objetivo mostrar a montagem e aplicação de um modelo de *Internet* das coisas a fim de realizar a automação de casas inteligentes focado na mudança de estado de *switches* elétricos com o auxílio das tecnologias das placas programáveis Raspberry Pi e Arduino, extraíndo das mesmas benefícios e ferramentas que cada uma pode oferecer, além de realizar a integração de ambas de forma a se comunicarem através de sinais elétricos compreendidos pela programação interna dos mesmos e transmitirem os dados captados por sensores, por toda a aplicação, podendo desta forma haver a recepção e controle de tais dados por meio de uma aplicação da *internet*, tal qual também possibilitará o usuário realizar a interação com os dispositivos de alteração de estado do *switch* elétrico através do acesso remoto à aplicação.

Palavras-chave: Internet das coisas, Raspberry Pi, Casas Inteligentes, Automação, Acesso Remoto.

Abstract: This article aims to show the installation and application for an Internet of things template in order to accomplish a smart home automation focused on electrical switches state changes supported by programmable boards Raspberry Pi and Arduino technologies, taking advantage of its available tools and values, in addition to integrate both in order to communicate through electrical signals comprised by their internal script code and transmit it's sensors collected data throughout the application, thus allowing reception and control of these data across an internet application, as it will also enable the user to interact directly with electric switch devices through a remote access application.

Keywords: Internet of things, Raspberry Pi, Smart homes, Automation, Remote access.

LISTA DE ILUSTRAÇÕES

Figura 1: Quantidade total de casas inteligentes, em milhões (América do Norte e Europa 2012-2017). (VUJOVIC´ V, MAKSIMOVIC´ M, 2015, p.2).	13
Figura 2: Previsão de crescimento da IoT(1988-2020).(IoT Online Store, 2016).....	17
Figura 3: 10 aplicações mais populares de IoT. (IoT Online Store, 2016).	19
Figura 4: Arquitetura <i>Middleware</i> de IoT baseado em SOA. (WANG <i>et al</i> ,2015).	20
Figura 5: Raspberry Pi 1 modelo B+. (Raspberry Pi Foundation, 2018).	23
Figura 6: Raspberry Pi 3 modelo B. (Raspberry Pi Foundation, 2018).	24
Figura 7: Modelo representativo de <i>protoboard</i> . (Wikipédia, 2018).	25
Figura 8: Modelo representativo do sensor LDR. (Cactus.io, 2018).	26
Figura 9: Logomarca do Raspbian OS. (Raspberry Pi Foundation, 2018).....	28
Figura 10: Arduino Uno (Arduino.cc, 2018).....	31
Figura 11: Funcionalidades do Arduino Uno (SOUZA, Fabio. 2018).	32
Figura 12: Tela inicial da IDE Arduino (feita pelo autor, 2018).....	33
Figura 13: Tela de inicialização do NOOBS. (Raspberry Pi Foundation, 2018).....	34
Figura 14: Comando para atualização do sistema. (Feita pelo autor via SSH, 2018).....	35
Figura 15: Comando para atualização de pacotes. (Feita pelo autor via SSH, 2018).	35
Figura 16: Configuração de IP estático. (Feita pelo autor via SSH, 2018).....	36
Figura 17: Ativação de porta serial da GPIO. (Feita pelo autor via SSH, 2018).	37
Figura 18: Consulta de portas seriais. (Feita pelo autor via SSH, 2018).	38
Figura 19: Remover console. (Feita pelo autor via SSH, 2018).....	38
Figura 20: Reativar Bluetooth. (Feita pelo autor via SSH, 2018).	39
Figura 21: Verificação das portas seriais atuais. (Feita pelo autor via SSH, 2018).	39
Figura 22: Prototipação dos componentes físicos. (Feita pelo autor via Fritzing, 2018). ...	40
Figura 23: Fração da página teste do servidor. (Feita pelo autor, 2018).	41

Figura 24: Fração da página teste do PHP. (Feita pelo autor, 2018).....	42
Figura 25: Código de automação no Arduino. (Feita pelo autor, 2018).	43
Figura 26: Código de controle manual no Arduino. (Feita pelo autor, 2018).....	44
Figura 27: Requisição de funcionamento automático. (Feita pelo autor, 2018).....	45
Figura 28: Requisição de funcionamento manual (Feita pelo autor, 2018).....	45
Figura 29: Trecho HTML/PHP da Interface gráfica. (Feita pelo autor, 2018).....	46
Figura 30: Interface gráfica para o usuário. (Feita pelo autor, 2018).	46

LISTA ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i> (interface de Programação de Aplicação)
ARM	<i>Advanced RISC Machine</i> (máquina RISC avançada)
CPU	<i>Central Process Unit</i> (Unidade central de processamento)
DNS	<i>Domain Name System</i> (sistema de nome de domínios)
DDNS	<i>Dynamic DNS</i> (DNS dinâmico)
EEPROM	Electrically-Eraseble Programmable Read-Only Memory
FULL HD	<i>Full high definition</i> (alta definição total)
GND	<i>Ground</i> (pino negativo ou “terra”)
GPIO	<i>General purpose input/output</i> (entradas/saídas de propósitos gerais)
HDMI	<i>High-definition multimedia interface</i> (interface multimídia de alta definição)
HTML	<i>Hypertext Markup Language</i> (linguagem de marcação de hipertexto)
IOT	<i>Internet of Things</i> (internet das coisas)
LDR	<i>Light Dependent Resistor</i> (resistor com dependência de luz)
LED	<i>Light Emissor Diode</i> (diodo emissor de luz)
NOOBS	<i>New Out Of Box Software</i>
NoSQL	<i>Not Only SQL</i> (não apenas SQL)
OS	<i>Operational System</i> (sistema operacional)
PHP	PHP: <i>Hypertext Processor</i> (PHP: processador de hipertexto)
RAM	<i>Random access memory</i> (memória de acesso aleatório)
RISC	<i>Reduced Instruction Set Computer</i> (computador de instruções reduzidas)
RJ-45	<i>Registered Pack 45</i> (pacote de registro 45)
RX	<i>Reception</i> (recepção de dados)
SD	<i>Secure digital</i>

SOA	<i>Service-oriented Architecture</i> (arquitetura orientada a serviços)
SSH	<i>Secure Shell</i>
SQL	<i>Structured Query Language</i> (linguagem de consulta estruturada)
TX	<i>Transmission</i> (transmissão de dados)
URL	<i>Uniform Resource Locator</i> (localizador uniforme de recursos)
USB	<i>Universal serial bus</i> (barramento serial universal)

SUMÁRIO

INTRODUÇÃO.....	13
1. INTERNET DAS COISAS.....	17
1.1. FUNCIONAMENTO DA INTERNET DAS COISAS.....	19
1.2. IOT INTEGRADO AO MICRO CONTROLADOR RASPBERRY PI	21
2. RASPBERRY PI.....	22
2.1. RASPBERRY PI 3 MODELO B.....	23
2.2. COMPONENTES EXTERNOS DO RASPBERRY PI.....	25
2.3. SOFTWARES PARA O RASPBERRY PI.....	27
3. ARDUINO.....	30
3.1. ARDUINO UNO.....	30
3.2. ARDUINO IDE.....	32
4. CONFIGURAÇÃO DOS COMPONENTES.....	34
4.1. CONFIGURAÇÃO DO RASPBERRY PI.....	34
4.1.1. CONFIGURAÇÃO DE PORTAS SERIAIS.....	36
4.2. MONTAGEM DOS COMPONENTES FÍSICOS.....	39
4.3. CONFIGURAÇÃO DO SERVIDOR.....	41
5. IMPLEMENTAÇÃO.....	43
5.1. COMUNICAÇÃO ENTRE COMPONENTES FÍSICOS.....	43
CONCLUSÃO.....	47
REFERÊNCIAS.....	49

INTRODUÇÃO

No mundo atual encontra-se em uma busca cada vez maior de facilidades e comodidades para a nossa vida com o intuito de reduzir custos e o tempo gasto para coisas simples do nosso cotidiano de forma a poder também realizar tarefas à distância, por meio da internet, tornando essas situações mais práticas. Com isso, o sistema de Internet das Coisas vem mostrando um aumento notável nos últimos anos, por conta de seu alto dinamismo além da variedade de componentes que são capazes de se comunicar e interagir com o usuário (Vujovic´ V, Maksimovic´ M.). Outro recurso dinâmico muito utilizado para inúmeras áreas tanto da computação quanto da robotização é o Raspberry Pi. Este componente trata-se de um micro controlador com diversos periféricos de entrada e saída de dados, resolução de vídeo, conexão à internet, além da capacidade de inserir outros acessórios como sensores, *displays*, além de inúmeros componentes específicos para determinada área de aplicação, tornando-o um elemento de grande utilidade na elaboração de um sistema utilizando Internet das coisas. A figura 1 relata esse considerável aumento de *home automation* nos últimos anos.

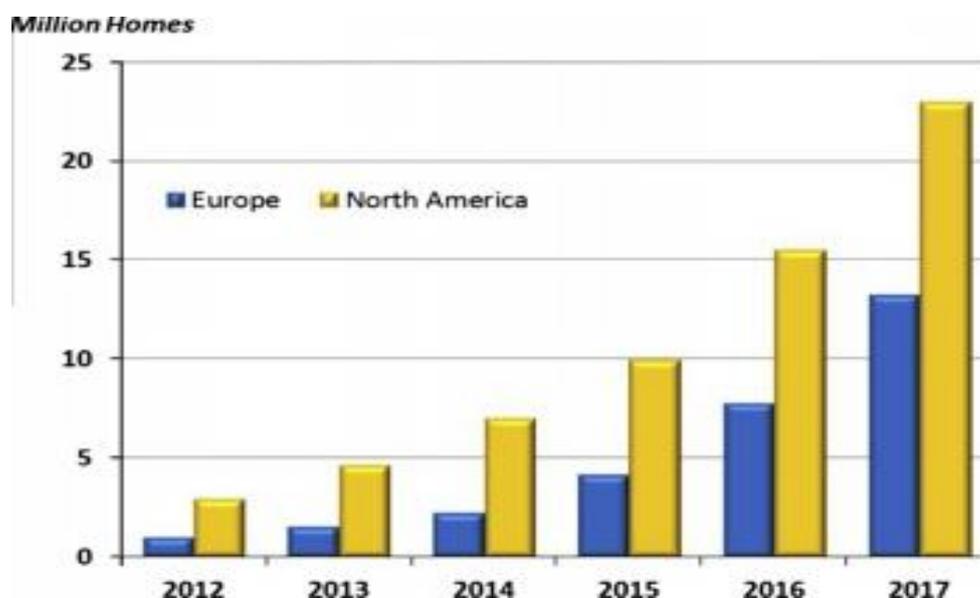


Figura 1: Quantidade total de casas inteligentes, em milhões (América do Norte e Europa 2012-2017). (VUJOVIC´ V, MAKSIMOVIC´ M, 2015, p.2)¹.

¹ Disponível em <<http://www.sciencedirect.com/science/article/pii/S0045790615000257>>. Acesso em 27 out. 2017.

Diversos projetos foram criados com o intuito de controlar e manipular objetos a distância, sendo um deles o projeto de detecção de movimento utilizando Raspberry Pi, criado por Ansari, Sedkyl, Sharma e Tyagi (2013) ao qual realiza diversas fotografias quando é captado alguma alteração no ambiente, salvando-as internamente no Raspberry Pi e enviando-as por meio de uma conexão cabeada (RJ-45) para um servidor externo.

Com esse microcomputador é possível também realizar o monitoramento da saúde do usuário, como Gupta, Patchava e Menezes (2015), que programaram o Raspberry Pi com sensores capazes de mostrar em tempo real o batimento cardíaco do usuário do equipamento, podendo ser monitorado pessoalmente por um display ou em qualquer lugar no mundo pela internet.

Para diminuir o custo de um sistema de gerenciamento de casas inteligentes, é necessário o uso de máquinas de custo inferior, porém que possuam o mesmo poder computacional e maior dinamismo na elaboração dos projetos. Por isso o uso do micro controlador Raspberry Pi que possui infinitas utilidades, por meio de acessórios de prototipação, captação de dados e programação, facilitando todo o processo.

O objetivo do projeto é realizar um protótipo utilizando Raspberry Pi para o controle de dispositivos eletrônicos (*switches/interruptores*) em residências, com captação de dados e acesso remoto por meio de uma aplicação na *internet*, para desta forma possibilitar saber se há necessidade de algum desses dispositivos estarem ligados, ativando-os automaticamente, ou se algum deles estiver em uso, ser possível realizar o gerenciamento da aplicação, além do controle da mesma (ligar/desligar independentemente do nível de luminosidade captado). Com isso, é possível interagir com o dispositivo podendo supervisioná-lo à distância e também ativá-lo a qualquer momento. Tal aplicação também transmite os dados em tempo real, que poderão ser acessados pelo usuário através da interface *web* (criada nas linguagens HTML/PHP) dentro do servidor Apache (Raspberry Pi), ao qual serão transmitidas por meio dos códigos Python, somente após ter realizado seu *login* no sistema e obtiver conexão com a *Internet*.

A aplicação será simples e possuirá as opções básicas para a manipulação dos recursos instalados fisicamente ao Raspberry Pi, além dos próprios segmentos eletrônicos (interruptores de energia, *switches on/off*, etc) e para o monitoramento e alteração de estado dos segmentos.

A aplicação em Python servirá como interface entre os segmentos acoplados ao Raspberry Pi com o dispositivo do usuário final. Tal linguagem foi escolhida por conta de sua portabilidade tanto com projetos IoT quanto ao próprio Raspberry Pi, ao qual está acoplada ao sistema operacional Raspbian, ao qual será utilizado no desenvolvimento do projeto.

Para a realização da modelagem poderá ser utilizado outras ferramentas auxiliares, como adaptadores de sinal (digital para analógico), a placa Arduino, outra placa Raspberry Pi, adaptadores de vídeo, etc., de forma a facilitar o procedimento.

Portanto, o desenvolvimento de um projeto de Internet das Coisas é muito proveitoso para o contexto geral do aluno, visto que se refere a uma tecnologia atual e que vem crescendo exponencialmente nos últimos anos, além do uso de programação *web*, *hardwares* programáveis e uso de banco de dados, tornando-o um projeto completo, possibilitando assim o aluno a utilizar do conhecimento adquirido em futuras aplicações.

A motivação dada foi pelo fato de poder utilizar das tecnologias do Raspberry Pi, brevemente visto em aula. Nota-se que se trata de uma tecnologia muito dinâmica e apropriada para inúmeros tipos de projeto, além disso, combina com a tecnologia crescente da Internet das Coisas, que está em busca cada vez maior de ideias e planejamentos de projetos, tornando-a ainda mais presente no cotidiano das pessoas.

O conceito desta pesquisa é de reduzir o custo e facilitar o uso de ferramentas que possibilitem o acesso remoto e automático de componentes em sua própria casa, por um baixo custo de peças e manutenção, visto que uma placa programável com sensores possui um custo muito inferior ao de sistemas de automatização prontos, além de possuir o dinamismo de poder ser reprogramados. Também se deve ao fator de que esta área está crescendo muito e tende a ficar cada vez mais comum no nosso dia-a-dia, tornando-se uma área com necessidade maior de pesquisas e ideias, havendo assim grande potencial na área.

O desenvolvimento do projeto será realizado por etapas, sendo cada uma delas acompanhadas pelo orientador pessoalmente ou via *e-mail*, e desta forma o mesmo poderá auxiliar com ideias e material didático para melhor construção do projeto. Um grupo de estudo será realizado entre os alunos com o intuito de uma aprendizagem aprofundada sobre o funcionamento dos conceitos do Raspberry Pi, simplificando o uso do equipamento, suas ferramentas e seus acessórios.

Nos capítulos subsequentes será possível observar uma abordagem mais profunda dos temas utilizados, seus conceitos e maneiras de aplicação, desta forma realizar a interação ao leitor sobre IoT, aplicada principalmente no conceito de *home automation* e a iniciação com as placas micro controladoras Raspberry Pi 3 e Arduino Uno (configurações de comunicação, montagem dos recursos físicos, maneiras de utilização), suas formas de funcionamento e suas determinadas ocupações dentro de uma aplicação de acesso remoto. Posteriormente será exemplificada uma aplicação tal qual desempenha funções, métodos e tecnologias relatadas durante o processo, além das configurações necessárias para o funcionamento geral da aplicação, ferramentas utilizadas e iniciação das plataformas (sistema operacional Raspbian Jessie, interface de comandos do Raspbian, atuação do Raspberry Pi como um servidor, etc.) de maneira didática, formando uma base para futuras alterações e segmentação de pesquisa/aplicação.

1. INTERNET DAS COISAS

A Internet das Coisas, habitualmente conhecido por IoT, trata-se da utilização de internet em recursos do nosso cotidiano, ou seja, a computação designada às coisas e lugares. Ela atua a partir de um determinado problema, e por meio de seus recursos de coleta, transferência e armazenamento de dados pode propor alternativa ou realizar tarefas de maneira que o problema inicialmente dado seja diminuído ou eliminado por meio de uma aplicação de acesso remoto.

Esta tecnologia vem crescendo nos últimos anos, como podemos ver na figura 2, o que induz cada vez mais à necessidade do conhecimento sobre as utilizações e aplicações da mesma.

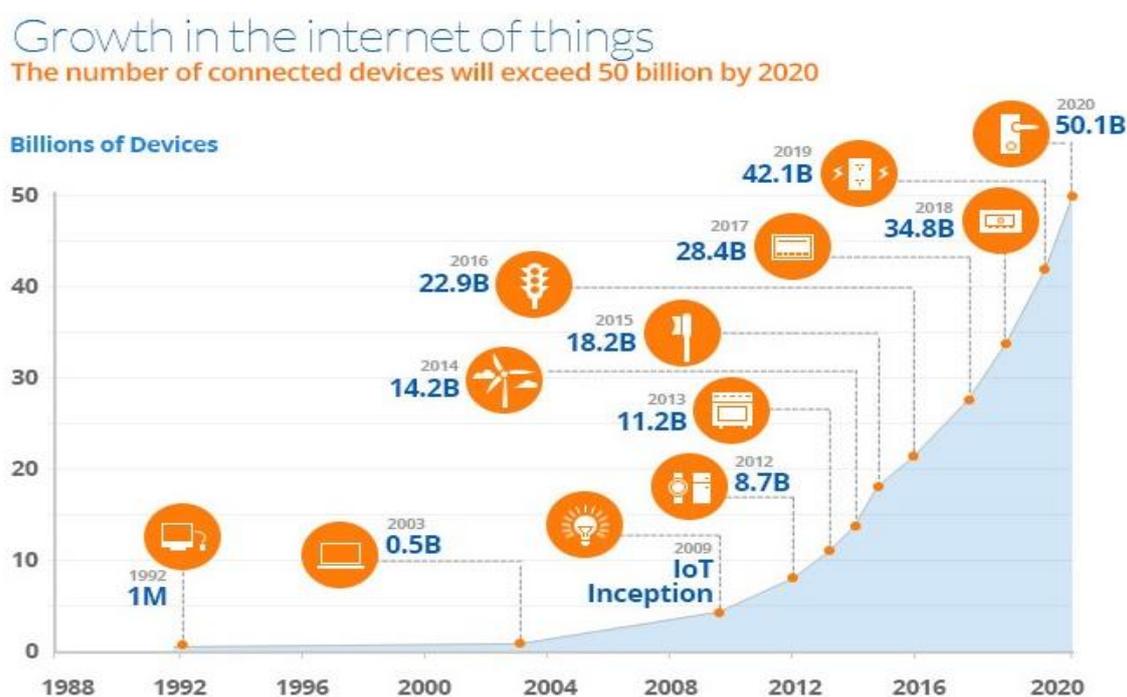


Figura 2: Previsão de crescimento da IoT(1988-2020).(IoT Online Store, 2016)².

² Disponível em <http://www.iotonlinestore.com>. Acesso em 03 mar. 2018.

Com isso a IoT tornou-se um tema de muita discussão e estudos na área da tecnologia nos últimos anos. Isso se atribui ao seu dinamismo e sua gama de possibilidades para resolver problemas da sociedade, assim como encontrar maneiras de facilitar a vida da população e de empresas que cada vez mais prezam pela agilidade, automação e produtividade. Por meio de aplicações que são acessadas remotamente é possível realizar monitoramentos corporais como o de pacientes com alguma doença ou desequilíbrio na saúde, ou de atletas, de forma que possam melhorar seu desempenho utilizando dispositivos transportáveis (*wearable devices*), assim como os métodos realizados por Dijkstra *et al* (2014), que aplicaram esses dispositivos para gerenciar tanto a saúde quanto o desempenho para atletas de alta *performance*. Há também utilização na área empresarial e industrial, que buscam soluções na redução de custos, ou no aumento da produtividade, gerando assim mais lucros. Alguns exemplos como o de Ożadowicz, Grela (2015), evidenciam a integração dos recursos de IoT para o gerenciamento de energia, que por meio de sensores coletores de dados e nós de rede encontra os problemas de consumo, podendo assim ser realizado a implementação da solução na mesma plataforma.

Outro recurso importante para quem procura a tecnologia de internet das coisas é o da automação de algum mecanismo, tanto no setor industrial quanto para cidades (*smart cities*) e casas inteligentes (*home automation*). Segundo Ansari, *et al* (2015), nesse aspecto a integração de aplicações com IoT deve-se a fatores como dinamismo na criação de projetos, que se desenvolvem de acordo com a necessidade de resolver um determinado problema, juntamente pelo baixo custo de realizar tais operações (levando em consideração outros métodos para resolver os mesmos tipos de problema), tanto na infraestrutura física utilizada quanto em suas aplicações e manutenções, além da praticidade que tal sistema trará a seu usuário.

Assim como se pode ver na figura 3, onde foi realizado uma análise de vendas das diversas aplicações IoT de um determinado *site* de vendas, constata-se então uma grande busca de produtos para o cotidiano, como os de *smart home* ou de *wearable devices*.

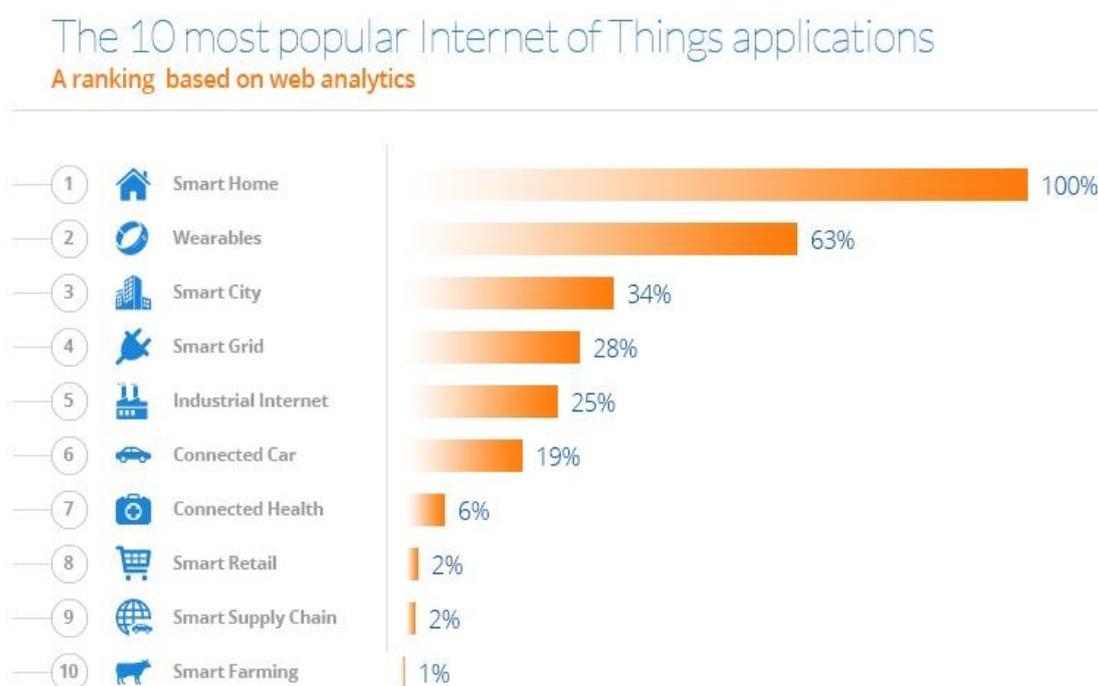


Figura 3: 10 aplicações mais populares de IoT. (IoT Online Store, 2016)³.

1.1. FUNCIONAMENTO DA INTERNET DAS COISAS

Como a Internet das coisas realmente funciona? A base de uma aplicação IoT constitui-se de um aplicativo capaz de controlar e gerenciar dados coletados por sensores ou aparelhos e armazenar as informações como resultado final, podendo posteriormente utilizá-los para outros fins. Contudo ainda falta um componente para que esse sistema funcione. O meio capaz de integrar, relacionar todos os protocolos e garantir segurança dos dispositivos utilizados é chamado de *Middleware*, sendo ele capaz de interagir com as múltiplas plataformas utilizadas em um sistema de IoT. Segundo Ngu, et al (2016), “diversos *middleware* de IoT e protocolos de conectividade estão sendo desenvolvidos, e o número aumenta a cada dia”, mostrando assim uma ampla multiplicidade de usos do

³ Disponível em <http://www.iotonlinestore.com>>. Acesso em 03 mar. 2018.

IoT e a necessidade de protocolos correspondentes. Também citado por Ngu, et al (2016), há duas categorias de projetos IoT: “coleta de dados do ambiente para análise e aplicações reativas em tempo real”; a primeira maneira, utilizada para análise futura dos dados coletados, de modo que os dados possam ser guardados na máquina de coleta (sem necessidade de internet) e posteriormente enviados ao servidor da aplicação, assim que obter conexão com a internet. Já a segunda maneira identifica-se pela resposta em tempo real, ao qual os dados são enviados automaticamente ao servidor da aplicação. Segundo Bandyopadhyay et al (2016), “o objetivo principal de todas as iniciativas de desenvolvimento de *Middleware* é de desenvolver um *framework* de forma a obter uma camada de adaptação *plug-n-play*”, onde possibilita utilizar a mesma plataforma *Middleware* em distintas aplicações IoT . Pode-se ver logo abaixo na figura 4 um exemplo das atividades exercidas pelo *Middleware* em uma arquitetura de IoT orientada a serviços (SOA).

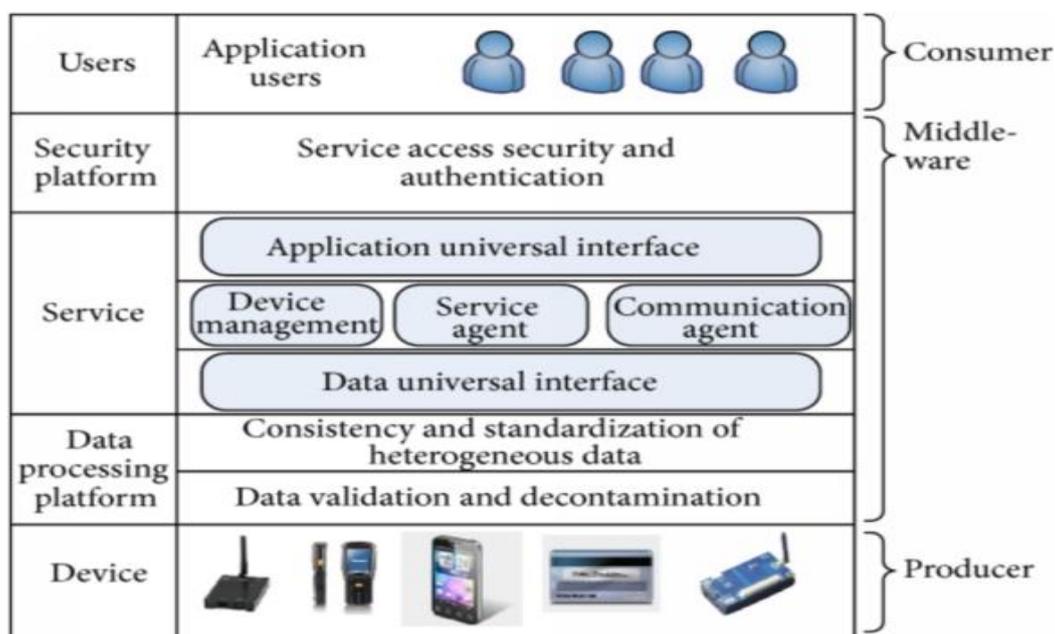


Figura 4: Arquitetura *Middleware* de IoT baseado em SOA. (WANG et al,2015)⁴.

⁴ Disponível em <<http://downloads.hindawi.com/journals/js/2015/827045.pdf>>. Acesso em 04 mar. 2018.

1.2. IOT INTEGRADO AO MICRO CONTROLADOR RASPBERRY PI

A partir dos conceitos anteriormente ditos, podem-se então realizar inúmeros projetos. O que facilita a realização de tais atividades é a variedade de micro controladores no mercado, com elevada capacidade de processamento da informação além da capacidade de captar a informação por meio de seus diversos tipos de sensores. Estes micro controladores são indispensáveis em um sistema IoT. Em especial, utiliza-se o micro controlador Raspberry Pi modelo 3, ao qual possui bom custo-benefício levando em conta seu poder de processamento, as múltiplas ferramentas ao qual poderá ser acoplada ao mesmo e o valor de mercado da placa.

Segundo Maksimovic et al (2014), o Raspberry Pi possui as vantagens de que “é pequeno, poderoso, baixo custo, hackeável e orientado a educação”, e ainda conclui que “possui preços acessíveis, sendo uma plataforma de interface perfeita para diversos dispositivos”.

Outros fatores como o desempenho (processadores ARM), memória RAM (podendo ultrapassar 1 gigabyte) e por possuir sistema operacional com interface gráfica (geralmente baseado em Linux), demonstra o porquê é utilizado em sistemas de pequeno (fins didáticos), médio e grande porte (aplicações empresariais). Outro aspecto importante é o de sua “pinagem” (GPIO), ao qual o torna capaz de realizar múltiplas tarefas, dependendo da necessidade dos usuários. Seu *hardware* também constitui de diversas entradas de periféricos, como o da fonte de alimentação, entrada serial universal (USB), dispositivo de som, entrada para cartão de memória *Secure Digital* (SD), entrada de vídeo digital (HDMI), além da antena para sinal de *internet wireless*, todos eles já acoplados à placa.

Como um todo, este *hardware* mostra-se flexível independente do software e ferramentas utilizadas para ou por ele, de modo a ser eficaz dentro de um sistema integrado à Internet das Coisas.

No capítulo a seguir demonstra modelos do micro controlador Raspberry Pi, suas capacidades físicas (*hardware*), modelos de aplicação, possíveis ferramentas físicas a serem utilizadas e a interação de programas/aplicações relacionadas ao funcionamento do controle dos demais componentes da plataforma.

2. RASPBERRY PI

Este *hardware* consiste em uma placa do tamanho de um cartão de crédito equipado com diversos conectores para múltiplas funcionalidades. Os primeiros Raspberry Pi foram lançados no início de 2012, tendo como principal fundamento o uso para ensino de computação em escolas por um baixo custo, podendo assim ser utilizados em países subdesenvolvidos ou em desenvolvimento (Raspberry Pi Foundation, 2018). A base desse micro controlador é seu processador da arquitetura ARM ao qual ainda utiliza nos modelos atuais, sendo este uma referência para outros micro controladores concorrentes ao Raspberry Pi.

O tamanho, poder de processamento e custo sempre foram aliados do Raspberry Pi, podendo então realizar tarefas que placas de menor poder computacional como o Arduino não seriam capazes. Relacionado a isso, diversos projetos de eletrônica, IoT, *cloud computing*, processamento paralelo, etc., são realizados com o mesmo aproveitando as qualidades anteriormente citadas, fazendo com que sua comunidade cresça a cada dia.

Os Raspberry Pi modelo A e B, que foram os primeiros lançados pela Fundação Raspberry (Raspberry Pi Foundation) foram substituídos pelos modelos A+ e B+ respectivamente, ao qual receberam uma melhoria no tamanho, economia no consumo de energia, aumento de entradas seriais e, sua maior mudança, o aumento de 26 para 40 pinos GPIO.

Atualmente os melhores modelos do Raspberry Pi disponíveis são o modelo Zero W, ao qual concilia a variedade de utilidades com o menor dos preços e o Raspberry Pi 3 modelo B, que possui o melhor desempenho, sendo este o mais utilizado em projetos em que se busca poder computacional, porém respeitando o limite do *hardware* utilizado (Raspberry Pi Foundation, 2018).

Tanto seu *hardware* quanto seu *software* é *open source*, ou seja, podem ser modificados, utilizados para estudos ou até mesmo comercializados, levando em conta que o produto final não seja especificamente o Raspberry Pi, e sim a aplicação ao qual ele foi integrado, além disso, os códigos fonte dos projetos, mapeamento de construção das placas e utilitários são disponibilizados pela própria fundação, de forma livre.

Na figura 5 apresenta uma placa Raspberry Pi 1 modelo B+, sendo esta a revisão final do Raspberry Pi original.

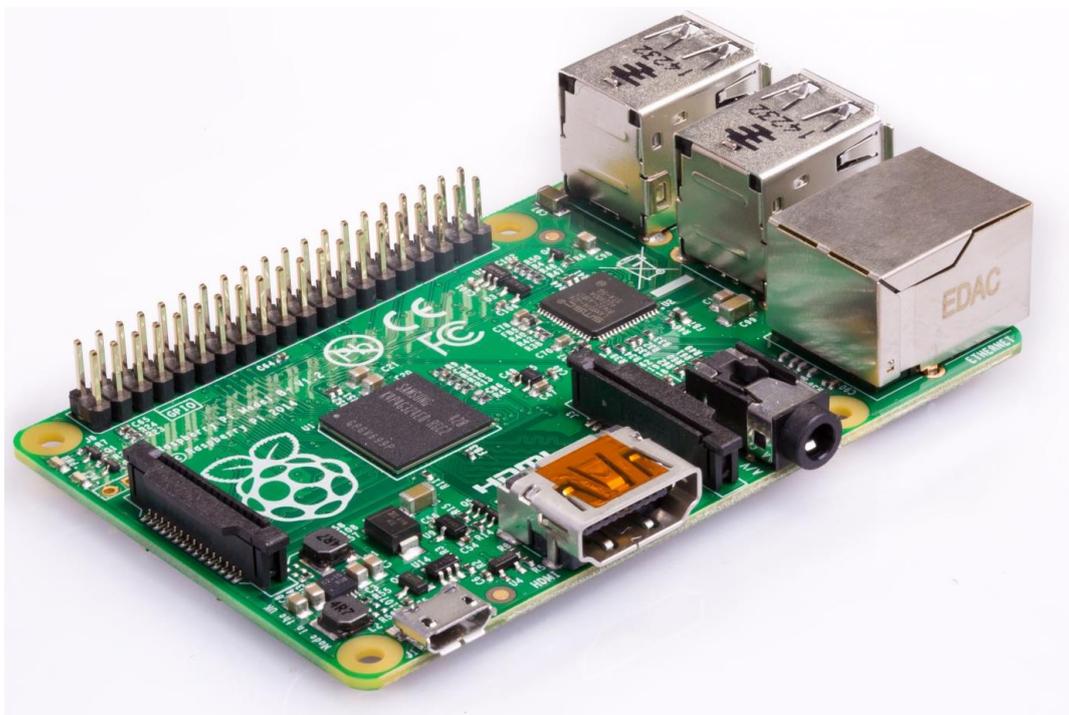


Figura 5: Raspberry Pi 1 modelo B+. (Raspberry Pi Foundation, 2018)⁵.

2.1. RASPBERRY PI 3 MODELO B

O Raspberry Pi 3 é o último modelo até então do micro controlador da Raspberry Pi *Foundation*. A variação deste modelo para o Raspberry Pi 2 constitui-se pelo aumento da velocidade de *clock* (pulso de envio de informações) e aumento na amperagem, dado pela mudança de sua fonte de alimentação para um barramento micro USB, desta forma obtendo maior desempenho em aplicações ao qual necessita um nível maior de operação da placa.

Abaixo estão as especificações do Raspberry Pi 3, (Raspberry Pi Foundation, 2018):

⁵ Disponível em < <https://www.raspberrypi.org/products/raspberry-pi-1-model-b-plus/> >. Acesso em 10 mar. 2018.

- CPU Broadcom BCM2837, de arquitetura ARM Cortex-A53 de 64Bits *Quad Core*, com 1.2GHz de velocidade de *clock*.
- 1GB de memória RAM.
- GPIO com 40 pinos de extensão.
- 4 portas USB 2.0.
- 4 portas de saída de vídeo analógicas *Pole Stereo*.
- Porta Ethernet 10/100 Mbit/s, conexão 802.11n *wireless*, Bluetooth versão 4.1.
- Porta HDMI *Full HD*.
- Porta de conexão para câmeras.
- Porta de conexão para *displays touchscreen*.
- Entrada para cartão Micro SD para armazenamento do sistema operacional e de dados.
- Fonte de alimentação Micro USB com fonte de energia de até 2,5A.

Na figura 6 abaixo, podemos ver um exemplo do Raspberry Pi 3 modelo B:

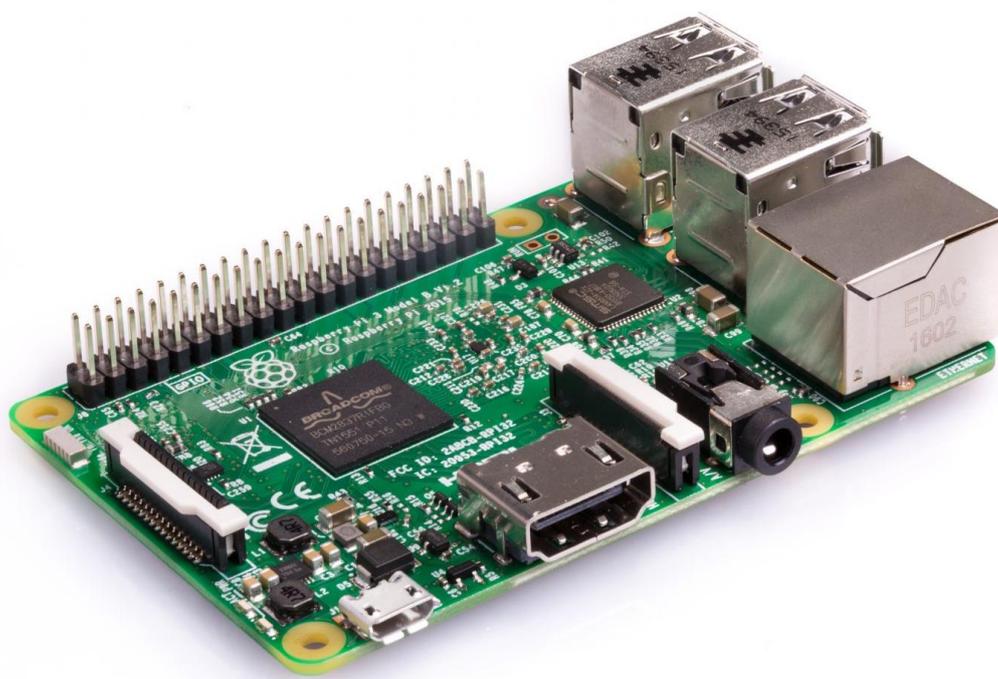


Figura 6: Raspberry Pi 3 modelo B. (Raspberry Pi Foundation, 2018)⁶.

⁶ Disponível em < <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>>. Acesso em 10 mar. 2018.

2.2. COMPONENTES EXTERNOS DO RASPBERRY PI

Tratando-se de componentes extras ao Raspberry Pi aos quais não fazem necessariamente parte de seu conjunto (não acoplados à placa), podemos citar a *protoboard* (ou placa de ensaio). Este componente pode ser integrado a micro controladores como o próprio Raspberry Pi, Arduino, etc., para a realização do desenvolvimento de projetos de eletrônica. No geral, ela consiste em uma placa com múltiplos furos que servem como barramento de tensão de cobre para circuitos⁷, variando seu uso para cada tipo de projeto, componentes acoplados e maneiras de ligação como os cabos *jumper* (condutor elétrico para placas integradas), *resistores* (dispositivo limitador de corrente elétrica) e capacitores de carga (componente armazenador de cargas elétricas de forma a proteger contra sobrecargas sobre um circuito elétrico), variando também o formato da protoboard, quantidade de furos de tensão e variações de voltagem de acordo com o modelo, fabricante e patente utilizada. Com ela pode-se realizar a conexão entre o Raspberry Pi (provedor de comandos, energia e sinais elétricos) para outros periféricos (circuitos, LEDs, *displays*, sensores), montando assim o esquema físico que será utilizado. Podemos observar a representação de um modelo de *protoboard* que possui 400 pontos de conexão, como mostra a figura 7.

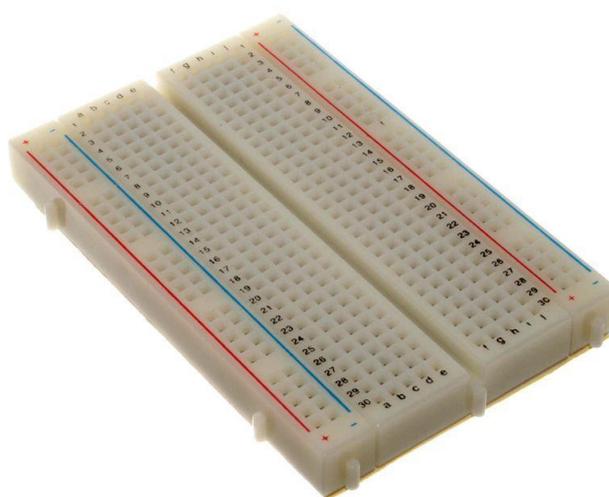


Figura 7: Modelo representativo de *protoboard*. (Wikipédia, 2018)⁸.

⁷ Disponível em <<http://tangentsoft.net/elec/breadboard.html>> Acesso em 11 jun. 2018.

⁸ Disponível em <https://en.wikipedia.org/wiki/Breadboard#/media/File:400_points_breadboard.jpg> Acesso em 11 jun. 2018.

Outro equipamento de ampla utilidade na montagem de um sistema de circuitos integrados é o relé. Este dispositivo possui como essência alterar os estados de uma integração, por meio da corrente elétrica que passa por ele de maneira eletromecânica (alterações de estado físico através de sinais elétricos). Com esse componente, torna-se mais simples a ativação/desativação de interruptores, auxiliando no processo de montagem e funcionamento da aplicação.

O sensor de luminosidade LDR (*Light Dependent Resistor*) provê a captação do nível de luminosidade do ambiente, realizando assim a identificação da necessidade de ativação da aplicação, sendo muito importante para aplicações automáticas ou em situações predefinidas. Abaixo podemos ver uma representação de tal sensor e sua montagem em um circuito, como mostra a figura 8.

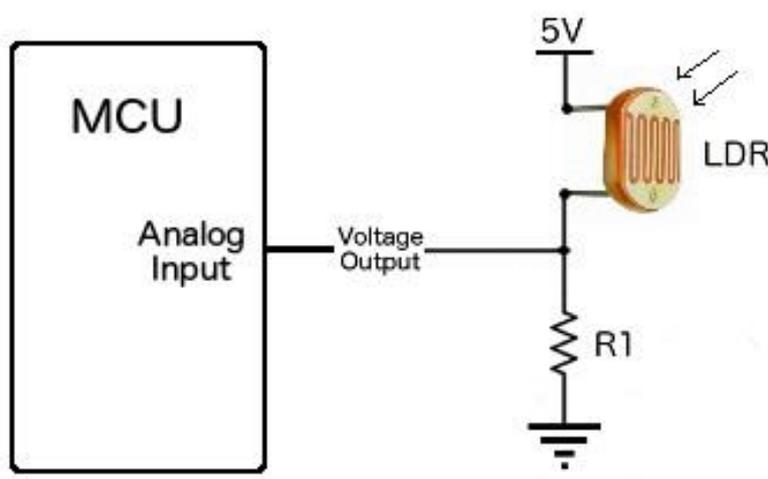


Figura 8: Modelo representativo do sensor LDR. (Cactus.io, 2018)⁹.

Há também os LEDs (*Light Emissor Diode*) ao qual possui como finalidade emitir um sinal de luz. Este sinal pode ser um recurso final ou apenas a representatividade de algum componente, que seja possível aplicar nesta mesma integração, mas por questões didáticas não são utilizadas.

Ao fim, utilizam-se outros componentes como forma de sintetizar a aplicação, de forma que seus recursos agreguem na modelagem da integração e supram a necessidade de uso para o fim específico proposto.

⁹ Disponível em <<http://cactus.io/hooks/sensors/light/ldr/hookup-arduino-to-ldr-sensor>>

Acesso em 13 jun. 2018.

2.3. SOFTWARES PARA O RASPBERRY PI

Com o lançamento do Raspberry Pi, foi necessário que houvesse um Software controlador para os segmentos da placa. Para isso a própria Raspberry Pi Foundation¹⁰, por meio de um projeto independente, lançou meses após sua placa, um sistema operacional com interface gráfica não somente usual, mas também otimizado para os componentes encontrados nos Raspberry Pi, sendo este conhecido como Raspbian (união de Raspberry com Debian), ao qual é baseado na plataforma de sistema operacional Linux, mais especificamente na distribuição Debian e com ele vieram diversas ferramentas e acessórios, em geral para seu uso inicial, o ensino de lógica computacional e informática. Posteriormente, diversos outros utilitários foram sendo desenvolvidos, não apenas pela fundação, mas também por desenvolvedores independentes (*software livre*), abrangendo ainda mais as possibilidades e utilidades de um Pi.

Atualmente existe uma infinidade de sistemas operacionais focados ao Raspberry Pi, para diversos interesses e finalidades, sendo a ampla maioria deles baseados em Linux, com uma pequena exceção ao Windows 10 para a plataforma, que possui como finalidade o uso em aplicações IoT. Há também um pacote de sistemas operacionais focado totalmente à inicialização na plataforma e ferramentas, sendo este o Noobs (Raspberry Pi Foundation, 2018). Através dele é possível realizar a instalação de um ou mais sistemas operacionais em um mesmo disco (*SD card*), também explana sobre a finalidade dos sistemas operacionais disponíveis, facilitando a escolha da plataforma aos usuários iniciantes. Todos esses sistemas operacionais tornam-se acessíveis pela interface de gerenciamento de *boot* conhecida como GRUB (*Grand Unified Bootloader*), sendo essa ferramenta proveniente do sistema operacional Linux e também inseridas nas configurações do Raspberry Pi (Raspberry Pi Foundation, 2018).

¹⁰ Disponível em <<https://www.raspberrypi.org>>. Acesso em 10 mar. 2018.

Abaixo na figura 9, podemos ver a logomarca do sistema operacional Raspbian, ao qual foi elaborado pela própria Raspberry Pi Foundation e será utilizada na realização do projeto.



Figura 9: Logomarca do Raspbian OS. (Raspberry Pi Foundation, 2018) ¹¹.

Valle et al (2017) mostra em seu artigo que seu Raspberry Pi “roda sob o sistema operacional Raspbian e seu código é desenvolvido na linguagem de programação Python para facilitar a automação de captura de imagem e análise”. Merchant e Ahire (2017) citam que “A programação geral do *software* é feita em Python, os sensores captam dados e são extraídos e armazenados por meio da aplicação”, mostrando assim ser válida a utilização de tal linguagem de programação para aplicação em sistemas integrados ao IoT, ao qual são necessários o uso de tais funcionalidades. O servidor Apache, tal qual será utilizado para a criação da aplicação *web* também servirá como meio de transmissão de dados e comunicação entre cliente/servidor, possibilitando o acesso remoto e inserção de meios de segurança de acesso aos dados do usuário (*tokens* de segurança, chaves de acesso, criptografia, etc.). Há também uma variedade de banco de dados disponíveis para as plataformas de Raspberry Pi juntamente com IoT, podendo eles seguir o padrão da linguagem SQL ou NoSQL, aos quais segundo Nayyar (2017), “há diversos fatores ao qual se deve levar em conta ao escolher um banco de dados para aplicações de IoT. As mais importantes são a escalabilidade (habilidade de lidar com enormes quantidades de dados em tempo adequado), código flexível, portabilidade com diversas ferramentas analíticas, segurança e custo”, sendo assim muito importante a escolha dos *softwares* para a criação de um projeto de Internet das Coisas.

¹¹ Disponível em <<https://www.raspberrypi.org/downloads/raspbian/>> Acesso em 10 mar. 2018.

No capítulo seguinte poderá ser observado a introdução sobre outra plataforma micro controladora de componentes físicos e circuitos eletrônicos denominada Arduino. A mesma tal como o Raspberry Pi é capaz de realizar o controle de componentes para IoT por meio de uma codificação predefinida. Ambas as placas podem realizar suas tarefas independentemente uma da outra, mas neste caso, ocorre uma coexistência e conectividade entre ambas, de modo a ser possível agregar novas funcionalidades e diversificar os meios de aplicação.

3. ARDUINO

O Arduino consiste em uma pequena placa de processamento de circuitos elétricos com foco tanto para *software* quanto para *hardware*. Ela realiza processos elétricos em conjunto com dispositivos físicos (*protoboard*, resistores, capacitores, *jumpers*, relés, LEDs, *displays*, botões, etc), além de também haver traços à área de robótica e afins. Para realizar tais operações, o Arduino executa informações inseridas no mesmo através de instruções dadas a partir da plataforma de programação Arduino IDE, ao qual controlará os dispositivos unidos no projeto através das portas seriais do Arduino (Arduino, 2018) ¹². Seguindo deste princípio, é possível realizar inúmeras aplicações utilizando várias ferramentas e para distintas finalidades, inclusive projetos relacionados à IoT, bastando a acoplagem de placas expansivas (*shields*) de uso específico ao requerido para a realização do esquema projetado, dando assim funcionalidades extras à base inicial de um Arduino comum¹³ ou até mesmo a conexão com outra placa com capacidade de transmissão remota (*internet/bluetooth*).

Podemos observar mais especificações embutidas em uma placa Arduino a seguir, assim como as implantações de código em seu *firmware* dado através da IDE Arduino e seu funcionamento como um todo em aplicações do meio de circuitos eletrônicos.

3.1. ARDUINO UNO

O Arduino Uno foi o primeiro modelo implantado de uma extensa série de micro controladores (que possui o mesmo nome) em forma definitiva para mercado. O baixo custo e a múltipla funcionalidade do mesmo chama a atenção em usos tanto para fins didáticos quanto para uso particular ou até mesmo de mercado. As especificações do Arduino apresentam o uso da CPU ATMEGA16U2, um dispositivo de 8 bits da arquitetura RISC onde opera com 32Kb de memória flash, 2Kb de memória RAM e 1Kb de memória EEPROM¹⁴, que opera em uma frequência de 16MHz podendo alcançar até 20MHz.

¹² Disponível em <<https://www.arduino.cc/en/Guide/Introduction>> Acesso em 01 set. 2018.

¹³ Disponível em <<https://learn.sparkfun.com/tutorials/arduino-shields>> Acesso em 01 set. 2018.

¹⁴ Disponível em <<https://www.embarcados.com.br/arduino-uno/>> Acesso em 02 set. 2018.

Pode-se atuar em tensões baixas de até 1.8V (*volts*), mas sua frequência de transmissão não passará dos 4MHz sendo, portanto, a tensão ideal para o funcionamento da placa aproximadamente 5V. Abaixo se pode ver a representação do Arduino Uno, relatado na figura 10.

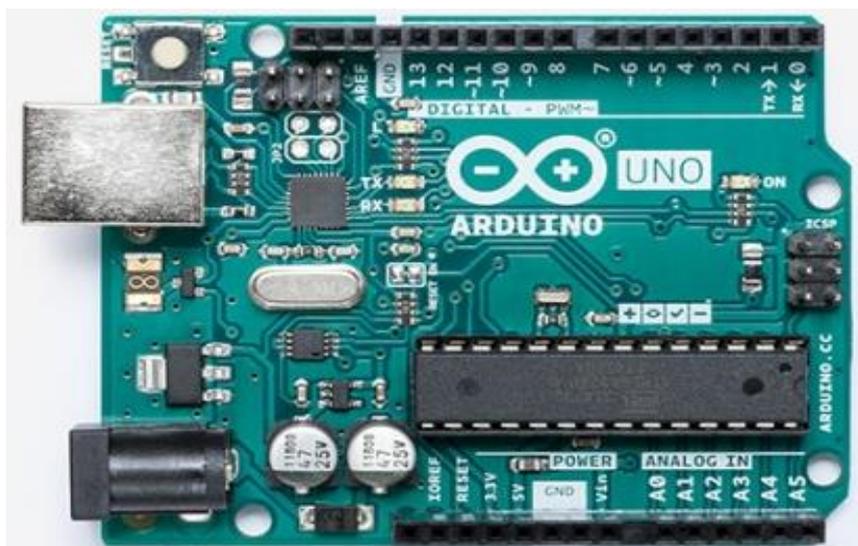


Figura 10: Arduino Uno (Arduino.cc, 2018)¹⁵.

A pinagem do Arduino consiste em *Inputs* (entradas) e *outputs* (saídas) de dados que operam a tensão de 5V em uma corrente de 40mA (miliampere), tais quais 6 destes pinos transmitem sinal analógico e outros 14 pinos possuem transmissão digital, sendo 6 destes capazes de prover pulsos PWM (*pulse width modulation*), ou seja, pinos capazes de modular a potência e velocidade por meio de ondas “quadradas”¹⁶. Há também pinos especiais provedores de energia, podendo haver transmissão de 3.3V, 5V ou GND (carga negativa), que podem suprir outros equipamentos acoplados ao mesmo.

¹⁵ Disponível em <<https://store.arduino.cc/usa/arduino-uno-rev3>> Acesso em 02 set. 2018.

¹⁶ Disponível em <http://www.mecaweb.com.br/eletronica/content/e_pwm> Acesso em 02 set. 2018.

A figura 11 mostra o resumo geral das funcionalidades propostas de um Arduino Uno, algumas de pouca relevância para um projeto relacionado a casas inteligentes, mas cujo no contexto geral devem ser citadas.

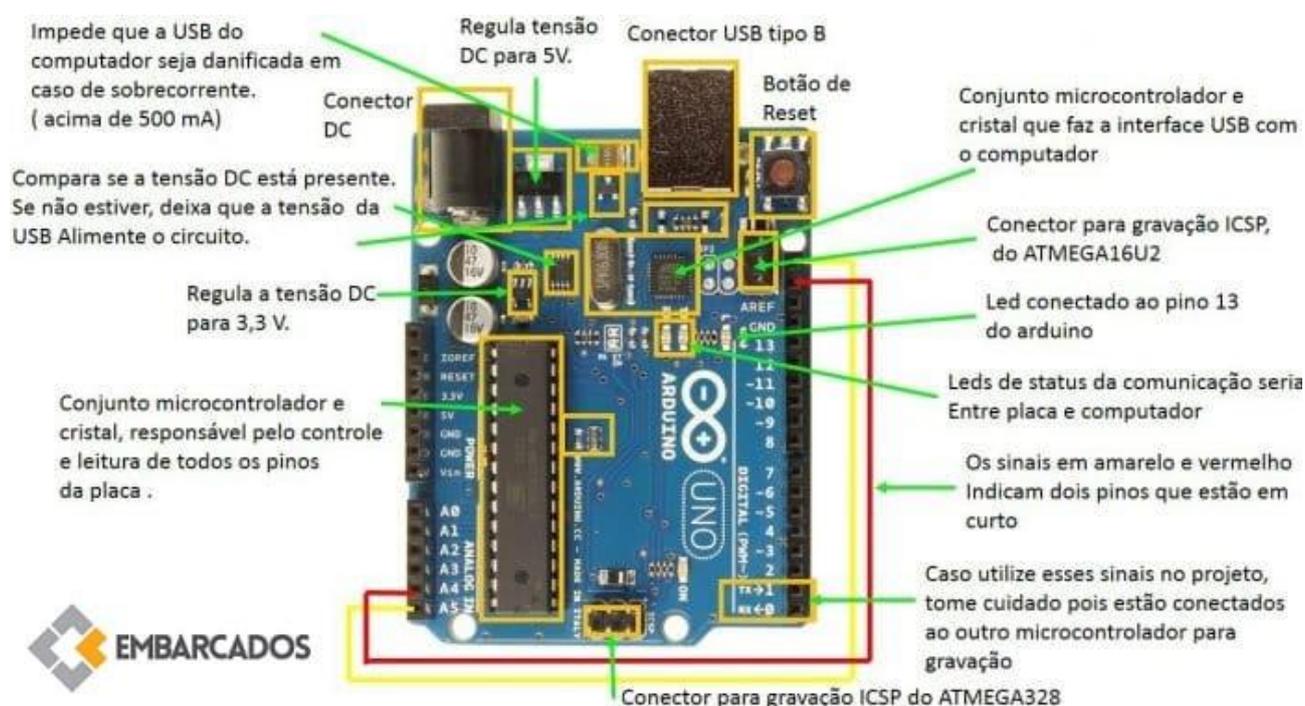


Figura 11: Funcionalidades do Arduino Uno (SOUZA, Fabio. 2018) ¹⁷.

3.2. ARDUINO IDE

A Arduino IDE trata-se da plataforma de programação (*sketch*) para as placas Arduino de uso em múltiplos sistemas operacionais, tal onde se desenvolve o *script* de controle dos componentes físicos. Sua programação possui funções únicas ao qual lida com *loops*, funcionamento dos componentes, transmissão de porta serial, frequência de transmissão de corrente, nível de luminosidade, etc.¹⁸, que serão interpretadas pelo micro controlador interno do Arduino. Outra curiosidade é que o código gerado pela IDE é inserido no

¹⁷ Disponível em <<https://www.embarcados.com.br/arduino-uno/>> Acesso em 02 set. 2018.

¹⁸ Disponível em <<https://www.arduino.cc/en/Guide/Environment>> Acesso em 03 set. 2018.

núcleo do mesmo, diretamente em seu *firmware*, desta forma há a possibilidade de guardar apenas a funcionalidade (*script*) atual do mesmo para utilizações futuras.

As linguagens de programação propostas a ela são C++ e Java, ambas com versões próprias para desenvolvimento da plataforma e possui inúmeras bibliotecas para aplicações específicas, criadas tanto pela empresa quanto pela comunidade¹⁹.

Para realizar a atribuição do código montado na plataforma Arduino dentro do *firmware* da placa é necessário que o componente esteja conectado ao dispositivo ao qual está sendo realizada a criação do código por meio de uma conexão via cabo USB e, uma vez inserido, o mesmo estará pronto para o uso e realizará sempre a mesma função especificada em sua programação até ocorrer alguma alteração no *script* interno do mesmo. Na figura 12 podemos identificar a tela inicial do Arduino IDE, onde se realiza a codificação dos componentes físicos da placa.

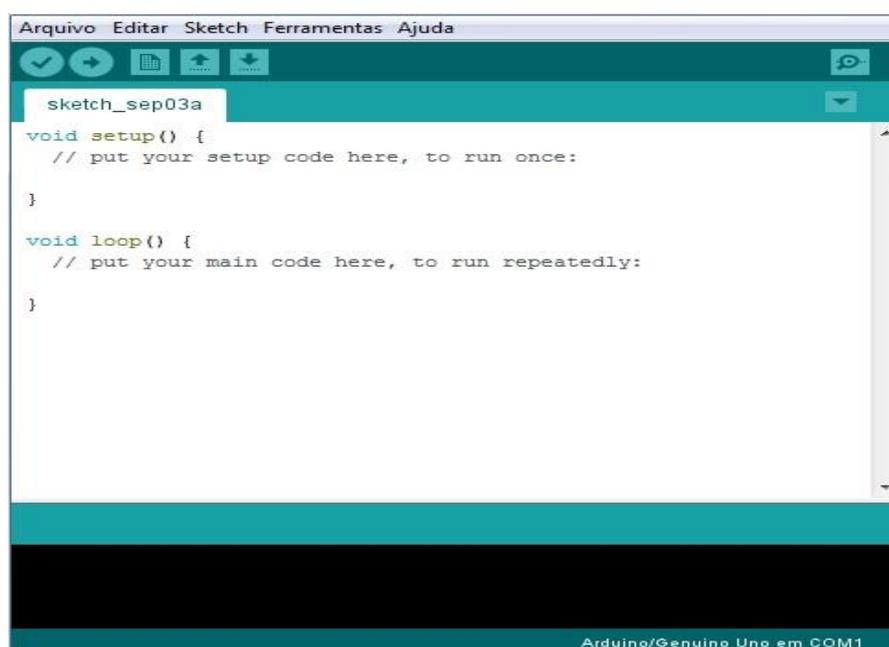


Figura 12: Tela inicial da IDE Arduino (feita pelo autor, 2018).

No capítulo seguinte se exhibe a configuração dos micro controladores Raspberry Pi e Arduino de maneira a coexistirem no mesmo contexto de IoT e suas necessárias alterações físicas e de sistema para que o projeto seja concluído.

¹⁹ Disponível em <<https://www.arduino.cc/en/Main/ReleaseNotes>> Acesso em 03 set. 2018.

4. CONFIGURAÇÃO DOS COMPONENTES

De forma a utilizar das tecnologias e ferramentas anteriormente redigidas, realiza-se então a modelagem e execução das plataformas, assim como suas configurações gerais de conexão entre as placas, montagem física de ambos com a *protoboard* juntamente com o restante dos componentes eletrônicos (LEDs, sensores, resistores) e preparação para a fase de programação dos códigos. Para isso informa-se da viabilidade e necessidade das configurações, tais quais estão designadas para esta devida implementação, onde por necessidade ocorre a comunicação serial entre os dispositivos Raspberry Pi 3 e Arduino Uno, de maneira a transmitirem dados entre através de suas portas *Reception* (RX) e *Transmission* (TX). Outras configurações internas do Raspberry Pi são também explanadas, como as configurações de sistema operacional, tal como a utilização das linhas de comando (*prompt*) do Raspbian (Linux), além da configuração de servidor também executada internamente ao sistema operacional do Raspberry Pi.

4.1. CONFIGURAÇÃO DO RASPBERRY PI

Para realizar a instalação inicial do sistema operacional, poderá ser utilizado o software NOOBS, desta forma optando pelo Raspbian, assim como mostrado abaixo na figura 13, onde o mesmo encontra-se predefinido como padrão e recomendado para usos gerais.

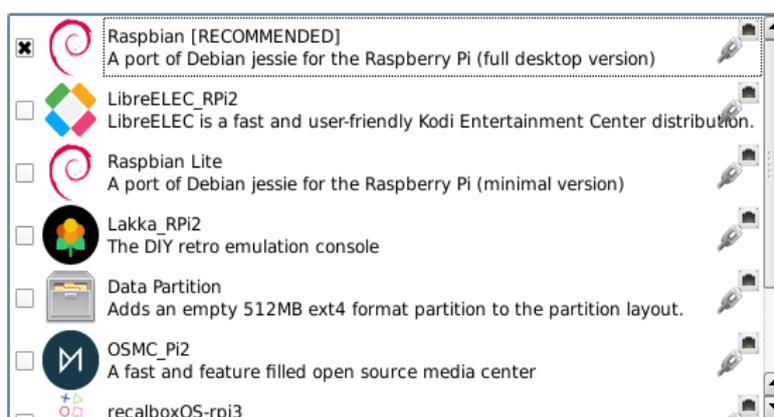


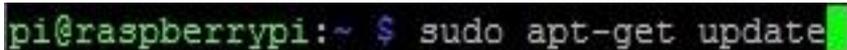
Figura 13: Tela de inicialização do NOOBS. (Raspberry Pi Foundation, 2018)²⁰.

²⁰ Disponível em <<https://www.raspberrypi.org/documentation/installation/noobs.md>>

Pode-se também ser efetuada a instalação do sistema operacional ao colocar os arquivos de instalação escolhidos diretamente no cartão SD, evitando a necessidade de uso do NOOBS, caso já possua conhecimento de quais ferramentas utilizarem.

Para realizar as configurações iniciais, é necessário que a versão da distribuição esteja atualizada (neste caso notifica-se o uso do Raspberry Pi 3 modelo B, com sistema operacional Raspbian Jessie), para que desta forma evite problemas de compatibilidade entre serviços ou até mesmo *bugs*.

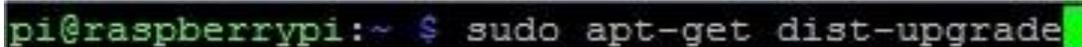
Com isso utilizam-se os comandos de atualização do sistema, onde todos os componentes inseridos no sistema receberão uma atualização para a versão atual (caso não estejam atualizados), como pode analisado na figura 14.



```
pi@raspberrypi:~ $ sudo apt-get update
```

Figura 14: Comando para atualização do sistema. (Feita pelo autor via SSH, 2018).

Já como mostrado na figura 15, realiza-se a atualização dos pacotes, acréscimo de recursos, novas funcionalidades e aplicações mais completas.



```
pi@raspberrypi:~ $ sudo apt-get dist-upgrade
```

Figura 15: Comando para atualização de pacotes. (Feita pelo autor via SSH, 2018).

Para que seja possível utilizar o Raspberry Pi como um servidor é necessário que o mesmo obtenha uma configuração de IP fixa, assim como seu referente DNS. Desta maneira acessamos o arquivo ao qual guardará essa configuração a partir do comando “sudo nano /etc/dhcpd.conf”, e é inserido os comandos como mostrado na figura 16. Esta configuração mantém tanto o endereço de IP do servidor fixo, quanto o sistema de domínios que será utilizado para acesso à rede.

```
interface eth0
static ip_address=192.168.1.10/24
static routers=192.168.1.1
static domain_name_server=192.168.1.1

interface wlan0
static ip_address=192.168.1.7
static routers=192.168.1.1
static domain_name_server=192.168.1.1
```

Figura 16: Configuração de IP estático. (Feita pelo autor via SSH, 2018).

4.1.1. CONFIGURAÇÃO DE PORTAS SERIAIS

O passo seguinte é realizar as alterações de porta serial para que as portas de comunicação do Raspberry Pi possam ser utilizadas com outros *hardwares*. Esta configuração é necessária para liberar os pinos para uso, visto que por padrão elas são utilizadas para o *login* de *console*, e com isso mantêm-se bloqueadas para outros usos. Outro problema é que para que o *Bluetooth* do Raspberry Pi funcione, é necessário que a porta serial “/dev/ttyAMA0” (uart de alto desempenho), que é a porta padrão do sistema esteja ativa. Mas para a utilização dos pinos de recepção (RX) e de transmissão (TX) de dados será necessário que o serviço seja apontado à porta principal. A porta “/dev/ttyS0”, também conhecida como “*mini* uart” refere-se à porta serial GPIO, mas para que seja utilizado qualquer dispositivo extra acoplado ao Raspberry Pi (principalmente na questão de transmissão de dados), torna-se substancial a troca da mesma com a porta principal.

Por padrão, a porta serial GPIO mantém-se desativada, e para que seja realizada a ativação da mesma utiliza-se o comando “sudo nano /boot/config.txt” para abrir o arquivo do diretório²¹.

São inseridas então duas linhas de código, como visto abaixo na figura 17, tal qual a primeira ativa a porta serial (uart), enquanto a segunda linha inserida serve para estabilizar a frequência dos *cores* do processador (que era realizado automaticamente antes da ativação desta porta serial). Logo após reinicie o sistema (reboot) para que as alterações obtenham resultado.

```
#dtoverlay=lirc-rpi

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on

# NOOBS Auto-generated Settings:
hdmi force hotplug=1
enable_uart=1
core freq=250
```

Figura 17: Ativação de porta serial da GPIO. (Feita pelo autor via SSH, 2018).

Por padrão, a porta da GPIO está mapeada nos pinos 14 e 15, diretamente na “/dev/ttyS0”, ao qual é dada o pseudônimo de porta “serial0”, enquanto a porta “/dev/ttyAMA0” é mostrada como porta “serial1”²². Para realizar a consulta do apontamento das portas seriais utiliza-se o comando “ls -l /dev”, que mostrará a ordem ao qual estão configuradas as portas seriais, se estão corretamente habilitadas e quais são elas. Também é possível observar outras portas e seu devido nível de acesso para alteração, visualização, remoção, etc.

²¹ Disponível em < <https://spellfoundry.com/2016/05/29/configuring-gpio-serial-port-raspbian-jessie-including-pi-3/>> Acesso em 01 jul. 2018.

²² Disponível em < <https://spellfoundry.com/2016/05/29/configuring-gpio-serial-port-raspbian-jessie-including-pi-3/>> Acesso em 01 jul. 2018.

Abaixo na figura 18, nota-se a configuração atual das portas seriais do Raspberry Pi 3, que, portanto, posteriormente deverão ser alteradas para que haja o uso correto da porta de comunicação entre o Raspberry Pi e o Arduino.

```
crw-rw-rw- 1 root root      1,   8 jul 22 13:26 random
drwxr-xr-x 2 root root      60 dez 31 1969 raw
crw-rw-r-- 1 root netdev   10,  58 jul 22 13:26 rfkill
lrwxrwxrwx 1 root root      5 jul 22 13:26 serial0 -> ttyS0
lrwxrwxrwx 1 root root      7 jul 22 13:26 serial1 -> ttyAMA0
drwxrwxrwt 2 root root     40 nov  3 2016 shm
drwxr-xr-x 3 root root    160 jul 22 13:26 snd
lrwxrwxrwx 1 root root     15 nov  3 2016 stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root     15 nov  3 2016 stdin  -> /proc/self/fd/0
lrwxrwxrwx 1 root root     15 nov  3 2016 stdout -> /proc/self/fd/1
crw-rw-rw- 1 root tty       5,   0 jul 22 13:26 tty
crw--w---- 1 root tty       4,   0 jul 22 13:26 tty0
crw--w---- 1 root tty       4,   1 jul 22 13:27 tty1
```

Figura 18: Consulta de portas seriais. (Feita pelo autor via SSH, 2018).

Para utilizar a porta serial para qualquer coisa além do console deve-se desabilitá-lo²³, liberando então o uso da porta serial “/dev/ttyS0” para outros serviços. Com isso, basta desabilitar o serviço “Getty” (*console*) através dos comandos “sudo systemctl stop serial-getty@ttyS0.service” e “sudo systemctl disable serial-getty@ttyS0.service”, além de remover o *console* do arquivo no diretório “/boot/cmdline.txt”, como mostrado na figura 19.

```
dwc_otg.lpm_enable=0 console=serial0,115200 console=tty1 root=/dev/mmc
```

Figura 19: Remover console. (Feita pelo autor via SSH, 2018).

Para reativar o Bluetooth, é necessário apontá-lo para a porta “/dev/ttyS0”, onde terá um desempenho inferior em relação à configuração padrão, porém será possível a utilização deste recurso, mesmo com o serviço de GPIO na porta principal “/dev/ttyAMA0”. Desta

²³ Disponível em < <https://spellfoundry.com/2016/05/29/configuring-gpio-serial-port-raspbian-jessie-including-pi-3/> Acesso em 01 jul. 2018.

forma, basta acessar o arquivo no diretório “/boot/config.txt” e realizar a inserção da linha de comando mostrada na figura 20, e reiniciar o sistema (reboot).

```
# NOOBS Auto-generated Settings:
hdmi_force_hotplug=1
enable_uart=1
core_freq=250
dtoverlay=pi3-miniuart-bt
```

Figura 20: Reativar Bluetooth. (Feita pelo autor via SSH, 2018).

Ao fim, realiza-se a verificação da posição das portas seriais, obtendo um resultado semelhante ao da figura 21.

```
drwxr-xr-x 2 root root          60 dez 31  1969 raw
crw-rw-r-- 1 root netdev    10,  58 jul 22  14:12 rfdkill
lrwxrwxrwx 1 root root          7 jul 22  14:12 serial0 -> ttyAMA0
lrwxrwxrwx 1 root root          5 jul 22  14:12 serial1 -> ttyS0
drwxrwxrwt 2 root root          40 nov  3  2016 sim
```

Figura 21: Verificação das portas seriais atuais. (Feita pelo autor via SSH, 2018).

Com essas configurações é possível realizar a transferência de dados entre componentes, assim como a montagem dos circuitos de forma adequada, sem que nenhum recurso do Raspberry Pi (como o próprio *Bluetooth*) seja perdido.

4.2. MONTAGEM DOS COMPONENTES FÍSICOS

Na realização da montagem serão utilizados além do Raspberry Pi, uma *protoboard*, um LED (diodo emissor de luz), um sensor de luminosidade LDR de sinal analógico e um Arduino (o mesmo está sendo utilizado apenas como conversor de sinal digital para

analógico, podendo ser substituído por um conversor básico para Raspberry Pi) além de resistores e *jumpers* utilizados para manter o funcionamento da maquete.

Na figura 22 está exemplificado a modelagem dos componentes e suas devidas conexões para funcionamento, transmissão e recepção de dados entre o Raspberry Pi 3, o Arduino Uno e os componentes inseridos na *protoboard*.

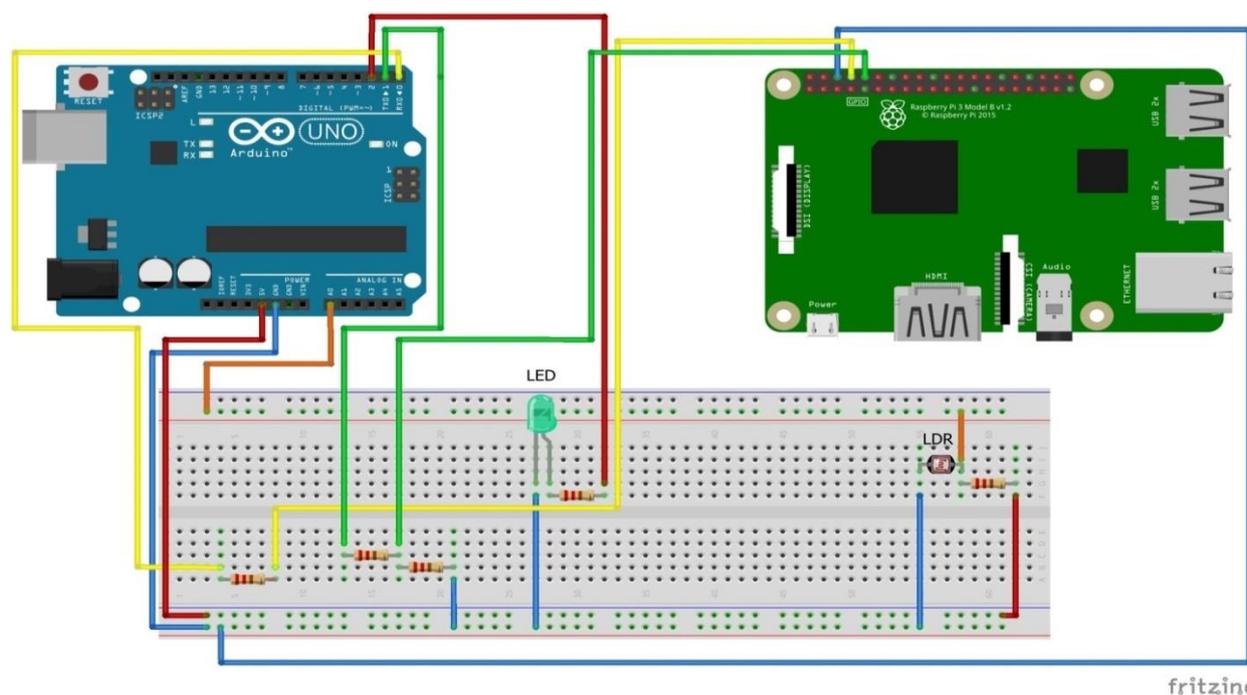


Figura 22: Prototipação dos componentes físicos. (Feita pelo autor via Fritzing, 2018).

Como visto acima, a modelagem mostra as conexões e transferência de dados realizados entre as placas. Os jumpers azuis se tratam dos fios de carga negativa, que são os GND (terra), os jumpers vermelhos coincidem com a carga positiva de 5 volts providas do Arduino e 3 volts do Raspberry Pi, ao qual "energiza" a *protoboard* e seus componentes. O jumper laranja refere-se ao coletor de dados oriundos do sensor LDR, que será captado pelo Arduino e posteriormente enviado ao Raspberry Pi. Tal transferência de dados é realizada por meio dos jumpers verdes que correspondem às portas de transmissão do Arduino (TX) e Recepção do Raspberry Pi (RX), que são ativados quando os sinais dos jumpers amarelos (TX do Raspberry e RX do Arduino) são ativados. O sinal é enviado do Raspberry Pi, como uma requisição de operação, gerando desta forma um gatilho de início para o funcionamento de todo o sistema.

4.3. CONFIGURAÇÃO DO SERVIDOR

A configuração de servidor em um Raspberry Pi é feita de maneira semelhante a qualquer distribuição Linux. Neste caso será realizado por meio da API de servidor Apache, que poderá ser instalado através do comando “sudo apt-get install apache2”. Para testar a instalação do mesmo basta buscar em qualquer *browser* (navegador de internet) pelo endereço IP ao qual o servidor (Raspberry Pi) está configurado. A figura 23 mostra o funcionamento da mesma.

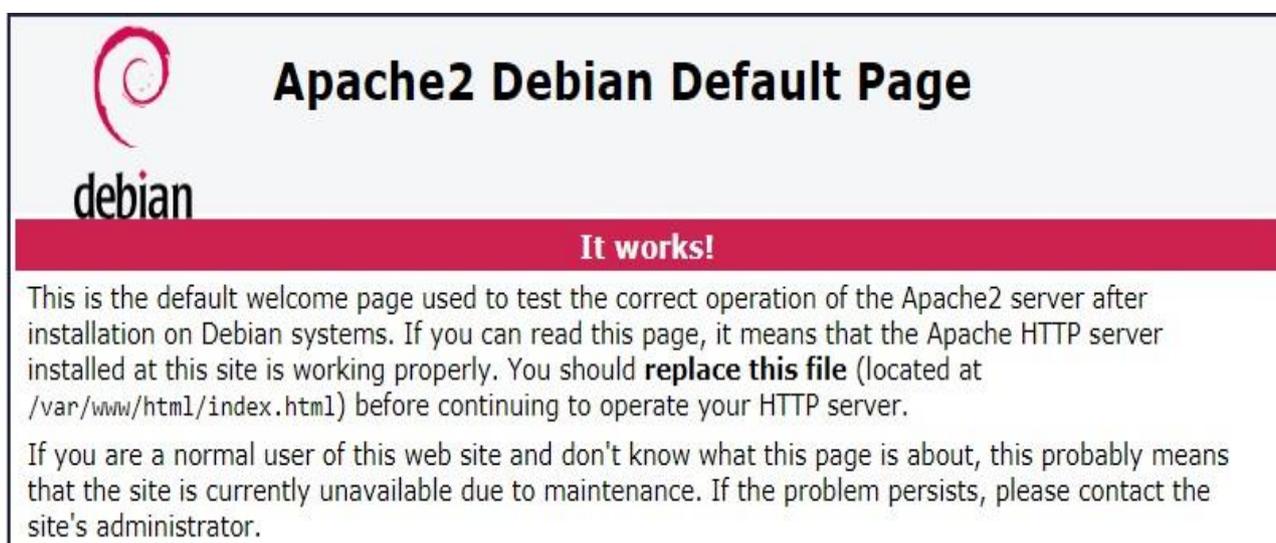
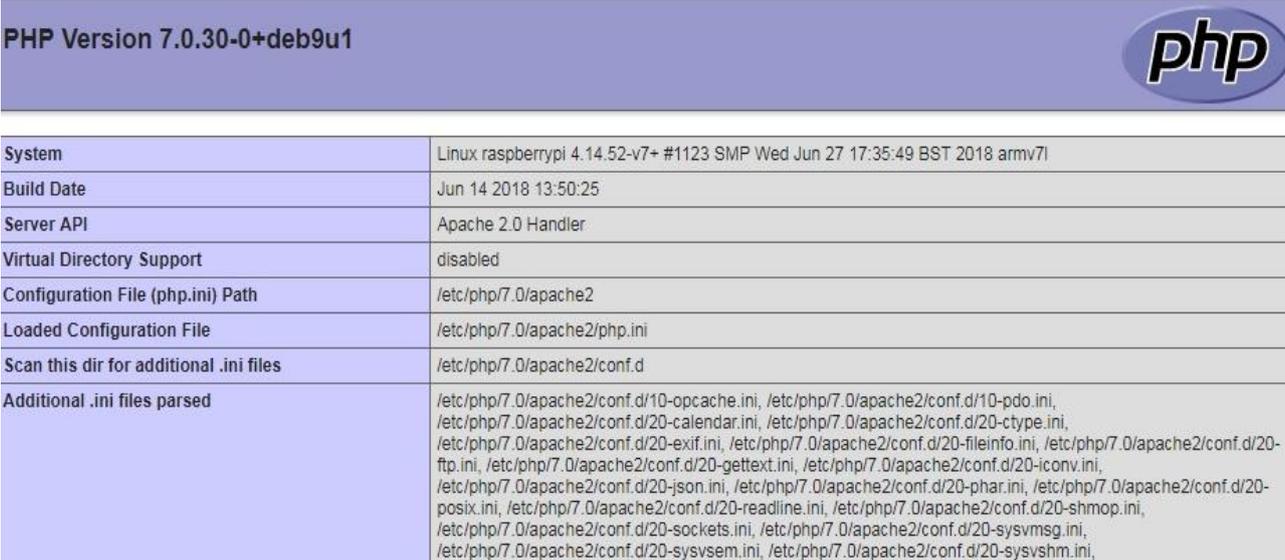


Figura 23: Fração da página teste do servidor. (Feita pelo autor, 2018).

O passo seguinte é realizar a configuração da linguagem PHP, que será responsável pelo modelamento e dinamismo da página *web* no servidor, podendo-se então realizar operações e inserções de valores coletados pelos componentes físicos do Raspberry Pi. Portanto é executado o comando para a instalação dos pacotes PHP “sudo apt-get install php”, desta forma poderá então utilizar seus devidos recursos. Para questões de teste, será necessário efetuar o comando “sudo rm /var/www/html index.html”, removendo assim a página de índice padrão com o intuito de poder criar um novo índice para o PHP, através de “sudo nano /var/www/html index.php”, onde será inserido o código “<? php phpinfo (); ?>”.

Caso já exista alguma versão antiga do PHP instalada, apenas realize o *upgrade* do sistema “sudo apt-get upgrade” que o mesmo corresponderá com a versão atual de instalação na plataforma.

Ao realizar o teste de servidor mostrado anteriormente, sendo acrescentado o arquivo “index.php” no campo da URL de pesquisa, será possível verificar a efetividade da instalação do PHP, assim como exibido na figura 24.



PHP Version 7.0.30-0+deb9u1	
System	Linux raspberrypi 4.14.52-v7+ #1123 SMP Wed Jun 27 17:35:49 BST 2018 armv7l
Build Date	Jun 14 2018 13:50:25
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini

Figura 24: Fração da página teste do PHP. (Feita pelo autor, 2018).

Com a instalação de ambos, a etapa seguinte é a implementação das aplicações, que sequenciam a comunicação entre o Raspberry Pi (controle dos dados e iniciação do programa) e Arduino (que controla os dispositivos acoplados à *protoboard*) e a comunicação entre os dispositivos integrados com o servidor *web*. Tal tarefa será realizada tanto pelo raspberry Pi, através da codificação Python quanto do programa HTML/PHP disponibilizado ao servidor Apache.

5. IMPLEMENTAÇÃO

Nesta etapa, ocorre a codificação para o funcionamento físico da modelagem, com a comunicação, envio de sinais gerados pelo Arduino (Arduino IDE), recebimento e redirecionamento de valores efetuados pelo Raspberry Pi (realizado pelo Python) para a interface gráfica de acesso realizada em HTML, com suas funcionalidades apresentadas por meio da aplicação do PHP sobre os códigos Python, onde serão gerados os dados em tempo real para a consulta do usuário.

5.1. COMUNICAÇÃO ENTRE COMPONENTES FÍSICOS

Iniciando-se pela programação do Arduino, tal qual opera os componentes da placa, será o responsável para ligar os sensores e LEDs, sendo dependente das requisições realizadas pelo Raspberry Pi e enviar os dados recebidos para a aplicação Python. Na figura 25 podemos ver o código para a funcionalidade automática da modelagem, onde se utiliza dos dados captados do sensor para ligar/desligar os LEDs automaticamente dependendo do valor coletado.

```
int sensores(){  
  
    float ldrValor = analogRead(0);  
    ldrValor = map(ldrValor, 1023, 0, 0, 100); //O valor lido será entre 0 e 1023  
  
    Serial.println(ldrValor);  
  
    if(buf == 'A'){  
        //ldr();  
        if (ldrValor <= 20) {  
            digitalWrite(ledPin3, HIGH);  
        }else{  
            digitalWrite (ledPin3, LOW);  
        }  
    }  
}
```

Figura 25: Código de automação no Arduino. (Feita pelo autor, 2018).

Quando o valor captado pelo LDR for abaixo de 20% (nível relativamente baixo de luz) o LED acenderá automaticamente, assim que a opção “A” for ativada. Tal caractere estará atrelado ao botão que realizará a automatização no servidor *web*.

Há também a possibilidade controle manual, para que o módulo do LED seja ligado e desligado pela escolha do cliente, independentemente do valor captado através do sensor de luminosidade.

```
int sensores() {  
  
    float ldrValor = analogRead(0);  
    ldrValor = map(ldrValor, 1023, 0, 0, 100); //O valor lido será entre 0 e 1023  
  
    Serial.println(ldrValor);  
  
    if(buf == 'D'){  
        //desl();  
        digitalWrite (ledPin3, LOW);  
    }  
  
    if(buf == 'L'){  
        //liga();  
        digitalWrite (ledPin3, HIGH);  
    }  
}
```

Figura 26: Código de controle manual no Arduino. (Feita pelo autor, 2018).

Como visto na figura 26 logo acima, há também opção de ligar e desligar a energia do LED, independentemente de qual valor for recebido do sensor de luminosidade. Com a ativação de “L” o Raspberry Pi redireciona o sinal para ligar os componentes manualmente, enquanto que com a ativação de “D”, o mesmo desliga de forma manual. Ambas estas opções estão atreladas aos botões da página referentes ao controle manual.

Na etapa seguinte promovem-se as chamadas de requisições realizadas pelo cliente via *web*, onde é redirecionado para os códigos de Python, que desta forma interpretam o acesso e liberam o código Arduino para realizar sua devida programação.

Na figura 27 podemos ver o código da programação em Python responsável pelo envio da requisição do uso do sensor de luminosidade, que ativará automaticamente a luz no caso do nível estabelecido de luminosidade captado não for suficiente. O sinal de requisição “A” (automático) será enviado através do comando do usuário pela interface de acesso *web*, atribuindo então ao Arduino a aptidão de resolver tal requisição.

```
while(1):  
  
    ser.write("A")  
  
    resposta = ser.readline()  
  
    print(resposta)  
  
    file = open('/var/www/html/pagina/lumi.txt', 'w')  
    string = json.dumps(resposta)  
  
    file.write(string)  
    file.close()  
  
    time.sleep(0.3)
```

Figura 27: Requisição de funcionamento automático. (Feita pelo autor, 2018).

Na utilização manual do sistema utiliza-se o código Python indicado na figura 28, onde há a possibilidade de ligar e desligar o LED por conta própria, desativando assim o modo automático do sistema. Este código funciona através do envio das informações “L” (ligar) e “D” (desligar) durante a requisição de funcionamento do Arduino, ignorando a quantidade de luz captada pelo sensor LDR, porém mostrando o percentual adquirido de mesma forma.

```
ser = serial.Serial("/dev/ttyAMA0", 115200)  
  
while(1):  
    ser.write("L")  
    resposta = ser.readline()  
    print(resposta)  
    file = open('/var/www/html/pagina/lumi.txt', 'w')  
    string = json.dumps(resposta)  
  
    file.write(string)  
    file.close()
```

Figura 28: Requisição de funcionamento manual (Feita pelo autor, 2018).

A aplicação web ao qual o usuário terá o controle do switch de luz tem como base o código mostrado na figura 29, onde se pode ver a codificação dos recursos visuais gerados através do código HTML juntamente com a interface CSS.

```

<div id="div_Container" class="ind" >
<h1 align="center">TCC Smart Homes</h1>
  <div id="div_Topo">
    

  <div id="div_Menu">
    <ul>
      <li> <a href="index.php"> Inicio </a> </li>
      <li> <a href="contato.php"> Informação dos Dados </a> </li>
      <li> <a href="pag-register.php"> Sobre </a> </li>
    </ul>
  </div>

  <div>
    <button type="button" class="css3button"> Ligar </button>
    <button type="button" class="css3button"> Automatico </button>
    <button type="button" class="css3button"> Desligar </button>

    <br>
    <br>
  </div>

```

Figura 29: Trecho HTML/PHP da Interface gráfica. (Feita pelo autor, 2018).

A interface visual do procedimento está demonstrada na figura 30, onde se encontra as subpáginas de automação, ativação manual e desligamento da iluminação, que exhibe dentro de uma tabela os últimos resultados capturados, de maneira quase simultânea para o acesso do usuário.

TCC Smart Homes



CODIGO	SENSOR	VALOR	TIPO	DATA/HORA
01	Luminosidade	"34"	%	23:04:37 09-07-2018

Vinicius de Luca 2018 - Casas Inteligentes.

Figura 30: Interface gráfica para o usuário. (Feita pelo autor, 2018).

CONCLUSÃO

O uso de tecnologias programáveis como os modelos do micro controlador Raspberry Pi pode trazer diversos benefícios no contexto de casas automáticas e casas inteligentes, onde demonstram capacidade de facilitar o uso e manuseio de recursos elétricos/eletrônicos de sua casa remotamente através de uma interface sem a necessidade de alterações físicas dos componentes. Este recurso pode ser explorado em outros setores da área de automação de casas devido à maleabilidade ao alterar o foco da integração, onde, uma vez que sua configuração básica estiver pronta será necessário apenas ajustes nos componentes físicos e a inserção de novos códigos.

Com isso abre-se uma lacuna de possibilidades para partir em direções variadas desta área, como em casos específicos de segurança residencial, utilidade medicinal, entretenimento ou até mesmo a automação total dos componentes eletroeletrônicos encontrados em uma casa.

Pode-se também realizar o uso deste tipo de tecnologia em casos pontuais de empresas, onde através de apenas um acesso poderá evitar a necessidade de realizar tal tarefa manualmente, poupando tempo ou em alguns casos até mesmo um cargo, diminuindo custos.

Desta forma nota-se que através da IoT é possível realizar inúmeras tarefas que possam facilitar o controle residencial e, ao utilizar o micro controlador Raspberry Pi torna-se uma aplicação flexível e de custo-benefício superior ao de circuitos prontos encontrados no mercado. O Raspberry Pi mostra-se adequado pois pode-se acoplar outros tipos de controladores como no caso do Arduino, além disso o mesmo se beneficia de seus componentes físicos (de integração) e programas (compilação/interpretação de códigos, configuração de servidor *web*, capacidade de armazenamento de dados, etc.). Consequentemente todo o conjunto de ferramentas e componentes torna a integração ideal para as práticas anteriormente vistas, que estão cada vez mais comuns no nosso cotidiano e que serão cada vez mais requisitados por quem necessite facilitar ou encurtar algum procedimento requisitado frequentemente.

Portanto, através do trabalho anteriormente realizado disponibiliza-se de uma gama de oportunidades e meios ao qual seguir para o foco principal do projeto, com isso é possível

realizar a inserção de novos módulos e ferramentas, que de alguma forma deem suporte para o projeto nas áreas de segurança, acessibilidade, custo e manutenção.

Como propriamente mostrado, a modulagem e configuração básica remete-se ainda a nível acadêmico, para fins de estudo ou eventual uso próprio. Para que o mesmo torne-se um produto há a necessidade de implantação de alguns módulos e configurações, assim como incrementar os códigos e modelagem física do projeto.

Tendo em vista a possibilidade de aumento de funcionalidades, sensores, alterações de componentes físicos (inserir relés e lâmpadas de 220 volts) e movimentação de dados, propõe-se que seja realizado então a acoplagem de uma base de dados no próprio servidor (Raspberry Pi), podendo obter então uma recepção maior dos dados, no momento em que foram coletados e seus devidos resultados, sendo capaz então de emitir relatórios que mostre padrões e identifiquem a melhor maneira de utilizar o sistema, melhorando o desempenho para auxiliar ainda mais o cliente.

Busca-se também a efetivação da ampliação da área de alcance do sistema, tal qual terá acesso remoto através de qualquer lugar do planeta através da *internet*. O acesso de rede externa pode ser realizado ao configurar as portas de um roteador de forma que o mesmo mantenha apenas a porta correspondente ao servidor (Raspberry Pi) aberta, onde será acessível apenas pela aplicação. O redirecionamento de portas, juntamente com a configuração de DDNS (DNS dinâmico), ao qual terão papel essencial no acesso através de rede externa, é geralmente voltado à área profissional, onde necessita a requisição da compra de um domínio da internet, tal domínio é atrelado ao IP externo do servidor *web*, possibilitando então o seu acesso. Devido a tais configurações, é necessário o aumento na segurança tanto do domínio quanto do servidor, onde deverão conter diversos níveis de segurança, por meio de senhas e criptografia ou até mesmo restrição de acesso por meio de um *software* de *firewall*.

Por via deste nota-se que há um longo caminho para a conclusão total do sistema de maneira que se torne um produto comercializável, mas também se remete que opcionalmente o projeto poderá haver mudanças de contexto, de acordo com a necessidade e oportunidade, tendo sempre como pretensão viabilizar o custo do protótipo, ao mesmo tempo em que sua devida configuração funcione, facilite e agrade a quem o utilizar.

REFERÊNCIAS

ANSARI, Aamir Nizam, SEDKYL, Mohamed, SHARMA, Neelam e TYAGIL, Anurag. An Internet of Things Approach for Motion Detection using Raspberry Pi. 2015. 131p-133p. Disponível em <<http://ieeexplore.ieee.org/abstract/document/7111554/>>. Acesso em 28 out. 2017.

Arduino. Arduino - Introduction. Disponível em <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em 02 set. 2018.

Arduino. Arduino Software Release Notes. Disponível em <<https://www.arduino.cc/en/Main/ReleaseNotes>>. Acesso em 03 set. 2018.

Arduino. Arduino Software (IDE). Disponível em <<https://www.arduino.cc/en/Guide/Environment>>. Acesso em 03 set. 2018.

Arduino. Arduino Uno Rev3. Disponível em <<https://store.arduino.cc/usa/arduino-uno-rev3>>. Acesso em 02 set. 2018.

BANDYOPADHYAY, Soma, SENGUPTA, Munmun, MAITI, Souvik and DUTTA, Subhjit. A Survey of Middleware for Internet of Things. 2011. 288p-296p. Disponível em <https://link.springer.com/chapter/10.1007/978-3-642-21937-5_27>. Acesso em 04 mar. 2018.

Breadboard. Disponível em <https://en.wikipedia.org/wiki/Breadboard#/media/File:400_points_breadboard.jpg>. Acesso em 11 jun. 2018.

Cactus.io. How to Hookup Light Dependent PhotoResistor (LDR) to an Arduino. Disponível em <<http://cactus.io/hookups/sensors/light/ldr/hookup-arduino-to-ldr-sensor>>. Acesso em 13 jun. 2018.

How to Raspberry Pi. How to Install a Web Server on the Raspberry Pi (Apache + PHP + MySQL). 2018. Disponível em <<https://howtoraspberrypi.com/how-to-install-web-server-raspberrypi-lamp/>>. Acesso em 1 jul. 2018.

H. P. Dijkstra, N. Pollock, R. Chakraverty, and J. M. Alonso. 2014. Managing the health of the elite athlete: a new integrated performance health management and coaching model. *British Journal of Sports Medicine* 48, 7. Disponível em <<https://bjsm.bmj.com/content/48/7/523>> Acesso em 21 jan. 2018.

Iotonlinestore. The Internet of Things: What is and Why You Should Care. Disponível em <<http://www.iotonlinestore.com>>. Acesso em 03 mar. 2018.

KURKINEN L. Smart homes and home automation, Berg Insight M2M Research Series; [16.01.14]. Disponível em <<http://www.berginsight.com/ReportPDF/ProductSheet/bi-sh2-ps.pdf>>. Acesso em 28 out. 2017.

MAKSIMOVIĆ, Mirjana, VUJOVIĆ, Vladimir, DAVIDOVIĆ, Nikola, MILOSEVIĆ, Vladimir and PERISIĆ, Branko. Raspberry Pi as Internet of Things Hardware: Performances and Constraints. 2014. Disponível em <https://www.researchgate.net/profile/Vladimir_Vujovic/publication/280344140_EL16_Maksimovic_Vujovic_Davidovic_Milosevic_Perisic/links/55b3368608ae9289a08594aa.pdf>. Acesso em 04 mar. 2018.

MecaWeb Education Site. PWM – Modulação Por Largura de Pulso. Disponível em <http://www.mecaweb.com.br/electronica/content/e_pwm>. Acesso em 02 set. 2018.

MERCHANT, H. K, AHIRE, D. D. Industrial automation using IoT with Raspberry Pi. Disponível em <<https://pdfs.semanticscholar.org/87cb/92d1ae21ab73994698a0735b06afb0fe8f73.pdf>>. Acesso em 11 mar. 2018.

NAYYAR, Anand. Best Open Source Databases for IoT Applications. 2017. Disponível em <<http://opensourceforu.com/2017/05/best-open-source-databases-iot-applications/>>. Acesso em 17 mar. 2018.

NGU, H. H. Anne, GUTIERREZ, Mario, METSIS, Vangelis, NEPAL, Surya and SHENG, Quan Z. IoT Middleware: A Survey on Issues and Enabling Technologies. 2016. Disponível em <<http://ieeexplore.ieee.org/abstract/document/7582463/>>. Acesso em 03 mar. 2018.

NOOBS. Raspberry Pi foundation. Disponível em <<https://www.raspberrypi.org/documentation/installation/noobs.md>>. Acesso em 30 jun. 2018.

OŻADOWICZ, Andrzej, GRELA, Jakub. Control Application for Internet of Things Energy Meter – a Key Part of Integrated Building Energy Management System. 2015. Disponível em <<http://ieeexplore.ieee.org/abstract/document/7301607/>>. Acesso em 21 jan. 2018.

PAVITHRA D, BALAKRISHNAN, Ranjith. Iot based Monitoring and Control System for Home Automation. Disponível em <<http://ieeexplore.ieee.org/abstract/document/7342646/?reload=true>>. Acesso em 28 out. 2017.

Raspberry Pi Foundation. Raspberry Pi. Disponível em <<https://www.raspberrypi.org>>. Acesso em 10 mar. 2018.

Raspberry Pi Foundation. Raspberry Pi 1 Model B Plus. Disponível em <<https://www.raspberrypi.org/products/raspberry-pi-1-model-b-plus/>>. Acesso em 10 mar. 2018.

Raspberry Pi Foundation. Raspberry Pi 3 Model B. Disponível em <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>>. Acesso em 10 mar. 2018.

Raspberry Pi Foundation. Debian ‘armhf’ (hard-float ABI) for Raspberry Pi. 2018. Disponível em <<https://www.raspberrypi.org/forums/viewtopic.php?p=47787>>. Acesso em 10 mar. 2018.

Raspberry Pi Foundation. Raspbian OS. Disponível em <<https://www.raspberrypi.org/downloads/raspbian/>> Acesso em 10 mar. 2018.

SOUZA, Fabio. Arduino Uno. Disponível em <<https://www.embarcados.com.br/arduino-uno/>>. Acesso em 02 set. 2018.

SparkFun. Arduino Shields. Disponível em <<https://learn.sparkfun.com/tutorials/arduino-shields>>. Acesso em 02 set. 2018.

VALLE, Benoît, SIMONNEAU, Thierry, BOULORD, Romain, SOURD, Francis, FRISSON, Thibault, RYCKEWAERT, Maxime, HAMARD, Phillipe, BRICHET, Nicolas, DAUZAT, Myriam and CHRISTOPHE, ANGÉLIQUE. PYM: A new, affordable, image-based method using a Raspberry Pi to phenotype plant leaf area in a wide diversity of environments. 2017. Disponível em <<https://plantmethods.biomedcentral.com/articles/10.1186/s13007-017-0248-5>>. Acesso em 10 mar. 2018.

VUJOVIĆ V, MAKSIMOVIĆ M. Raspberry Pi as a Sensor Web node for home automation. *Comput Electr Eng* (2015). Disponível em <<http://www.sciencedirect.com/science/article/pii/S0045790615000257>>. Acesso em 27 out. 2017.

YOUNG, Warren. What is a “breadboard”? 2015. Disponível em <<http://tangentsoft.net/elec/breadboard.html>>. Acesso em 11 jun. 2018.

WANG, Feng, HU, Liang, ZHOU, Jin and ZHAO, Kuo. The IoT middleware architecture based on SOA. 2015. Disponível em <<http://downloads.hindawi.com/journals/js/2015/827045.pdf>>. Acesso em 04 mar. 2018.

WATKINS, Jon. Configuring the GPIO Serial Port on Raspbian Jessie and Stretch Including Pi 3. 2017. Disponível em <<https://spellfoundry.com/2016/05/29/configuring-gpio-serial-port-raspbian-jessie-including-pi-3/>>. Acesso em 1 jul. 2018.

ZHAO, Cheah Wai, JEGATHEESSAN, Jayanand, LOON, Son Chee. Exploring IOT Application Using Raspberry Pi. 27p-31p. Disponível em <<https://pdfs.semanticscholar.org/ac8b/8cf9dd6dcacd4963da40200abcc56ce8ba49.pdf>>. Acesso em 27 out. 2017.

