

**GLAUCO FAGUNDES FERNANDES**

**Automação Residencial:** utilizando Internet das Coisas e ESP-8266

Guaratinguetá - SP  
2018

**Glauco Fagundes Fernandes**

**Automação Residencial:** utilizando Internet das Coisas e ESP-8266

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia Elétrica da Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Elétrica.

Orientador (a): Paloma Maria Silva Rocha Rizol

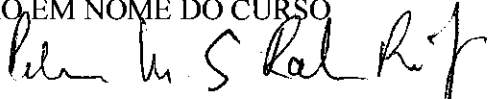
Guaratinguetá - SP  
2018

F363a	<p>Fernandes, Glauco Fagundes Automação residencial: utilizando internet das Coisas e ESP-8266 / Glauco Fagundes Fernandes – Guaratinguetá, 2018. 72 f : il. Bibliografia: f. 65-67</p> <p>Trabalho de Graduação em Engenharia Elétrica – Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2018. Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Paloma Maria Silva Rocha Rizol</p> <p>1. Automação residencial. 2. Arduino (Controlador programável). 3. Software gratuito. I.Título.</p> <p style="text-align: right;">CDU 681.3.068</p>
-------	--

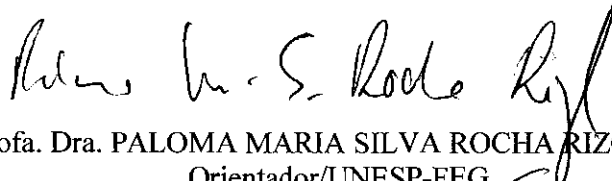
**GLAUCO FAGUNDES FERNANDES**

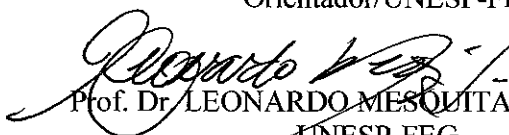
ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO  
PARTE DO REQUISITO PARA A OBTENÇÃO DO DIPLOMA DE  
"GRADUADO ENGENHARIA ELÉTRICA"

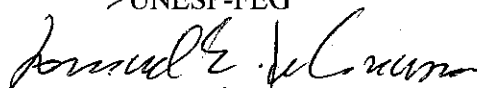
APROVADO EM SUA FORMA FINAL PELO CONSELHO DE CURSO DE  
GRADUAÇÃO EM NOME DO CURSO

  
Prof. Dra. PALOMA M. S. ROCHA RIZOL  
Coordenador

**BANCA EXAMINADORA:**

  
Prof. Dra. PALOMA MARIA SILVA ROCHA RIZOL  
Orientador/UNESP-FEG

  
Prof. Dr. LEONARDO MESQUITA  
UNESP-FEG

  
Prof. Dr. SAMUEL EUZÉDICE DE LUCENA  
UNESP-FEG

Novembro de 2018

## **DADOS CURRICULARES**

### **GLAUCO FAGUNDES FERNANDES**

**NASCIMENTO** 20.12.1992 – Taubaté / SP

**FILIAÇÃO** Wanderley Fernandes  
Vera Lucia Fagundes Fernandes

Aos meus pais, Vera e Wanderley  
pelo apoio e amor incondicional

“Só por que alguma coisa não faz o que você planejou  
que ela fizesse não quer dizer que ela seja inútil.”

Thomas Edison

## RESUMO

Este trabalho apresenta o estudo e a implementação de um protótipo de uma residência automatizada de baixo custo de implementação, com o uso de um *hardware* de controle denominado NodeMCU, ou ESP8266, que utiliza a linguagem baseada em C/C++ no ambiente de desenvolvimento de livre acesso do Arduino, o Arduino IDE, visando criar uma plataforma *Open Source* para que qualquer entusiasta consiga reproduzir ou melhorar o projeto criado.

A automação residencial vem ganhando cada vez mais espaço no mercado devido as novas tecnologias que tornam os projetos mais acessíveis, tanto com relação ao valor quanto em relação a materiais de consulta, originando em um tempo menor de pesquisa e desenvolvimento, além de menor quantidade de *hardware* a ser implementado, tendo como uma das principais tecnologias desenvolvidas e utilizadas, atualmente, a Internet das Coisas (IoT, *Internet of Things*), que interliga todos os equipamentos eletroeletrônicos da residência a rede de internet doméstica e possibilita que os usuários controle-os de qualquer rede de internet e de forma prática através de um aparelho de celular, por exemplo.

**PALAVRAS-CHAVE:** *Open Source*. Automação Residencial. Internet das Coisas. Baixo Custo. NodeMCU. Arduino IDE.

## ABSTRACT

This work covers the study and the implementation of a prototype of an automated house, focusing on low cost of implementation and using a hardware to control the project called NodeMCU, or ESP8266 that use to develop the software a language based on C/C++ on the free platform to develop of Arduino, called Arduino IDE, aiming to build an open platform so that any enthusiast can reproduce or increase the project by their own.

Residential automation has been gaining more and more space in the market due to new technologies that make projects more accessible, cheaper and easier to find reference material, resulting in a shorter research and development time spend, beyond a lower quantity of hardware to be implemented, having one of the main technologies developed and used, is the Internet of Things (IoT), witch interconnects all the electronics equipment with the internet network, making possible to users easy control them from any internet connection using a cellular device, for example.

**KEYWORDS:** Open Source. Residential Automation. Internet of Things. Low Cost. NodeMCU. Arduino IDE.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Índice de uso do celular no Brasil de acordo com o IBGE .....	15
Figura 2 – Previsão de uso do IoT .....	25
Figura 3 – Protocolo TCP/IP .....	26
Figura 4 – Arquitetura de Microcomputadores .....	28
Figura 5 – Arduino UNO.....	29
Figura 6 – Pinagem ATmega328P.....	31
Figura 7 – NodeMCU.....	32
Figura 8 – Esquemático do Projeto .....	42
Figura 9 – Módulo Relé de 2 Canais .....	43
Figura 10 – Instalando a biblioteca ESP8266 parte 1.....	45
Figura 11 – Instalando a biblioteca ESP8266 parte 2.....	45
Figura 12 – <i>Feeds</i> Adafruit IO parte 1 .....	47
Figura 13 – <i>Feeds</i> Adafruit IO parte 2 .....	48
Figura 14 – <i>Dashboards</i> Adafruit IO parte 1 .....	48
Figura 15 – <i>Dashboards</i> Adafruit IO parte 2 .....	49
Figura 16 – Protótipo Real .....	59
Figura 17 – Tabela de Acesso.....	61

## LISTA DE TABELAS

Tabela 1 – Tabela de Preços .....	40
Tabela 2 – Preço Total.....	40

## LISTA DE ABREVIATURAS E SIGLAS

BNDES	Banco Nacional do Desenvolvimento
CEO	Chefe Executivo de Ofício
CI	Circuito Integrado
CPU	<i>Central Processing Unit</i>
DTIM	<i>Delivery Traffic Indication Message</i>
EPROM	<i>Erasable Programmable Read Only Memory</i>
GPIO	<i>General-Purpose Input/Output</i>
HTTP	<i>Hypertext Transfer Protocol</i>
I2C	<i>Inter-Integrated Circuit</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	<i>Integrated Development Enviroment</i>
IO	<i>Input and Output</i>
IoT	<i>Internet of Things</i>
IP	Protocolo de Internet
LAN	<i>Local Area Network</i>
LDR	<i>Light Dependent Resistor</i>
LED	<i>Light Emitting Diode</i>
LoRaWAN	<i>Long Range Wide Area Network</i>
MCTIC	Ministério da Ciência, Tecnologia, Inovações e Comunicações
MCU	<i>Micro Controller Unit</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
M2M	<i>Machine to Machine</i>
NFC	<i>Near Field Communication</i>
PIB	Produto Interno Bruto
PWM	<i>Pulse Width Modulation</i>
RADAR	<i>Radio Detection and Ranging</i>
RAM	<i>Random Access Memory</i>
RFID	<i>Radio Frequency Identification</i>
RSSF	Redes de Sensores sem Fio
SPI	<i>Serial Peripheral Interface</i>
TCP	<i>Transmission Control Protocol</i>
UDP	Protocolo de Datagrama de Usuário
USB	<i>Universal Serial Bus</i>
WAN	<i>Wide Area Network</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	13
1.1	CONTEXTUALIZAÇÃO .....	15
1.2	MOTIVAÇÃO .....	16
1.3	OBJETIVO .....	17
1.4	ETAPAS DO TRABALHO .....	18
<b>2</b>	<b>AUTOMAÇÃO RESIDENCIAL</b> .....	20
2.1	ORIGEM E PRINCÍPIOS DA DOMÓTICA .....	21
2.2	INTERNET DAS COISAS, ORIGEM E PRINCÍPIOS .....	23
2.3	PROTOCOLOS DE REDES, PRINCÍPIOS E TIPOS .....	25
2.4	MICROCOMPUTADORES E SUAS ARQUITETURAS .....	28
2.4.1	<b>Arduino</b> .....	29
2.4.1	<b>ESP8266</b> .....	31
2.5	GESTÃO DE ENERGIA .....	33
2.6	SENSORES E ATUADORES .....	35
2.7	MQTT .....	36
<b>3</b>	<b>PESQUISA E DESENVOLVIMENTO</b> .....	38
3.1	DEFINIÇÃO DOS COMPONENTES .....	38
3.2	ESTUDO DE CUSTO .....	39
3.3	TUTORIAL DE DESENVOLVIMENTO .....	41
3.3.1	<b>Arduino IDE</b> .....	43
3.3.2	<b>Adafruit IO</b> .....	46
3.3.3	<b>IFTTT</b> .....	50
3.3.4	<b>Código</b> .....	51
<b>4</b>	<b>RESULTADOS</b> .....	58
4.1	PONTOS A SEREM MELHORADOS .....	59
4.2	ACERTOS .....	60
<b>5</b>	<b>CONCLUSÃO</b> .....	62
	<b>REFERÊNCIAS</b> .....	64

## 1 INTRODUÇÃO

Independente do grau de desenvolvimento do país, é imprescindível o debate sobre o declínio do setor industrial, pois argumentos como, o crescimento dos serviços, perderam força para explicar a queda das indústrias na participação do PIB, porém é notório que apesar das dificuldades apresentadas, a indústria ainda é de suma importância para o desenvolvimento tecnológico e para os sistemas de inovação (NOVOS ESTUDOS CEBRAP. v.36. no.3. 2017).

A fim de reverter o declínio, a indústria passa por uma nova revolução, a quarta em sua história, podendo ser resumida em qualidade de informações, portanto necessita de sistemas confiáveis de processamento e armazenamento de dados, além de gerar dados em tempo real através do monitoramento e processamento constante. Fruto de anos de pesquisa e desenvolvimento, todo processo de digitalização, incluindo o *hardware* e *software* embarcados, só se remete realmente a uma revolução se tiver resultados obtidos de forma sistemática, além de qualidade comprovada (MAZZAFERRO, 2018).

Por meio de um cenário de um mundo digitalizado e tecnológico, residências inteligentes, que exercem tarefas complexas e interagindo com o meio interno da residência e seu usuário, vem atraindo cada vez mais adeptos. Mesmo com o avanço da qualidade da Internet e da computação pessoal, este sistema peca ainda com relação a transparência ao usuário e sua aplicação, afetando a segurança e conforto do usuário. Portanto, muito mais do que somente uma área ligada a tecnologia e computação, a domótica é uma área multidisciplinar, abrangendo áreas como Psicologia, Sociologia e Filosofia. (BOLZANI, 2004b).

Atualmente não há restrições para sistemas digitalizados e sua utilização, abrangendo desde sistemas hospitalares até sistemas residenciais, sendo cada vez mais popular devido ao acesso mais fácil e barato aos dispositivos de automação.

Esses sistemas permitem não apenas monitorar remotamente o estado de saúde (*home care*), mas também compensar deficiências funcionais (controle remoto de dispositivos e portas), prover segurança (alarmes de incêndio, de intrusão e mecanismos de desligamento automático para equipamentos como fogões e ferros de passar roupa) e facilitar o acesso aos meios de comunicação. (BOLZANI, Caio, 2012)

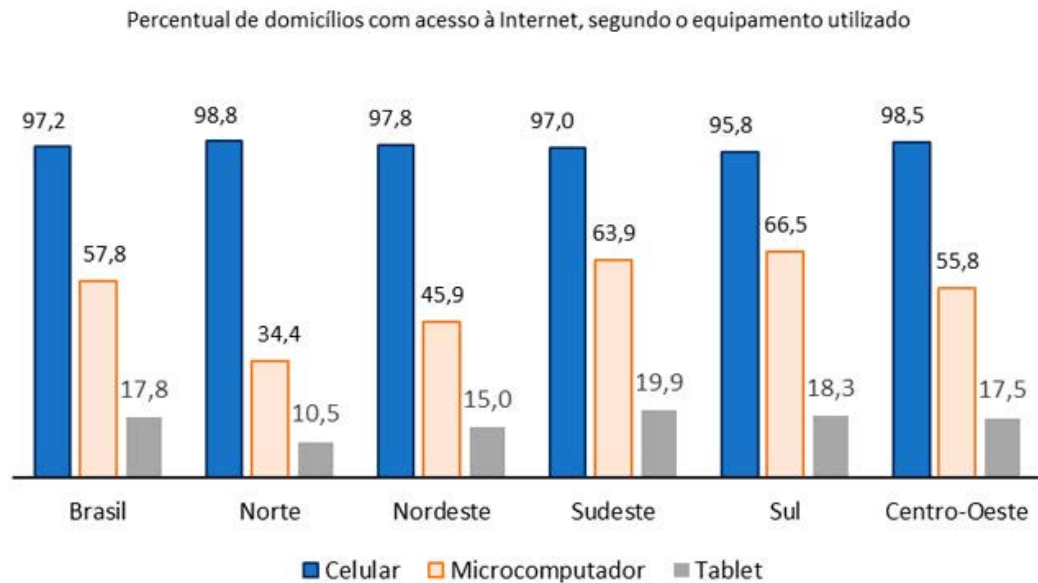
Com o crescimento das grandes metrópoles, o desenvolvimento de suas residências, transportes, economia, entre outras, nem sempre acompanham o crescimento real do local, ocasionando em problemas aos moradores, uma vez que falta, por exemplo, infraestrutura e

segurança. É sobre esse cenário urbano que atua a Internet das Coisas (IoT – *Internet of Things*), com o intuito de organizar as tecnologias de informações, os dados obtidos a partir delas, com o cotidiano dos indivíduos, buscando, com aporte tecnológico, transformar positivamente a sociedade, modernizando sistemas e automatizando processos, se tornando importante aliada na realização de tarefas diárias. (REVISTA BRASILEIRA DE GESTÃO URBANA. 2017).

Internet das Coisas é muito mais que apenas ligar lâmpadas pelo smartphone. Não é somente ligar as “coisas” pela internet, mas também torná-las inteligentes, capazes de coletar e processar informações do ambiente ou das redes às quais estão conectadas. A implementação de Internet das Coisas está mudando totalmente a forma como nos relacionamos com as coisas que estão ao nosso redor, transformando segurança, energia, meio ambiente, trânsito, mobilidade e logística. E o momento atual é propício a essa integração, visto que os dispositivos necessários, agregados as tecnologias para a integração e a conexão, alcançaram preços compatíveis com as “coisas” que se desejam integrar. (OLIVEIRA, 2017).

A Internet das Coisas está no centro das atenções devido a alta utilização de celulares e acesso a celulares atualmente, onde no Brasil, de acordo com pesquisa do IBGE atualizada no dia 10 de abril de 2018, 94,6% da população com 10 anos ou mais se conectaram a internet via celular (IBGE, 2018). Uma vez que o celular está implantado no cotidiano da população, o trabalho utilizará o mesmo para ser o intermediador do controle de automação, fazendo a comunicação Homem-Máquina, método já implantado por algumas marcas e empresas para comercializar produtos do mesmo segmento, porém com valores pouco acessíveis para a grande maioria da população, e o intuito deste trabalho é realizar um protótipo de baixo custo, para ser possível que qualquer pessoa com interesse possa utilizar da tecnologia desenvolvida.

Figura 1 – Índice de uso do celular no Brasil de acordo com o IBGE



Fonte: IBGE (2018)

Neste trabalho será apresentado um sistema de automação residencial inteligente que gera controles para a ação de atuadores, gerando dados e os armazenando, mostrando, assim, princípios básicos de automação residencial, internet das coisas, além de ser o elo de ligação para ações como a utilização de inteligência artificial para o aprendizado a partir do comportamento dos habitantes de uma residência (SGARBI, 2007), ou até mesmo, para sistemas de *health care*, onde se especifica e configura os elementos que consistem um sistema inteligente residencial (POTTER, 2013).

## 1.1 CONTEXTUALIZAÇÃO

Uma vez sabendo que o projeto se baseia no conceito de Internet das Coisas, como já enunciado, é importante saber como o Brasil e o mundo está lidando com a tecnologia, levando como base dados do BNDES, em parceria com o MCTIC, pode-se ter uma noção da grandeza da área, uma vez que a coleta, armazenamento e análise de dados podem contribuir para a expansão do universo relacionado ao IoT, auxiliando em tomadas de decisões empresariais, estudantis ou governamentais, em virtude que nem todos dados gerados são armazenados ou analisados. Gerando assim, um meio unificado de criação, gerando debates cada vez mais sofisticados baseados no Plano de IoT do BNDES, além de estar direcionado para os indicadores de usuários e suas soluções (BRASIL, 2018). Além de contar com o incentivo financeiro do Estado, no valor estimado de R\$ 0,3 milhões para investimento

inicial, e um saldo de operações de R\$ 0,2 milhões (BRASIL, 2018), demonstrando assim que a internet das coisas está altamente relacionado a conhecimento, uma vez que dados armazenados e analisados podem gerar conhecimento, e eventualmente, lucro para empresas e até mesmo para o governo.

Com relação a implementação do IoT para a automação residencial, vale ressaltar que uma residência inteligente e automatizada vai além do luxo ou apenas mais uma infraestrutura necessária para a residência, ela engloba a comunicação do sistema da residência com os dispositivos existentes, gerando uma integração dos sistemas (iluminação, ventilação, comunicação, entre outras), proporcionando segurança, conforto, modernidade, praticidade e economia (ALVES; MOTA, 2003), fator essencial em um mundo que cada vez mais se busca soluções eficientes, rápidas e baratas.

Por fim, porém um dos fatores cruciais para toda onda de pesquisa e desenvolvimento tecnológico relacionado a sistemas de automação, é a popularização dos *hardwares* embarcados, com foco deste projeto no Arduino, que vai além de ser somente um microcomputador, ele é uma plataforma de desenvolvimento de baixo custo, sendo acessível para todos (curiosos, entusiastas, estudantes, entre outros), e fornece uma gama recursos *Open Source*, facilitando ainda mais sua utilização. Sem contar com que a popularização deste dispositivo foi favorável para o seu desenvolvimento, uma vez que módulos adicionais e complementares começaram a ser desenvolvidos diretamente para o Arduino, o que levou ao seu barateamento no mercado (OLIVEIRA, 2017). Sendo assim que começa a motivação para o desenvolvimento deste projeto, baseando-se em uma tecnologia de alta inovação e em destaque, ligadas a sistemas cotidianos e de baixo custo, e tendo como fator principal ele ser uma fonte de recursos aberta, possibilitando o alcance a muitas pessoas.

## 1.2 MOTIVAÇÃO

Desde a década de 1990 já se pensava em interligar equipamentos cotidianos, em virtude da popularização da internet (OLIVEIRA, 2017), sem contar que houve uma quebra na barreira existente para troca de informações, uma vez implementado sistemas de redes de dados e telecomunicação (BOLZANI, 2004a). Filmes como, De Volta para o Futuro 2 em 1989, e A Casa Inteligente em 1999, abrangem o início da popularização da internet, até mesmo filmes recentes como Homem de Ferro lançado em 2008 demonstra a interação de pessoas com os equipamentos, além da interação entre os próprios equipamentos, consolidando a popularização da internet e sua vasta aplicação.

Uma vez com a internet estando consolidada, o mundo se encontra na Era da Computação Obíqua, da Internet das Coisas, onde gera-se uma enorme quantia de informações em nuvem, devido ao alto número de dispositivos (celulares, relógios, tablets, *notebook*, e muitos outros), agregando assim, mesmo sem perceber, uma gigantesca quantidade de serviços e interações (SILVA, parte 1, 2018). Em primeira instância buscou-se ter uma ideia do que se trata informação, pois deve-se retirar conhecimento dela, e registrou-se que a definição de informação é qualquer acontecimento, desde que, para o interlocutor, faça sentido (BOLZANI, 2004a). Considerando a possibilidade de ligar essas informações obtidas com a internet das coisas (sendo neste momento submetida a uma Cidade Inteligente, sendo um modo de gestão pública e interação dos cidadãos entre si e com a cidade), entra o conceito de Big Data, que nada mais é do que o conjunto de informações, melhoradas, obtidas a partir dos dados gerados, possibilitando assim, uma melhor análise e uma melhor tomada de decisão (SILVA, parte 2, 2018).

Porém, atingido uma abrangência como o conceito de Cidade Inteligente, ultrapassou o nível de interesse do autor, logo buscou-se como aplicar estes mesmos conceitos apresentados em uma área que estimulasse a curiosidade. A área encontrada foi a de Automação Residencial, possibilitando gerar uma Residência Inteligente, e aplicar em residências comuns, uma vez que o projeto se baseava em *Open Source* e baixo custo. Além do fato de ser um investimento baixo e com alto retorno, nos Estados Unidos tem-se uma noção que o orçamento de uma obra residencial, somente 10% do investimento é feito para a automação do domicílio, e com retorno de 30%, sendo este de médio prazo, e virtude da racionalização dos serviços de manutenção, como energia, água e gás (BOLZANI, 2004a).

Portanto, a motivação pela escolha do tema se deve ao fato da grande quantia de informações geradas no cotidiano que não se tornam dados relevantes, logo, o projeto se baseará em transformar as informações de uma residência em dados que possibilitam para os moradores uma maior economia, para que o investimento de baixo custo, tenha um retorno de baixo ou médio prazo, levando assim para os objetivos do projeto.

### 1.3 OBJETIVO

O objetivo deste trabalho é planejar e desenvolver um protótipo de sistema de automatização residencial com foco no baixo custo de desenvolvimento e aplicação, além de ser um projeto *Open Source*. Tendo como proposta de valor a segurança, tecnologia, economia e conforto para a vida domiciliar. Porém para se oferecer esse objetivo se faz

necessário definir canais de comunicação com o usuário, além de definir os recursos e atividades chaves para que se tenha sucesso na entrega da proposta de valor, levando a conclusão que trabalho realizado contém objetivos paralelos, que são:

- a descrição dos *hardwares* utilizados - características físicas, funcionamento, metodologia de controle, comunicação com outros dispositivos;
- o planejamento da rede doméstica – protocolos específicos para Internet das Coisas, utilização de máquina virtual;
- eficiência energética – consumo dos dispositivos de Internet das Coisas, soluções sustentáveis;
- atender as necessidades dos usuários – desenvolvimento de interface e descrição dos sistemas.

#### 1.4 ETAPAS DO TRABALHO

Este trabalho está dividido em três etapas:

- Etapa 1 – Conceitos Fundamentais, que abordará os fundamentos básicos de Domótica, Redes, IoT e Microprocessadores, descrevendo seu funcionamento e como será aplicado ao projeto, e contém 2 capítulos;
- Etapa 2 – Desenvolvimento, que descreverá a elaboração do projeto, contendo 1 capítulo;
- Etapa 3 – Resultados e Conclusão, abordando os resultados obtidos pelos testes, as conclusões que se pode retirar sobre a utilidade do produto obtido e como poderá ser melhorado, assim como as falhas obtidas, que contém 2 capítulos.

Como já informado, o projeto contém capítulos que constituem cada etapa do projeto, sendo eles:

- Capítulo 1 – Introdução, que é o capítulo que constitui esta introdução e compõe a Etapa 1;
- Capítulo 2 – Automação Residencial, que abordará os conceitos fundamentais sobre os dispositivos de uma residência automatizada, assim como os componentes de rede e eletroeletrônicos que a ela pertence, além de familiarizar o leitor aos termos técnicos contidos no texto. Sendo assim, finalizando a Etapa 1 do projeto.
- Capítulo 3 – Pesquisa e Desenvolvimento, abrange toda a concepção do projeto, além de descrever o passo a passo de sua elaboração, os componentes necessários para sua

elaboração, uma estimativa de preço, além de um tutorial elaborado para que qualquer entusiasta consiga reproduzir a ideia. Sendo esta, a Etapa 2 do projeto.

- Capítulo 4 – Resultados, constitui o funcionamento real do projeto, descrevendo os erros que ele apresenta e possíveis melhorias, iniciando a Etapa 3 do projeto.
- Capítulo 5 – Conclusão, que finaliza o projeto concluindo as etapas descrevendo a interação da teoria com a prática e se o objetivo inicial do projeto foi obtido.

## 2 AUTOMAÇÃO RESIDENCIAL

A modernização sempre foi um ramo de diversos tipos de estudos, há quem prefere focar na modernização industrial e comercial, porém há quem opta pelo estudo nos impactos sociais, uma vez que há quem diz que a modernização causa mudança de cultura, por exemplo, logo os estudiosos que focam nestes impactos, em sua grande maioria, concordam que a modernização cita as coordenadas das mudanças (FISCHER, 1992).

Mesmo sem saber, grande parte da população já se encontra utilizando dispositivos com grandes inovações tecnológicas, se equiparando a equipamentos robóticos, porém, como já citado, muitas temem o termo robótica, por se tratar de uma grande inovação tecnológica, como por exemplo a substituição do trabalho humano por uma máquina, onde Larry Page, cofundador e CEO do Google, em entrevista ao Financial Times em outubro de 2014, defende que cerca de 90% do trabalho feito atualmente por trabalhadores humanos será realizado por robôs nos próximos anos. Portanto, utilizando uma das frases de Thich Nhat Hanh “As pessoas têm dificuldade de abandonar seus sofrimentos. Por medo do desconhecido, elas preferem sofrer com o que é familiar.”, mais os tópicos já citados, é possível concluir que a implantação de sistemas ligados a modernização, incluindo o tema deste trabalho, a Domótica, sofrem muita rejeição ao longo dos anos.

Computadores, smartphones, *notebooks*, eletrodomésticos no geral contém sistemas computadorizados em sua essência, fazendo o controle do dispositivo através da leitura de uma ação humana, auxiliando na solução de problemas cotidianos, apoiando a ideia de que uma nova tecnologia pode ser construtiva para a sociedade, e indo contrário aos mais extremistas, como os ludistas que eram pessoas adeptas de um dos seus líderes Ned Ludd, e que eram contrários a mecanização durante a Revolução Industrial na Inglaterra. Portanto, neste contexto residências inteligentes podem ser incluídas como uma ferramenta para inspecionar tarefas cotidianas, como monitoramento de crianças, ou até mesmo auxiliar na realização de tarefas, tornando-as mais rápidas e prazerosas (BOLZANI, 2004a).

Atualmente ao se utilizar eletrodomésticos de uso rotineiro não há uma comunicação entre os dispositivos, nem ao menos uma comunicação do dispositivo para saber o seu melhor funcionamento, como por exemplo um forno com a função de temporizador em seu sistema, não indica ao usuário se ele programou no tempo correto de preparo do alimento inserido, ou até mesmo uma janela que não abre ou fecha conforme a intensidade que o sol está refletindo, são tarefas que a domótica tenta, aos poucos, atingir e aprimorar, para tornar cada vez mais acessível ao consumidor, possibilitando, assim, um controle facilitado e absoluto, além de

estável e prático, levando em consideração o comportamento descrito para residências inteligentes, é notório uma dependência de comunicação contínua entre usuário, sistema, ambiente e a função, pré-determinado, a ser exercida, levando a necessidade de existir um sistema com todos os aspectos devidamente descritos simultaneamente, para que se torne capaz a definição do comportamento correto de uma residência inteligente (BOLZANI, 2004a).

## 2.1 ORIGEM E PRINCÍPIOS DA DOMÓTICA

Domótica, é a palavra que representa o controle automático de uma casa, oriunda da junção das palavras em latim *Domus*, que significa casa, e da palavra em francês *Imotique*, ou Robótica, que pode ser entendido como o controle automático de algo, podendo assim, chamar de maneira abrangente, a casa de Residência Inteligente (ALVES; MOTA, 2003).

Deve-se dizer, então, que uma residência ou edifício, não é necessariamente inteligente, mas sim a maneira pela qual é feita a utilização de seus recursos, portanto é coerente chamar de residências que agrupam características que facilitam a vida daqueles que ali habitam, como por exemplo: segurança, economia, conforto, ecologia e principalmente a integração entre os sistemas domóticos utilizados e os dispositivos de controle já existentes (ALVES; MOTA, 2003).

A palavra inteligente quando é aplicada a residências almeja dizer que não é somente capaz de realizar funções lógicas, preestabelecidas ou não, mas também de aprender e, ou, sugerir melhorias para os ambientes (ALVES; MOTA, 2003), no caso deste projeto o sistema aprenderá através do algoritmo de controle usado. Portanto, os dispositivos ou o sistema completo tem um grau de autonomia, tomada de decisão e inferência nas ações da residência, como iluminação, o som presente no local ou climatização, de acordo com a utilização do usuário residente (BOLZANI, 2004b).

A automação de uma residência, ou o processo de torná-la inteligente, poderá ajudar em tarefas diárias para economizar tempo ou evitar preocupações como fechar uma janela em um dia propenso a chuva. Porém o quanto será impactante na vida dos usuários dependerá do tipo de vida, dos gostos pessoais e recursos disponíveis.

Uma residência inteligente se difere das usuais por oferecer:

- Versatilidade - Tem a capacidade de se comunicar com todos os sistemas, além de uma fácil interface de relação com o usuário usando por exemplo o comando por voz, ou com telas sensíveis ao toque (SILVA, 2009);

- Economia – Utiliza conceitos modernos e ecológicos em sua arquitetura, a fim de obter o uso mais próximo do ideal dos recursos naturais disponíveis: energia, luminosidade, aquecimento e resfriamento, ventilação. Portanto, é capaz de reduzir de maneira significativa o consumo de energia elétrica. Além de permitir, desde que a concessionária permita, um monitoramento em tempo real das informações sobre o consumo de bens essenciais de uma residência, logo evita desperdício, ou até mesmo permite perceber alguma falha no sistema instalado na residência, como vazamento de substâncias líquidas ou gasosas (SILVA, 2009);
- Conforto – Oferece funções de gestão para a residência ou pessoal não encontrada nas residências usuais, pois, de forma automática ou pré-programadas, permite a realização de tarefas, evitando assim ordens diretas (SILVA, 2009);
- Segurança – É um dos principais objetivos e impulsionadores da automação residencial (SILVA, 2009). Podendo conter câmeras e alarmes ligados ao sistema de controle, além de sistemas de acionamento de trancas e alarmes de forma remota.

A domótica tornou-se realidade à escala mundial, devido a constante modificação dos sistemas de comunicação, equipamentos eletrodomésticos, que levam a uma maior integração das pessoas com o ambiente pelo qual reside ou ocupa, mudando de vez a forma pelo qual utilizamos os recursos do mesmo. Conforme a interação da sociedade com os equipamentos e residências inteligentes, a área de atuação deixará de ser restrita a produtores de tecnologia eletroeletrônica e de telecomunicação, partindo para uma vasta lista de possibilidades, como arquitetura, design e ciências humanas (ALVES; MOTA, 2003).

A realidade, portanto, vem se aproximando da ficção científica recente, levantando questões filosóficas e éticas, visando sempre a capacidade ou incapacidade da sociedade em se adaptar as novas formas de comunicações, como por exemplo quando se refere ao comportamento uma vez que não existe ainda um padrão social estabelecido ou legislaturas que suportem todas as novidades. Gerando, assim, novos desafios para transformar as novas tecnologias em algo favorável a sociedade, sem restringir sua liberdade, além de prevenir ao máximo que terceiros utilizem as informações de uma residência inteligente com a intenção de prejudicar os moradores (Alves; Mota, 2003).

A domótica, portanto, não deve fugir de questões sociais, e deve focar em uma forma inteligente e de reflexão profunda, a seguir um caminho ético visando o conforto e segurança dos usuários, pois um sistema residencial inteligente irá, cada vez mais, representar os valores, cotidiano, relação familiar, relação com o meio ambiente dos usuários (Alves; Mota, 2003).

## 2.2 INTERNET DAS COISAS, ORIGEM E PRINCÍPIOS

Um fator determinante durante a concepção de um projeto que demanda uma tecnologia avançada é o planejamento de seu sistema de informação. Logo, na implementação de uma residência inteligente pressupõe temas como a organização do sistema de informática, tanto interno a residência quanto ao externo, para se ter uma melhor integração dos serviços a serem implementados que gerarão um novo valor agregado. Porém, o nível de automação a ser atingida limita-se aos sonhos e capacidade financeira do usuário, limitando, assim, muitas vezes a sofisticação tecnológica implementada. Para atender a todos os fatores, já citados, que diferenciam uma casa automatizada de uma residência padrão, escolheu-se a utilização do método da internet das coisas (BOLZANI, 2004b).

Baseada no princípio da tecnologia de detecção de objetos, originada durante a Segunda Guerra Mundial e em virtude dos avanços científicos sobre eletromagnetismo da época, tem como primeira patente o popularmente conhecido como RADAR (OLIVEIRA, 2017), que primeiramente surge como uma abreviação de detecção e identificação via rádio de algum objeto, porém com sua popularização tornou-se uma palavra contida em dicionários, com significado de que é um aparelho com função para assinalar, pela reflexão de ondas hertzianas ultracurtas, os objetos afastados e determinar a sua localização exata (AURELIO, 2018). Durante a guerra, os países em conflito usavam os radares, ou transponders, para alertar sobre a aproximação de aviões, porém sem identificar se era inimigo ou aliado. Porém os alemães buscaram resolver esse empecilho e descobriram que ao girar os aviões, no sentido de suas bases, o sinal refletido era alterado e tornando possível a identificação dos aliados, ou não. Sendo este o primeiro sistema de RFID registrado, além de ser a primeira tecnologia associada a internet das coisas (OLIVEIRA, 2017).

Posteriormente IoT foi ligado a Redes de Sensores sem Fio (RSSF), que diferem dos computadores usuais pois utiliza o método de rede composta, contendo nós distribuídos com capacidade de atuar sem a utilização de fios, necessitando de fontes de energia para seu funcionamento, além de possuir mecanismos para autoconfiguração. A utilização prevista para esta tecnologia é o monitoramento em situações de extrema necessidade, como guerras. Interessante sobre este método foi que pela primeira vez na história buscou-se viabilizar dispositivos de comunicação sem fio com valores extremamente acessíveis (TAVARES, 2002).

Em conjunto com o RSSF, devido a necessidade da padronização dos métodos de comunicação utilizados pelos dispositivos, surge os protocolos TCP/IP e, posteriormente, o WiFi, que possibilitaram a comunicação utilizando a internet, junto com essas tecnologias, as redes de celulares conhecidas como: 2G, 3G, 4G, a popularização de redes sem fio se fez completa e com protocolos de comunicação definidos, tornando o acesso sem fio a internet algo trivial e rotineiro.

A internet das coisas foi primeiramente uma ideia conhecida, na década de 1970, como internet embutida ou computação e somente em 1999, Kevin Ashton durante seu trabalho na Procter & Gamble, denominou por *Internet of Things*, no intuito de etiquetar eletronicamente os produtos da empresa.

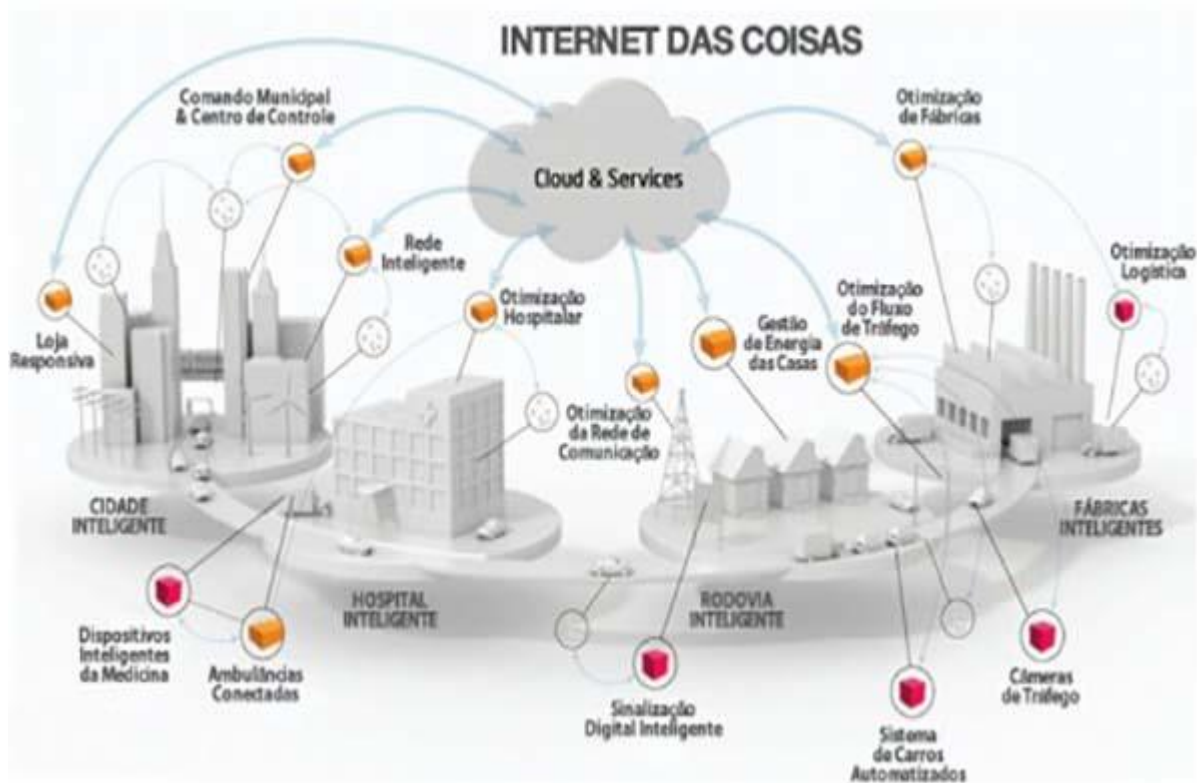
Mas o que eu quis dizer à época, e ainda considero isso válido, se baseia na ideia de que estamos presenciando o momento em que duas redes distintas – a rede de comunicações humana (exemplificada na internet) e o mundo real das coisas – precisam se encontrar. Um ponto de encontro onde não mais apenas “usaremos um computador”, mas onde o “computador se use” independentemente, de modo a tornar a vida mais eficiente. Os objetos – as “coisas” – estarão conectados entre si e em rede, de modo inteligente, e passarão a “sentir” o mundo ao redor e a interagir. (FINEP, 2015).

Se faz necessário saber que o IoT é um *framework*, ou seja, é a união de diversos *softwares* para formar uma funcionalidade em comum, logo, não é um dispositivo nem uma tecnologia específica, logo não deriva de tecnologias existentes e sim as utiliza para existir. proporcionando que a utilização não seja restringida, porém torna necessário a criação de novos métodos de comunicação, uma vez que cada dispositivo terá um padrão diferente e precisarão coexistir (FACCIONI FILHO, 2016).

Em âmbito corporativo, a internet das coisas é vista com bons olhos pois pode transformar a manutenção em algo prescritivo e preditivo, melhorar a gestão de ativos e do desempenho devido ao baixo custo de sensores e de estarem conectados a nuvem, facilitando assim a obtenção dos dados, portanto, aumenta os dados disponibilizados, além de um controle industrial inteligente, ou seja, aumenta a flexibilidade e rentabilidade dos processos de manufatura da empresa (MADEIRA, 2016).

Porém como é um *framework* capaz de abranger qualquer tipo de indústria, objeto, função, sua utilização, arquitetura, protocolos ainda não são muito bem definidos, logo um dos desafios principais que deve ser resolvido é o de integrar de uma maneira inteligente os mais diversificados padrões de sistemas e dispositivos. (MADEIRA, 2016).

Figura 2 – Previsão de uso do IoT



Fonte: Rockheine (2018)

### 2.3 PROTOCOLOS DE REDES, PRINCÍPIOS E TIPOS

Antes mesmo de se falar sobre a necessidade da criação de novos protocolos de comunicação para a internet das coisas, deve-se primeiro definir resumidamente o que são protocolos, posteriormente citar os mais populares atualmente e por fim citar quais são os protocolos criados ou adaptados para a utilização de internet das coisas.

Portanto, protocolos não existem somente para computadores, mas também para interação humana, como por exemplo existem maneiras comuns de se relacionar com outras pessoas, de acordo com sua idade, idioma, entre outros fatores. Os computadores, não se são diferentes, eles se comunicam entre si e com o usuário. Entre dois ou em uma rede de computadores, deve existir protocolos que permitem a comunicação entre essas máquinas, uma vez que nem todas estão utilizando o mesmo sistema operacional. Porém um protocolo não serve somente para comunicação, ele pode transformar uma tarefa complicada em várias fáceis, tornando mais rápido a realização da tarefa inicial (BOLZANI, 2004b).

A linguagem de comunicação e conexão mais comum utilizada entre computadores é a TCP/IP, que mesmo parecendo ser somente dois protocolos, o TCP e o IP, é um conjunto de protocolos que permitem computadores de se conectarem a internet (LEIDEN; WILENSKY,

2009). Onde, o protocolo capta os pacotes trafegando pela rede mundial de computadores e adiciona a eles um endereçamento virtual, conhecido como IP, que contém todas as informações necessárias para que o pacote seja entregue ao destinatário, podendo ser feita uma analogia a cartas ou pacotes enviados pelos correios, onde se faz necessário selos com todos as informações do destino para que seja possível a entrega correta (PISA, 2012). Esta parte da comunicação e protocolos é conhecida como camada da Internet.

A camada da Internet direciona as informações para o destino, porém sem garantir que os pacotes cheguem de forma correta, ou seja, em sua ordem inicial, ou até mesmo, ela não permite que seja captada erros durante a entrega dos pacotes. Isto se deve ao fato que é um dos trabalhos da camada de transporte e não da camada da internet. O TCP funciona na camada de transporte para garantir que os pacotes cheguem na ordem correta e sem apresentar erros, portanto, a internet não seria confiável sem essa camada. A UDP também trabalha na camada de transporte e compartilha uma função com TCP: o de mover os dados para a próxima camada, porém como os processos as vezes requerem uma velocidade de processamento muito rápido, a UDP não tem a função de indicar se houve erros na entrega do pacote (LEIDEN; WILENSKY, 2009).

Figura 3 – Protocolo TCP/IP



Fonte: Devmedia (2018)

Por fim, a comunicação para sistemas de internet das coisas mesmo que ainda não especificados, já existem alguns protocolos adaptados ou criados para essa finalidade, onde

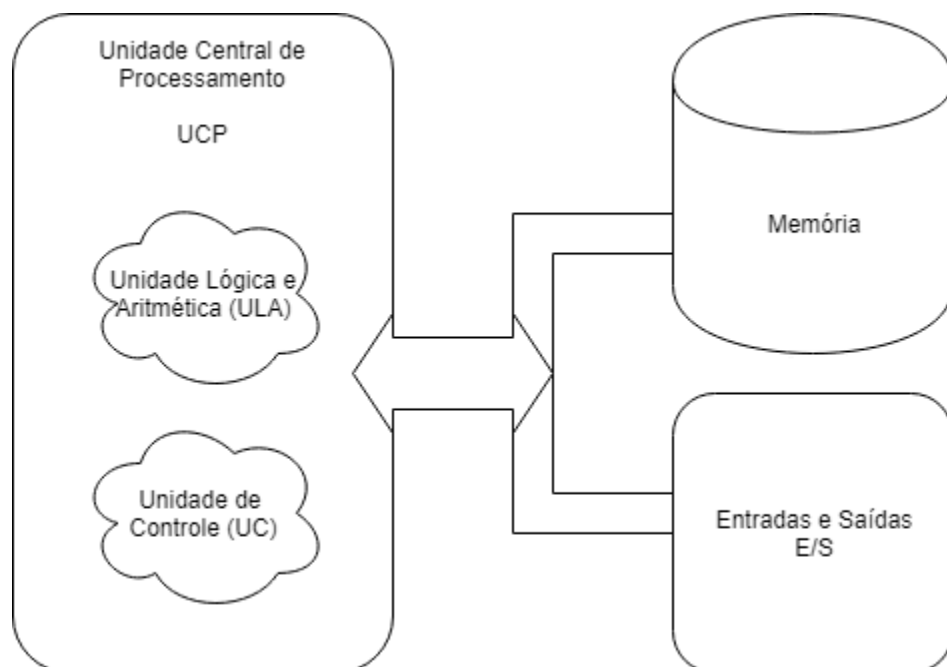
alguns clássicos se destacam como o WiFi e Bluetooth, ou ainda aqueles que estão em ascensão, como Thread e Z-Wave que são mais focados para automação residencial (VIDAL, 2017). Portanto para um maior conhecimento dos protocolos de rede designados ao IoT, serão devidamente detalhados a seguir:

- Thread: Desenvolvido para domótica de baixo consumo energético, sendo complementar ao sistema WiFi, e lançado em 2004 pelo Thread Group, e atualmente vários fabricantes já possuem chips registrados (VIDAL, 2017);
- Z-Wave: Outro protocolo desenvolvido especificamente para automação residencial, se diferenciando por ser um protocolo de fácil integração e podendo operar com alcance de até 30 metros (VIDAL, 2017);
- NFC: Tecnologia utilizada para troca de informações entre dois equipamentos eletrônicos, atualmente muito utilizado por celulares, abrangendo desde troca de arquivos entre celulares, até mesmo pagamento, onde substitui a utilização de cartões de débito e crédito (VIDAL, 2017);
- LoRaWAN: Focada para redes de acesso de até 15 quilômetros de distância (dependendo da topologia de instalação), foi desenvolvida com foco no baixo custo energético e com alto nível de segurança. Tendo como aplicações mais óbvias a tarifação remota de serviços como água, luz e gás de uma residência, por exemplo (VIDAL, 2017);
- WiFi: Talvez o método mais popular de comunicação com a internet nos dias atuais, estando presente na maioria das residências e estabelecimentos, é um tipo de comunicação de baixa distância, se caracterizando como protocolo LAN com capacidade de gerenciar altas taxas de transmissão. Em contrapartida, os modelos mais usados apesar das altas taxas de velocidade de transmissão, possuem consumo energético que podem inviabilizar seu uso em algumas aplicações de IoT (VIDAL, 2017);
- Bluetooth: Protocolo com popularidade podendo ser comparado ao WiFi, e com diversas versões já lançadas, porém em suas versões mais recentes se destaca o baixo consumo energético viabilizando a utilização em aplicações de internet das coisas. O alcance é relativamente baixo, logo é caracterizado como LAN (VIDAL, 2017);
- Zigbee: Protocolo com alta utilização em âmbito corporativo, e com baixo uso de forma residencial. É basicamente um padrão de comunicação wireless, sem mudanças constantes nas taxas de transmissão (VIDAL, 2017);

## 2.4 MICROCOMPUTADORES E SUAS ARQUITETURAS

A escolha de um microcontrolador para o seu projeto é de suma importância. Para que seja possível realizar esta escolha, deve-se saber, de maneira resumida, o seu funcionamento, onde pode-se entender que é um integrador de *hardware* e *software*. Porém, de uma maneira um pouco mais técnica, o microcontrolador é um computador de tamanho reduzido por ser implantado em um único chip, para a realização de funções específicas (GARCIA, 2018).

Figura 4 – Arquitetura de Microcomputadores



Fonte: Garcia (2018)

Assim como computadores domésticos, um MCU, maneira pela qual se pode designar um microcontrolador, contém uma CPU, ou processador, e periféricos como a memória, podendo ser dividida entre memória de dados (RAM) e memória de programas (EPROM ou Flash), além de pinos de entradas e saídas que comunicam o microcontrolador a componentes eletrônicos externos, podendo ser digitais ou analógicos, permitindo, assim, a utilização de praticamente todos os tipos de sensores existentes, bastando se adaptar para a função estabelecida no projeto (GARCIA, 2018).

Para se definir o microcontrolador ideal para um projeto, deve-se levar em consideração critérios como: arquitetura, consumo, periféricos, velocidade e capacidade de armazenamento,

preço, suporte e disponibilidade (LIMA, 2014). Para esse projeto primeiro irá ser detalhado o MCU escolhido para realizar a função de núcleo da automação residencial, para posteriormente ser detalhado a maneira que atende aos critérios já citados para ter sido escolhido.

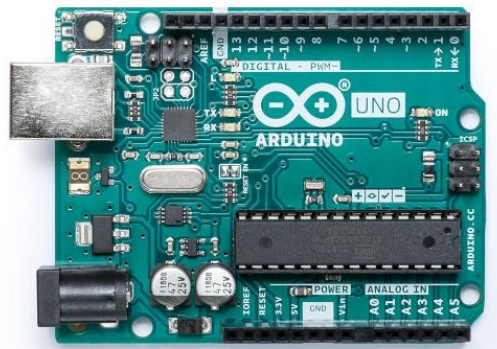
### 2.4.1 Arduino

Para o trabalho proposto, a princípio, foi escolhido a utilização da plataforma de microcontrolador Arduino, principalmente por ser uma plataforma baseada na facilidade de utilizar seu hardware e software, além da sua natureza aberta, tornando-o uma ótima opção para projetos eletrônicos, possibilitando até mesmo a criação caseira ou profissional de um Arduino genérico, uma vez que as placas de circuito impresso e os diagramas eletrônicos estão disponibilizados publicamente, criando assim um universo vasto, que abrange desde entusiastas até profissionais, oferecendo um produto barato (MONK, 2013).

No mercado existem diversas placas com a plataforma Arduino, sendo a mais popular delas o Arduino UNO (MONK, 2013), porém existem outros modelos que utilizam métodos de programação e conexões com o exterior de forma similar, sendo elas o Arduino MEGA, o Arduino Leonardo e o Arduino Nano, onde basicamente varia-se o tamanho da placa, a quantidade de entradas e saídas, a memória interna do MCU.

Tomando a placa do Arduino UNO como base para explicação da plataforma Arduino, por ser a plataforma mais usada e por ser a placa recomendada, pela própria desenvolvedora, para os novatos que desejam começar com eletrônica e programação, este trabalho começará a tratar as especificações técnicas do dispositivo.

Figura 5 – Arduino UNO



Fonte: Arduino (2018)

A placa pode ser alimentada tanto pela conexão USB quanto pela fonte de alimentação de conector Jack, com limite de 6-20 Volts, porém é considerado que acima de 12 Volts de alimentação pode danificar a placa pelo sobreaquecimento do regulador de tensão contido, e abaixo de 7 Volts a tensão de funcionamento do sistema, que é de 5 Volts, pode ficar instável (SOUZA, 2013).

Os conectores de alimentação contidos servem para a conexão de módulos externos adicionais a placa, podendo conter função diversas. A função de cada pino é descrita como:

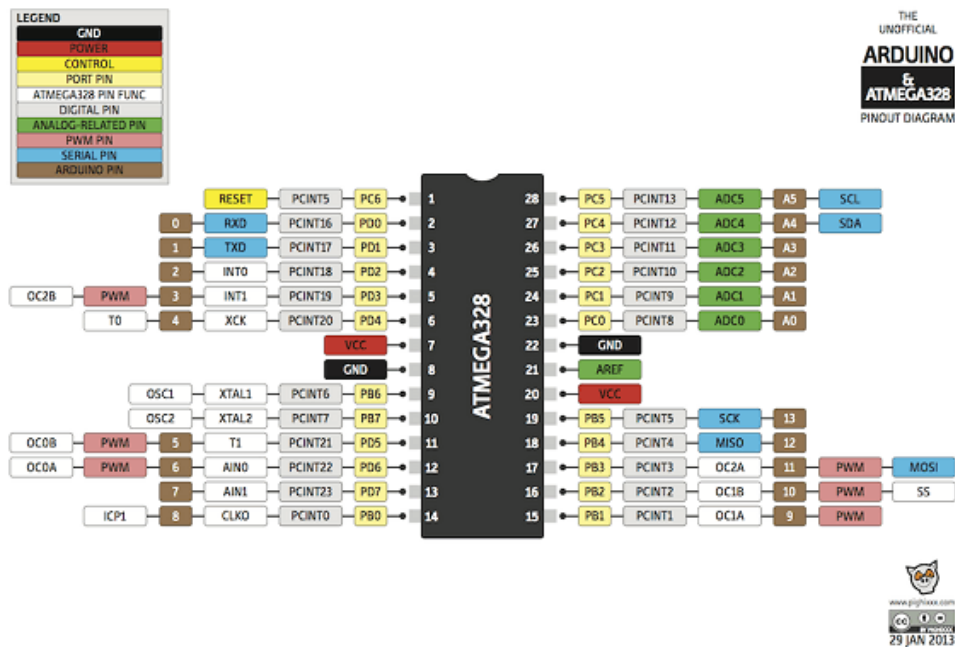
- IOREF – Possibilita a utilização de placas adicionais ao Arduino que funcionam com tensão de alimentação de 3,3V e 5V (SOUZA, 2013).
- RESET – Interligado ao pino interno, de mesmo nome, do microcontrolador, podendo ser utilizado para um *reset* externo ao Arduino (SOUZA, 2013).
- 3,3 V – Pino que fornece tensão de saída de 3,3 Volts para módulos externos, e corrente máxima de 50 mA (SOUZA, 2013).
- 5 V - Pino que fornece tensão de saída de 5 Volts para módulos externos (SOUZA, 2013).
- GND – Terra da placa, ou pino de referência (SOUZA, 2013).
- VIN – Utilizado para alimentar a placa do Arduino UNO com fontes externas, através de módulos(*shields*) ou baterias. Quando a placa é alimentada através do conector de fonte externa (Jack), a tensão da fonte estará contida nesse pino (SOUZA, 2013).

Os conectores analógicos da placa são ao todo 6 e com resolução de 10 bits cada. Por convenção do conversor analógico digital, os pinos estão alimentados com 5 Volts e a conversão analógica digital será de valor 1023 (SOUZA, 2013).

A placa do Arduino UNO contém 14 pinos digitais, podendo serem configurados para serem de entrada ou saída, fornecendo 5 Volts de alimentação e suportando no máximo uma corrente de 40 mA (SOUZA, 2013). Alguns pinos contêm funções especiais, como:

- PWM – Os pinos 3, 5, 6, 9, 10 e 11 podem ser utilizados como saídas PWM, ou seja, são capazes de fornecer sinal digital de frequência fixa e largura de pulso variável, comportando-se na prática como um sinal analógico (SOUZA, 2013).
- Comunicação Serial - Os pinos 0 (RX) e 1 (TX) são utilizados para comunicação serial e que contém comunicação direta com o microcontrolador responsável pela comunicação entre o USB e o computador (SOUZA, 2013).
- Interrupção Externa – Os pinos 2 e 3 podem ser utilizados para gerar interrupção externa através de uma configuração de software (SOUZA, 2013).

Figura 6 - Pinagem ATmega328P



Fonte: Efetividade (2018)

#### 2.4.2 ESP8266

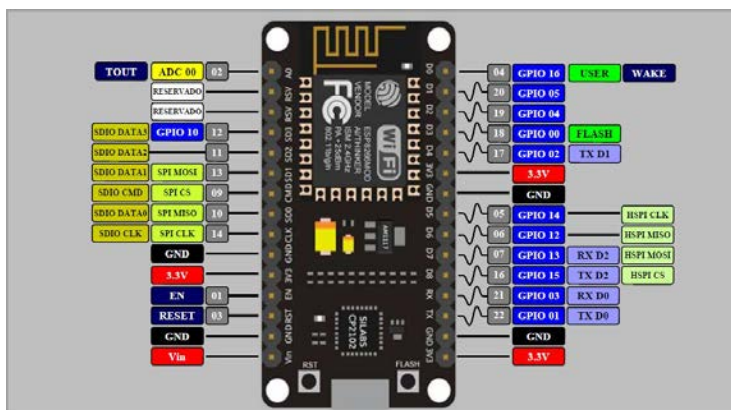
Portanto, o Arduino UNO é uma placa que contém funções de projeto que chamam atenção, mas como a intenção é utilizar os conceitos de internet das coisas, se faz necessário a comunicação com a internet, seja por WiFi ou por cabo Ethernet, função pelo qual não está contido no Arduino UNO.

Baseando-se em possíveis formas de se obter comunicação com a internet utilizando o Arduino como plataforma principal, o microcontrolador ESP8266 se destacou, de origem chinesa, da fabricante Espressif, é um microcontrolador de 32 bits e com tensão de funcionamento de 3,3 Volts (CURVELLO, 2015). Com relação aos pinos de entradas e saídas, esse microcontrolador contém 17 interfaces que podem ser utilizados como entradas e saídas, além de conter 4 interfaces de PWM e 1 interface analógica com 10 bits de precisão (OLIVEIRA, 2017).

Para facilitar a utilização desse chip, várias fabricantes produzem placas e módulos de desenvolvimento, variando de acordo com seu tamanho, quantidade de pinos e tipo de comunicação com o computador. Os modelos de módulos e placas de desenvolvimento que utilizam o microcontrolador ESP8266 são:

- ESP-01: É o módulo mais simples disponível, contendo em sua composição o microcontrolador, uma memória externa de 512 KB ou 1 MB, e 8 pinos que são compostos por alimentação de 3,3 Volts e comunicação UART. Uma dificuldade que sua utilização dispõe é que para a gravação de software se faz necessário adaptadores USB (OLIVEIRA, 2017).
- ESP 12: Módulo construído com o intuito de ser integrado em placas de circuitos impressos, possibilitando, assim, a integração de componentes extras, como conversor USB-serial para facilitar a comunicação com o computador para gravar o código fonte, ou até mesmo regulador de tensão que permite a utilização de fontes de alimentação externas, como a conexão direta com o USB do computador, uma vez que esta fornece 5 Volts e o módulo tem tensão nominal máxima de 3,3 Volts. Existe dois tipos desse módulo, o ESP-12E e ESP-12F, que variam somente na localização dos pinos. A plataforma de desenvolvimento mais utilizada com a aplicação desse módulo é a NodeMCU, pois além de ser possível utilizar o mesmo ambiente de desenvolvimento do Arduino, ele também permite o desenvolvimento de projetos utilizando a linguagem Lua que nasceu e cresceu no Grupo de Tecnologia em Computação Gráfica da Pontifícia Universidade Católica do Rio de Janeiro, combinando programação procedural com tabelas associativas e semântica extensível, se destacando pela fácil aprendizagem e integração, além de ser gratuita (OLIVEIRA, 2017).
  - NodeMCU: Placa de desenvolvimento que abrange todas as funcionalidades do microprocessador ESP8266, porém com conversor Serial e regulador de tensão próprio, além de conter pinos digitais para entrada e saída de sinais, 1 pino analógico, permite comunicação I2C, SPI entre outros (OLIVEIRA, 2017).

Figura 7 - NodeMCU



Fonte: Ifuturetech (2018)

A escolha do módulo ESP12 na placa de desenvolvimento NodeMCU como base das aplicações desse projeto se devem ao fato do preço ser acessível, encontrado facilmente em lojas chinesas por cerca de 3 dólares (com adicional de frete) e em lojas brasileiras por valores de 25 a 50 reais, além de utilizar a mesma plataforma de desenvolvimento de software do Arduino UNO, bastando a instalação da biblioteca adicional do ESP8266, assim como os módulos adicionais populares do Arduino são compatíveis com o microcontrolador fabricado pela Espressif, e por fim, como a placa de desenvolvimento contém limitações de hardware, como falta de pinos ou de memória, porém por ser possível a comunicação I2C se faz possível a expansão de pinos, memórias ou adição de funções extras, uma vez que seja realizado a comunicação com alguma placa da plataforma Arduino, por exemplo.

## 2.5 GESTÃO DE ENERGIA

Como já informado anteriormente uma das características que tornam uma residência apta a ser nomeada como inteligente é o fato dela ser eficaz energeticamente e ser econômica, não somente no ponto de vista ecológico, porém no ponto de vista financeiro com os gastos mensais que o usuário deverá pagar. Utilizando essas informações como base, deve-se fazer um levantamento a respeito do consumo energético dos componentes utilizados para a automação da residência para saber se eles se enquadram nos quesitos de economia e conseguir atingir o objetivo proposto de uma residência automatizada com consumo energético reduzido.

Na grande maioria dos dispositivos de internet das coisas, o consumo de energia é constante e elevado, o que inviabiliza o uso de baterias e faz com que muitas vezes sejam conectados à rede comum fornecida pelas concessionárias de energia, em tensão de 127V e 220V, que pode ser convertida facilmente para o propósito descrito, com a utilização de fontes de conversão assim como utilizados em telefones celulares. Porém sabe-se que dispositivos de internet das coisas que se conectam via WiFi consomem energia, pelo menos, em uma ordem acima do que os dispositivos que se comunicam pela rede padrão pessoal e sem fio do IEEE 802.15.4 (OLIVEIRA, 2017).

Nos módulos ESP8266 o gasto com energia será focado basicamente no seu microcontrolador, por ser o único componente ativo do módulo. Onde este, por sua vez, em seu funcionamento consome (OLIVEIRA, 2017):

- Transmitindo:
  - 17 dBm – 170 mA;

- 15 dBm – 140 mA;
- 13 dBm – 120 mA.
- Recebendo:
  - 80 dBm – 50 mA;
  - -65 dBm – 56 mA.

Porém deixar um microcontrolador ligado em seu modo de atividade normal ativo com tempo ocioso consome muita energia elétrica torna o consumo ainda maior. Para isso existe os *Sleep Modes*, que adormecem certas funções do MCU, aumentando a possibilidade de seu uso, são eles (OLIVEIRA, 2017):

- Modo modem - *sleep* - Desliga somente o rádio WiFi do MCU, porém não de forma completa, somente gerencia os *Beacons*, que são os pacotes enviados pelo dispositivo para acessar a rede sem fio, do seu roteador e desliga entre intervalos definidos. Normalmente utilizado quando o envio de informações se faz em intervalos de tempos predeterminados e que é preciso para o funcionamento do sistema em geral (OLIVEIRA, 2017).
  - Consumo – 15 mA.
- Modo light - *sleep* – Desliga o rádio WiFi e o *Clock* interno. A CPU fica pendente, respondendo a sinais de entrada e saída, de acordo com o programado em *software*. Esse modo geralmente é utilizado quando se faz uma ação somente na presença de algum evento de entrada e saída (OLIVEIRA, 2017).
  - Consumo – 0,5 mA.
- Modo *deep – sleep* – Este é o modo que melhor gerencia na questão de consumo, uma vez que praticamente todo MCU fica desligado durante um determinado período de tempo, sendo que quando a CPU reinicia se faz necessário realizar todos os procedimentos de configuração e conexão (OLIVEIRA, 2017).
  - Consumo – 10  $\mu$ A.

A importância dos modos *sleeps* é tamanha, que pode ser representado com números, onde em projetos pelos quais não é possível utilizar a rede elétrica comum se faz necessário a utilização de bateria. Tomando como base a utilização de carregadores portáteis de celulares, por serem extremamente convencionais no mercado, e escolhendo um modelo de 10 Ah, novamente escolhido por ser um dos modelos mais populares (OLIVEIRA, 2017), podemos estimar:

- Pior Caso: Maior consumo possível, quando está somente transmitindo informações e consumindo em média 170 mA, a bateria teria uma autonomia, até precisar ser recarregada, de aproximadamente 58 horas, ou, 2 dias e meio (OLIVEIRA, 2017).
- Melhor Caso: Menor consumo possível, sem a utilização do modo *sleep*, quando está somente recebendo informação e consumindo em média 50 mA, a bateria teria uma autonomia de aproximadamente 200 horas (OLIVEIRA, 2017).
- Modo *Sleep* Ativado: se for usado o modo *deep sleep*, para evidenciar a importância da utilização deste modo para um menor consumo energético, se utilizará em média 10  $\mu$ A e a autonomia da bateria será de aproximadamente 1 milhão de horas, sendo o equivalente a 114 anos sem precisar carregar ou substituir as baterias, claro que não está sendo levado em consideração a degradação das baterias com o tempo ou sua vida útil, esse exemplo é pra evidenciar somente o consumo energético (OLIVEIRA, 2017).

Por fim, pode ser estabelecido que há um aumento do consumo energético quando utiliza o microcontrolador do ESP8266, mesmo que podendo ser utilizadas técnicas que tornam esse consumo quase que ínfimo, deve-se levar em conta que automação residencial proposta pelo projeto permitirá o acesso aos controles de luz e eletrodomésticos de forma remota, permitindo um maior controle se existe algum dispositivo acionado de forma desnecessária, podendo, assim, gerar uma economia muito maior do que o consumo que o MCU proporciona.

## 2.6 SENSORES E ATUADORES

Quando se fala em automação residencial, refere-se a todo sistema capaz de realizar uma ação por si só, onde antes era necessária uma intervenção humana. Para que seja realmente possível uma ação controlada do sistema de forma automática, o mesmo deve conter em sua constituição dispositivos capazes de perceber um evento pontual e responder, de forma quase que imediata, na forma de impulsos elétricos que o controlador consiga captar a informação.

Existe inúmeros tipos de sensores, porém em automação residencial os mais utilizados são:

- Sensores de Temperatura: Geralmente é constituído por um termistor, que opera de acordo com a variação de temperatura, alterando a resistência elétrica em seu terminal.

Podendo ser também sensores infravermelhos, que detecta a energia radiante, de acordo com a cor específica captada (BOLZANI, 2004b).

- Sensores de Qualidade: São sensores denominados por este trabalho como dispositivos capazes de perceber as instâncias do ambiente, como por exemplo: sensores de umidade relativa do ar, que são normalmente sensores capacitivos e fornecem a medida instantânea de forma analógica; sensores de qualidade do ar que medem os níveis de compostos como dióxido de enxofre, nitrogênio ou carbono; sensores termovelocimétricos, que captam a velocidade de variação da temperatura em um determinado tempo, podendo assim, captar incêndios, por exemplo; sensores de líquido, capazes de informar a presença de líquidos em um recipiente (BOLZANI, 2004b).
- Sensores Magnéticos: Compostos por contatos envoltos em vidro, a fim de serem imunes a corrosão. Normalmente utilizados em portas e janelas, e instalados em seus batentes, onde quando se abre, ou fecha, há a variação dos contatos do sensor e a variação de seu campo magnético (BOLZANI, 2004b).
- Sensores de Intensidade Luminosa: Tendo como componente mais básico o LDR, funcionando similarmente como o termistor, ou seja, varia a resistência elétrica de seu terminal, porém, agora, de acordo com a variação de luminosidade (BOLZANI, 2004b).
- Sensores de Movimento: Normalmente são sensores infravermelhos, detectando fontes de calor como o corpo humano (BOLZANI, 2004b).

Uma vez detectado as informações dos sensores, os microcontroladores deverão agir de acordo com sua programação, enviando informações a outros dispositivos para compensar as variações do ambiente, esses dispositivos normalmente são capazes de gerar ações físicas, como movimentos, sendo denominados de atuadores. Os atuadores em uma residência normalmente estão conectados em portas e, ou janelas como motores ou em luzes, sendo os interruptores

## 2.7 MQTT

Inventado pela IBM no final dos anos 90 para atuar na comunicação dos sensores de petróleo com satélites, o MQTT (*Message Queue Telemetry Transport*), não é tão sofisticado quanto seus concorrentes, porém seu método simples e com funcionalidade suficiente para

abranjer grande parte dos projetos, ele pode ser considerado um dos maiores candidatos para implementação em sistemas embarcados (BARROS, 2015).

Seu funcionamento é baseado em publicações e assinaturas, ou seja, um servidor, denominado como *broker*, recebe todas as mensagens dos usuários e repassa as mensagens para os usuários destinos. O nome mais correto para os usuários de *brokers* é cliente, pois pode ser qualquer coisa capaz de enviar ou receber mensagens, podendo ser sensores ou um aplicativo. Então, o cliente pode assinar um tópico de mensagem no broker, como por exemplo, se um quarto for o tópico, suas publicações podem ser os acionamentos da lâmpada, tal conexão com o *broker* pode ser feita através de conexões com a internet usual, como a TCP/IP. Uma vez assinado o tópico, a cada nova publicação, ou variação dos elementos do tópico, uma mensagem é enviada ao cliente, que deverá agir de acordo com sua programação (BARROS, 2015).

Uma das palavras que define o protocolo MQTT, é a simplicidade, como já informado, não havendo muitas restrições de uso, sendo facilitada devido ao fato de existirem muitos brokers gratuitos, porém com restrições de utilização simultânea e quantidade de tópicos, porém para um protótipo de automação residencial este fato não é um empecilho, uma vez que já existem aplicativos gratuitos nas lojas de celulares de diversos sistemas operacionais capazes de serem um cliente, facilitando ainda mais a utilização desse sistema.

### 3 PESQUISA E DESENVOLVIMENTO

Este capítulo está dividido nas seguintes partes:

- Definição dos Componentes
- Estudo de Custo
- Tutorial de Desenvolvimentos

De forma resumida, será especificado o projeto, escolhendo-se quais e quantos dispositivos foram utilizados no projeto, para que, assim, se faça um estudo do custo do projeto, pois foi sempre visado o baixo custo do mesmo. Por fim, com todos os componentes definidos e obtidos será realizado o desenvolvimento do projeto e o detalhamento de cada passo realizado em forma de um tutorial para que seja possível a utilização futura de entusiastas, transformando o projeto em *Open Source*.

#### 3.1 DEFINIÇÃO DOS COMPONENTES

Sabendo que este trabalho, em específico, tem o intuito de ser de baixo custo e ser uma representação do funcionamento da integração da automação residencial utilizando internet das coisas, portanto será mais simples do que se fosse um projeto com a intenção de ser comercializado.

Levando em consideração estes fatores, estima-se que o rumo do projeto será fazer o acionamento de relés e LEDs de forma remota, pois o acionamento de relés seria para o acionamento de dispositivos ligados a rede de energia doméstica, de 110V ou 220V, ou seja, pode ser desde uma lâmpada até dispositivos eletrodomésticos, como ventilador ou cafeteira. Neste caso, o LED, representa outros dispositivos que não necessitam estar conectados a rede doméstica, e basta estar conectado na placa de controle para que seu funcionamento seja possível, podendo ser servomotor para fazer sistemas de alimentação de animais de estimação de forma automática, ou acionamento de travas de janelas e portas.

Conforme comentado no Capítulo 2, o dispositivo de controle utilizado neste trabalho será o NodeMCU. Mas é importante lembrar que uma vez definida e especificada a aplicação do projeto, deve-se verificar se o dispositivo estudado será capaz de suportar a conexão com os dispositivos necessários para a realização do projeto.

Sabe-se que relés são dispositivos com acionamento digital, portanto basta utilizar os GPIO contidos na placa. O uso restrito será com relação a sensores analógicos, uma vez que a placa contém somente uma porta analógica, portanto, caso seja necessário a utilização de mais

de um sensor ou saída analógica, deve-se ter um dispositivo externo, como multiplexador. Neste projeto, a placa de desenvolvimento NodeMCU conseguirá atender às especificações do projeto realizado.

A seguir, serão apresentados os componentes utilizados neste projeto:

- 1 placa de controle NodeMCU;
- Cabo Flexível para rede elétrica;
- Conectores de Tomada;
- Fios de conexão eletrônica de 0,5mm;
- Lâmpadas de 110V ou 220V (quantidade irá variar de acordo com o número de relés a ser utilizado);
- LEDS;
- Relés de 120V ou 220V, ou módulo de relé para Arduino com 2 ou 4 canais.

Os dispositivos listados podem sofrer pequenas alterações, como a entrada de sensores ou atuadores para uma melhor representação da idéia a ser implementada do projeto, porém estes são os dispositivos bases do trabalho.

### 3.2 ESTUDO DE CUSTO

Para a realização desta parte do capítulo será feita duas formas de pesquisa, um a respeito dos valores dos componentes no Brasil, pois é o país onde o projeto é realizado. Desta forma, será mais prático de encontrar estes produtos nos locais, porém a diferença de valor com relação a outra parte da pesquisa de preço, que será realizado levando em consideração aos preços em sites chineses, conhecidos por terem preços abaixo dos demais.

Os sites pesquisados no Brasil para a pesquisa de componentes eletrônicos foram principalmente o Filipeflop e a RoboCore. Para a pesquisa dos valores de importação da china será retirado os valores do site da AliExpress. Vale ressaltar que todos os valores foram pesquisados no dia 03 de Novembro de 2018.

Dispositivos de conexão com a rede, como os cabos e os conectores de tomada não foram pesquisados em sites chineses, principalmente pelo fato de que o padrão de tomada brasileiro é diferente do restante do mundo, o que mais se assemelha é o europeu, porém ainda é diferente ao brasileiro. O site utilizado para consulta foi o da Leroy Merlin.

Por fim, deve-se destacar que o autor escolheu a utilização de módulos de relé ao invés de comprar somente o relé, escolha que apesar de encarecer o projeto aumenta sua praticidade e a possibilidade de um *hobista* conseguir utilizar sem muitas dificuldades.

Tabela 1 – Tabela de Preços

Produto	No Brasil (em Reais)	Na China (em Reais)
Cabos de Energia	Leroy Merlin: R\$1,25 (por metro)	-
Cabos de <i>Jumper</i>	Filipeflop: R\$12,90 RoboCore: R\$8,00	R\$3,00
Conectores de Energia	Leroy Merlin: R\$4,19 (1 un.)	-
Lâmpadas de LED	Leroy Merlin: R\$5,99 (1 un.)	R\$3,00
LED	Filipeflop: R\$2,40 (10 un.) RoboCore: R\$2,30 (10 un.)	R\$1,50
Módulo de Relé com 1 canal	Filipeflop: R\$8,90 RoboCore: R\$12,90	R\$2,57
Módulo de Relé com 2 canais	Filipeflop: R\$12,90	R\$4,55
Módulo de Relé com 4 canais	Filipeflop: R\$26,90	R\$8,49
NodeMCU	Filipeflop: R\$44,90 RoboCore: R\$49,00	R\$10,00

Fonte: Aliexpress (2018); Robocore (2018); Merlin (2018)

Como é possível perceber, itens importados da China saem mais baratos, o que pode facilitar quem busca uma diminuição dos gastos de investimento em automação residencial. Também vale ressaltar que o autor escolheu em primeira instância utilizar um módulo de relé com 2 canais, pois já serviria para demonstrar o funcionamento da ideia do projeto, logo se faz necessário ter 2 Lâmpadas LED, uma para cada canal do módulo de relé. Estima-se que utilizar 5 metros de cabo de energia será o suficiente para o protótipo. Sendo assim, chega-se no valor total de investimento, tomando como base o menor valor para cada item e não listando o valor de frete para ambas opções de compra.

Tabela 2 – Preço Total

Produto	No Brasil (em Reais)	Na China (em Reais)
TOTAL	R\$ 90,52	R\$ 35,49

Fonte: Aliexpress (2018); Robocore (2018); Merlin (2018)

Como se pode perceber o custo do projeto não ultrapassará o valor de 100 reais, tornando-o algo acessível para entusiastas que desejem investir em sua própria automação residencial, somente tendo em vista que para a aplicação real do produto deve-se atentar a alguns detalhes a mais que o trabalho aqui realizado não se atenta, como, por exemplo, custo

de instalação e custos adicionais na rede, sendo fatores de atenção porém não fatores que inviabilizarão o custo do projeto.

### 3.3 TUTORIAL DE DESENVOLVIMENTO

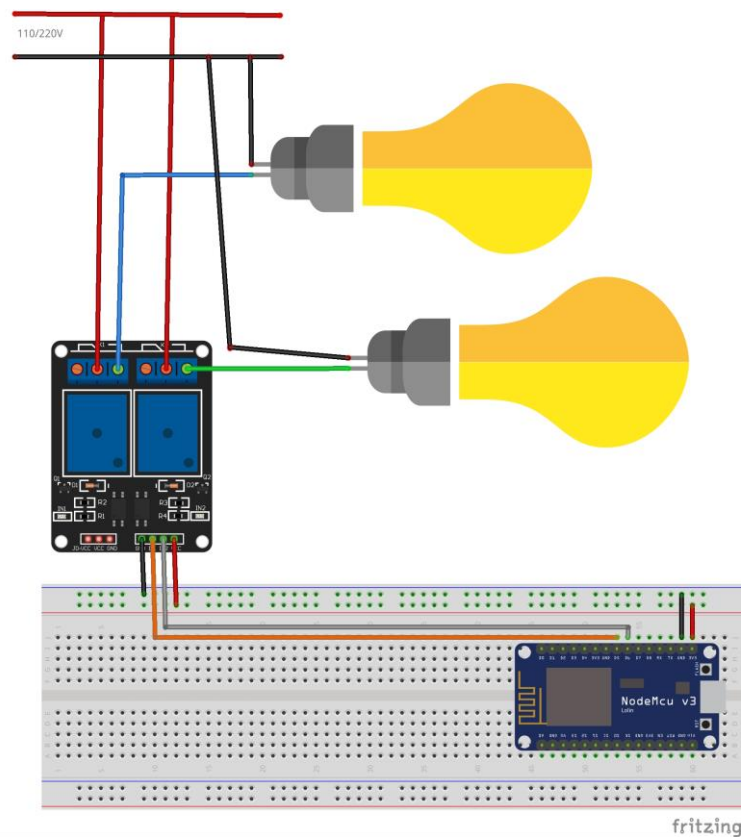
Uma vez especificado o projeto e definido os produtos que serão comprados, inicia-se a implementação do projeto e o desenvolvimento do mesmo.

Em uma primeira instância foi utilizada a plataforma aberta do Fritzing, por ser uma plataforma gratuita e de fácil manuseio. Esta plataforma está disponível na página [fritzing.org](http://fritzing.org) e será utilizada para montar o esquemático a ser desenvolvido pelo projeto, tornando-o muito mais fácil de ser compreendido.

Esta plataforma é disponível tanto para *Windows*, quanto para *Linux*, e se alguns componentes não estiverem em seu banco de dados, como o próprio NodeMCU, é possível baixar o modelo e adicioná-lo à biblioteca de pesquisa.

Este software somente será utilizado para a demonstração do modelo empregado neste trabalho, para facilitar a visualização do protótipo montado, por isso, não será detalhado o passo a passo realizado, somente mostrado o resultado final obtido na Figura 8.

Figura 8 – Esquemático do Projeto

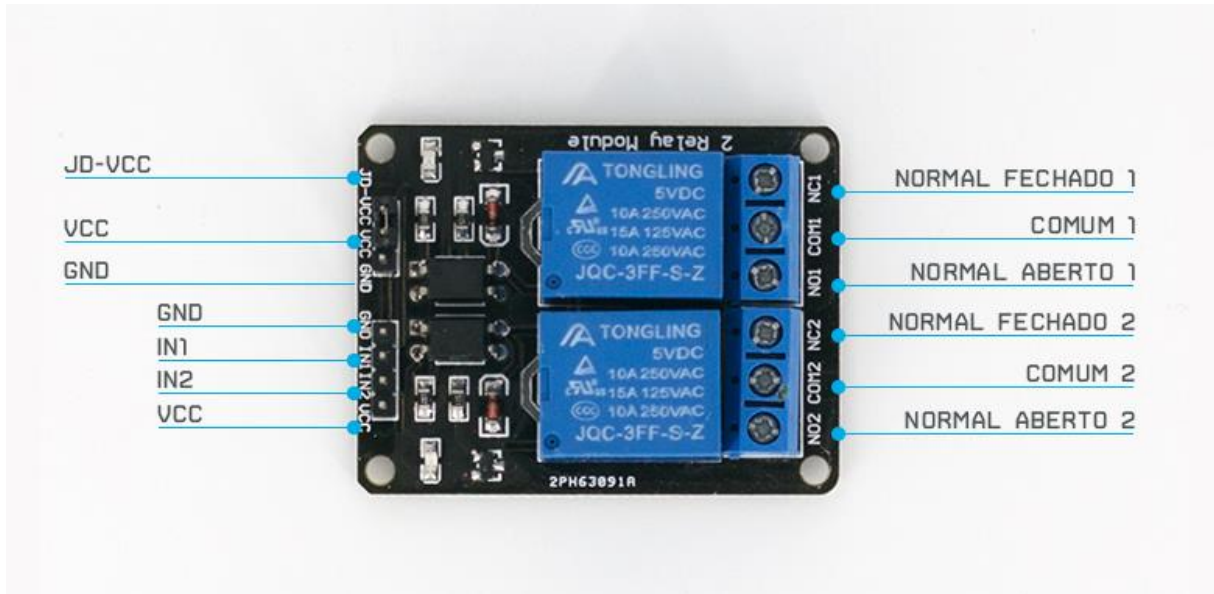


Fonte: Próprio Autor.

No esquemático montado os dois relés funcionam como chaves eletrônicas, fechando o contato que é normalmente aberto, e acende as duas lâmpadas quando acionados. Porém como é a primeira vez que o presente projeto apresenta o módulo de relé, vale mostrar sua configuração, pois é um componente pelo qual permite uma fonte externa de 5V para ser colocada no pino JD-VCC, os pinos de GND e VCC serão conectados juntos aos pinos de mesma nomenclatura da placa de desenvolvimento NodeMCU, e os pinos do módulo de relé IN1 e IN2 serão conectados aos GPIO desejados da NodeMCU, ou seja, serão conectados aos pinos de controle definidos por software, na figura mostrada foi colocado nos pinos 5 e 6 da placa de desenvolvimento por uma simples conveniência. Também deve-se atentar ao fato de que cada relé suporta no máximo 10A, ou seja, equipamentos que forem conectados a ele não podem ultrapassar esta medida. Os relés têm duas opções de conexão: normalmente aberto (NO) e normalmente fechado (NC), foi escolhido conectar pelo pino normalmente aberto (NO) do módulo. O conector de tomada deve ser colocado no fio vermelho de 110/220V e preto logo ao seu lado da Figura 10. Por fim, vale ressaltar muito cuidado no manuseio do

módulo, uma vez que qualquer manutenção ou manuseio deve ser feito antes de se conectar a rede elétrica, tanto residencial como da placa.

Figura 9 – Módulo Relé de 2 Canais



Fonte: Filipeflop (2018)

No esquemático montado os dois relés funcionam como chaves eletrônicas, fechando o contato que é normalmente aberto, acendendo as lâmpadas e servindo como interruptores digitais com controle remoto, concluindo assim o hardware necessário para implementar o trabalho e podendo começar a programação.

### 3.3.1 Arduino IDE

A primeira coisa que se deve fazer para iniciar a parte da programação do trabalho é instalar o software de desenvolvimento do Arduino, para realizar esta ação é muito simples, uma vez que o próprio fabricante disponibiliza em seu site versões para Windows, Mac OS X e Linux (tanto 32 bit quanto 64 bits), bastando acessar o link <<https://www.arduino.cc/en/Main/Software>> (Acessado em 2018) e escolher a versão que melhor se adapte com o computador do usuário.

Após realizar a instalação do Arduino IDE, se faz necessário instalar a biblioteca da ESP8266, uma vez que o programa de desenvolvimento do Arduino não contém esta biblioteca em seu modelo padrão. Para realizar tal feito deve seguir os seguintes passos:

1. Clicar em Arquivo;
2. Clicar em Preferências;

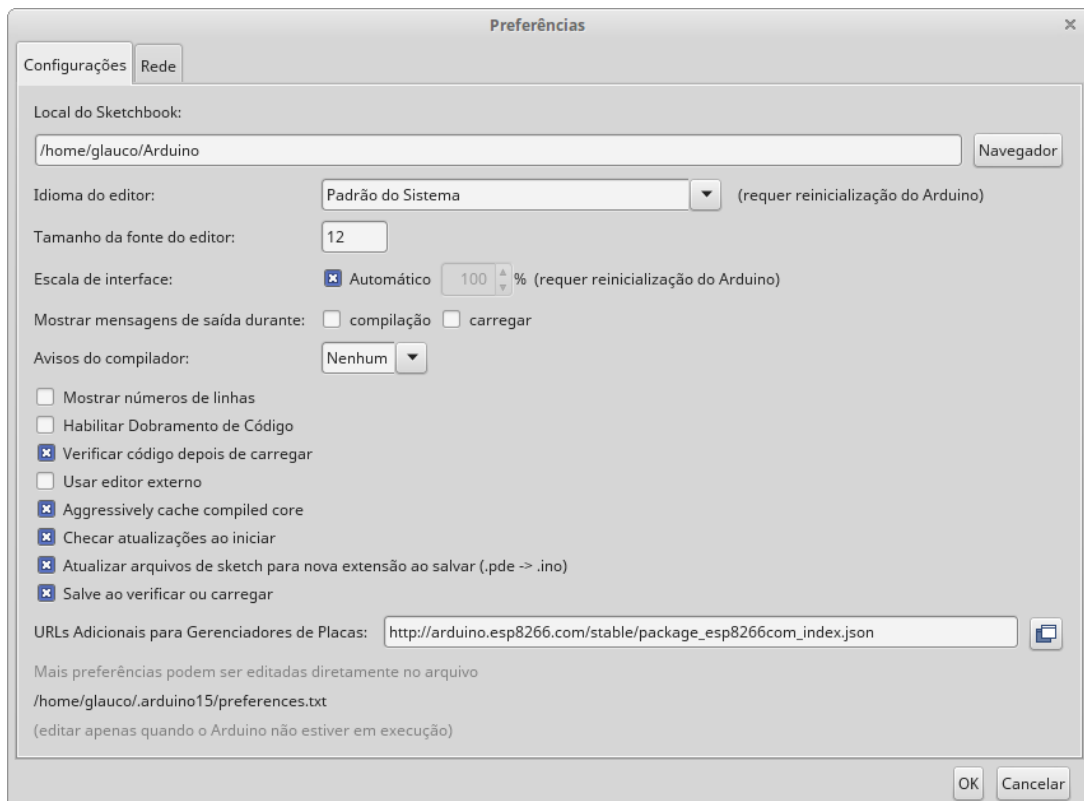
3. No campo (que deverá estar vazio) ao lado da frase “URLs Adicionais para Gerenciadores de Placas:” copiar o seguinte link `<http://arduino.esp8266.com/stable/package_esp8266com_index.json>`;
4. Clicar em OK;
5. Clicar em Ferramentas;
6. Clicar em Placas → Gerenciador de Placas;
7. No campo “Refine sua busca...” digite “esp8266”;
8. Selecione a versão desejada da placa, recomenda-se a mais atual, e clicar em Instalar.

Também é possível no lugar de realizar a primeira e a segunda ação, somente apertando simultaneamente o botão Ctrl e Vírgula.

Realizado este processo, basta selecionar a placa da NodeMCU clicando em Ferramentas → Placas e escolhendo a versão NodeMCU 1.0 (ESP-12E Module), que é a utilizada por este projeto. Importante notar que as configurações da placa devem ser especificadas, para diminuir os erros, em:

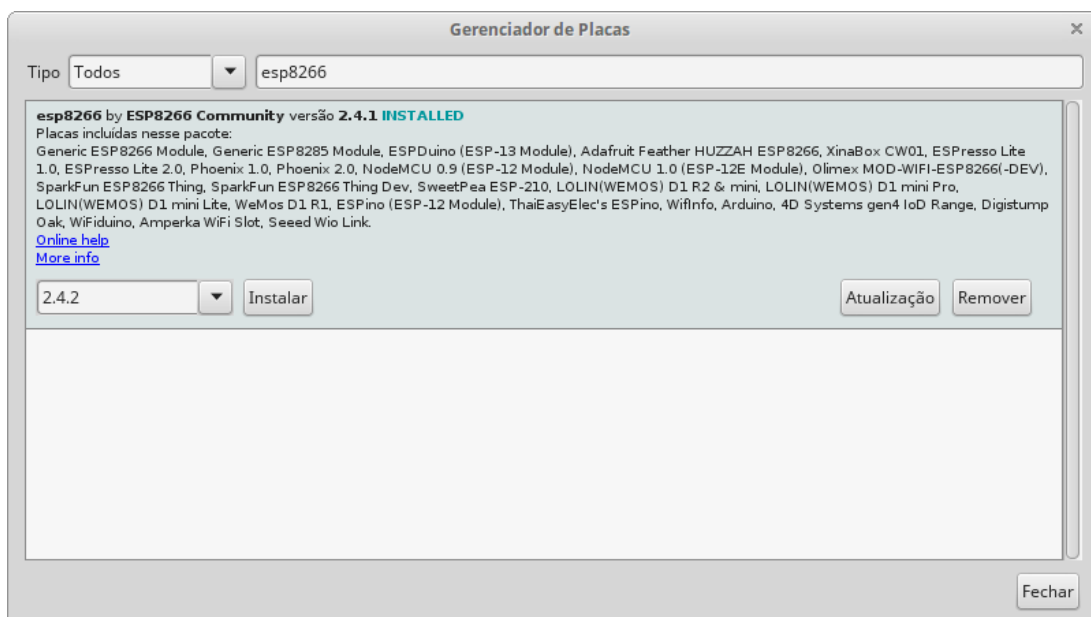
- Ferramentas → CPU *Frequency* : “” → 80MHz;
- Ferramentas → Upload Speed: “” → 115200;
- Ferramentas → Port → (Escolher a porta de USB que está conectado a NodeMCU, varia de acordo com o computador).

Figura 10 – Instalando a biblioteca ESP8266 parte 1



Fonte: Próprio Autor

Figura 11 – Instalando a biblioteca ESP8266 parte 2



Fonte: Próprio Autor

### 3.3.2 Adafruit IO

Antes de começar a programação, deve-se ter noção de como será feita a conexão com a internet, pois cada tipo de conexão requer uma configuração no código fonte diferente. Como já mencionado este projeto utilizará o protocolo MQTT, e buscou-se uma fonte desta tecnologia que fosse gratuita e estável encontrando a plataforma de Internet das Coisas da Adafruit, o Adafruit IO.

Que resumidamente é uma plataforma onde cada usuário teria um espaço interno ao servidor da Adafruit, onde poderá controlar e utilizar para os fins que preferir. A plataforma é gratuita porém limitada, pois ela disponibiliza formas de pagamentos onde desbloqueiam todas funcionalidades do sistema, porém para o desejado projeto, que seria um sistema com poucas ações, ela funciona de forma perfeita, portanto quanto mais complexo for o aprimoramento deste projeto por entusiastas, saiba que ou terá que pagar a plataforma de MQTT ou encontrar outro meio de conexão.

O Adafruit IO trabalha como o intermediador da NodeMCU e a plataforma que o usuário utilizará para controlar a placa, podendo também ser a própria plataforma de controle, pois pra ela é capaz de operar os dispositivos conectados a NodeMCU, tanto para acioná-los, quanto para realizar a telemetria dos mesmos. Para seu funcionamento se faz necessário criar *Feeds* que operam como um painel capaz de receber e guardar informações, para que seja incluso esses *Feeds* em um *Dashboard*, que será a própria plataforma de controle.

Mas com fins de tornar mais didático este projeto, será realizado o passo a passo desta parte, bastando seguir os passos a seguir:

1. Entre no site da Adafruit IO <<https://io.adafruit.com/>> (Acessado em 08 de Novembro de 2018);
2. Crie sua conta clicando na aba que está presente no canto superior direito da página: “*Get Started for Free*”;
3. Após criada a sua conta, você será direcionado para a página inicial/principal da plataforma, onde deverá ir no menu *Feeds* que fica do lado esquerdo da página, escrito em azul;
4. Uma vez na aba *Feeds*, clique: *Actions* → *Create a New Feed*;
5. Nomeie o seu *Feed* de uma forma que seja possível distinguir cada um, uma vez que este será o nome que se utilizará na hora de programar, pois será como nomeará os relés no *software*. Eu nomeei o meu de acordo com o cômodo da residência que eu desejava controlar, como: Quarto, Sala, Cozinha, Banheiro;

6. Uma vez criado os *Feeds*, deve-se acessar a parte de *Dashboards* que está presente no lado esquerdo da página, escrita em azul e logo abaixo de *Feeds*, que possivelmente é onde o usuário deve se encontrar caso esteja seguindo o passo-a-passo;
7. Na aba de *Dashboards*, clique em *Actions* → *Create a New Dashboard*;
8. Nomeie o seu *Dashboard* conforme desejar, não terá impacto direto no seu projeto o nome dado;
9. Abra o novo *Dashboard*, e o usuário será direcionado a uma página vazia;
10. Pressione o botão azul com um símbolo de soma (+), no canto superior direito da página;
11. Por hora, deseja-se apenas acionar os relés, logo crie um botão *Toggle*, que deverá ser a primeira opção que aparecerá;
12. Depois basta escolher qual *Feed*, que este botão representará.

Figura 12 – *Feeds* Adafruit IO parte 1

The screenshot displays the Adafruit IO interface for a user named 'Jabbinhathethutt'. The main content area is titled 'Feeds' and contains a table with the following data:

Key	Last value	Recorded
default		
banheiro	OFF	4 months ago
cozinha	ON	4 months ago
ligardesliggr	OFF	4 months ago
ligardesliggr	OFF	4 months ago
quarto	OFF	4 months ago
sala	ON	4 months ago

The left sidebar includes navigation options: Home, Feeds, Dashboards, Triggers, Services, View AIO Key, API Docs, FAQ, Learn, News, Support, Terms of Service, and Send Feedback. The top right corner shows the user's name 'Hello, Glaucio Fernandes' and links for 'Sign Out' and 'My Account'. The bottom left corner displays 'Free Usage' statistics: Feeds: 6 of 10, Dashboards: 1 of 5, Rate: 30 / minute, Current Usage: 0 / min, Storage: 30 days, and a 'Change Plan' link.

Fonte: Próprio Autor

Figura 13 – Feeds Adafruit IO parte 2

The screenshot displays the 'Feeds' page for a user named 'Jabbinhathehutt'. The specific feed is 'Liga/Desliga luz do banheiro'. The main content area features a graph with the message 'There is no data to display'. To the right of the graph is a settings panel with the following options:

- Feed Info:** Manage feed name, key, description, and tags.
- Privacy:** This feed is **private**. Only you can see it.
- Sharing:** Not shared yet.
- Feed History:** Feed history is **ON**. No data is recorded.
- Notifications:** You have no notifications active for this feed.
- Disable Feed:** Disabling a feed will remove it from your feed count and prevent you from adding new data to it.
- License:** No Default License.

The left sidebar includes navigation links: Home, Feeds, Dashboards, Triggers, Services, View AIO Key, API Docs, FAQ, Learn, News, Support, Terms of Service, and Send Feedback. At the bottom left, there is a 'Free Usage' section with details: Feeds: 6 of 10, Dashboards: 1 of 5, Rate: 30 / minute, Current Usage: 0 / min, Storage: 30 days, and a 'Change Plan' link.

Fonte: Próprio Autor

Figura 14 – Dashboards Adafruit IO parte 1

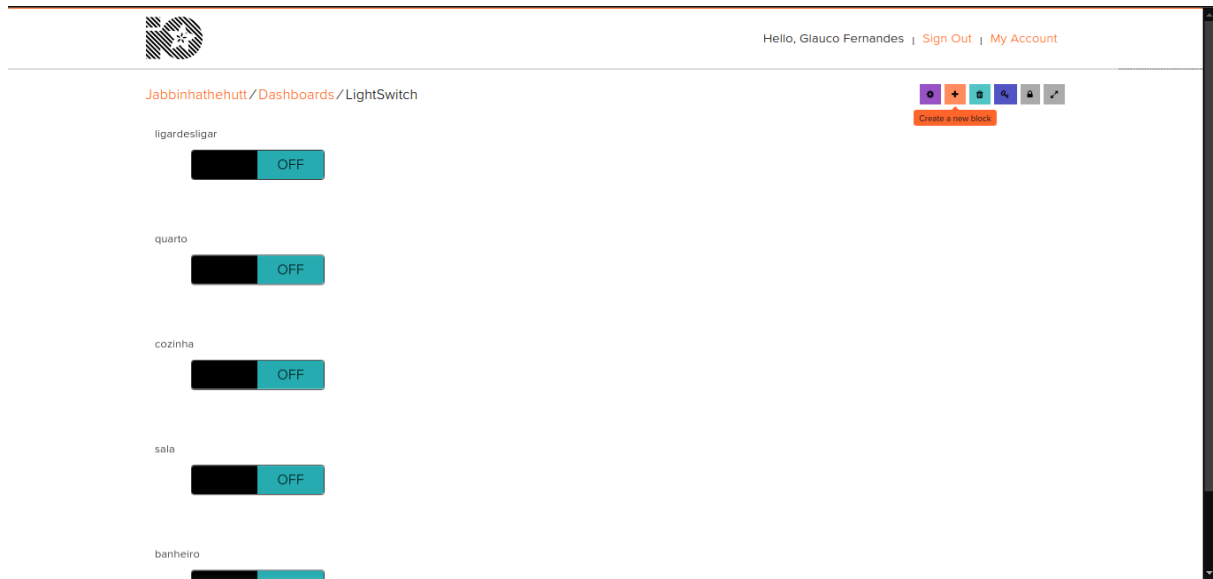
The screenshot displays the 'Dashboards' page for a user named 'Jabbinhathehutt'. The dashboard is titled 'lightswitch' and has a key 'lightswitch'. A dropdown menu is open over the 'Actions' button, showing the following options:

- Create a New Dashboard
- Edit Selected Dashboard
- Remove Selected Dashboards

The top right of the page shows the user 'Hello, Glauco Fernandes' with links for 'Sign Out' and 'My Account'. The left sidebar is identical to the previous screenshot, including navigation links and the 'Free Usage' section.

Fonte: Próprio Autor

Figura 15 – *Dashboards* Adafruit IO parte 2



Fonte: Próprio Autor

A plataforma do Adafruit IO está pronta para ser operada, percebendo que na aba de *Feeds* é possível ter um controle de quando o dispositivo foi acionado, pois ele contém um histórico, e na aba *Dashboards* foram criados botões que controlam os *Feeds*, e conseqüentemente, controlará os relés, já sendo, então, a primeira forma de controle que este trabalho fornece ao usuário, o controle via uma página da Internet.

Porém, novamente a biblioteca não está contida na base do Arduino IO, portanto deverá ser realizada a instalação da mesma, bastando seguir os passos:

1. Acesse o site <[https://github.com/adafruit/Adafruit\\_IO\\_Arduino](https://github.com/adafruit/Adafruit_IO_Arduino)> (Acessado em 08 de Novembro de 2018);
2. Clique em “Clone or Download” → *Download ZIP*;
3. Abra o Arduino IO;
4. Clique em *Sketch* → Incluir Biblioteca → Adicionar biblioteca .ZIP;
5. Procure a pasta *Adafruit\_IO\_Arduino-master* e instale-a.

Realizado este processo o Arduino IO já contém todas as bibliotecas e placas necessárias para o funcionamento do projeto. Vale ressaltar que caso o usuário queira utilizar outra plataforma de MQTT, esta biblioteca da Adafruit IO funcionará, pois o método de conexão entre plataformas que utilizam MQTT é a mesma, portanto o usuário poderá seguir estes passo a passo somente alterando os detalhes para a plataforma desejada.

### 3.3.3 IFTTT

Até o presente momento já foi apresentado o *software* de desenvolvimento (Arduino IDE), o meio de comunicação entre a placa de desenvolvimento e a internet (Adafruit IO), que também serve como um meio de controle da placa, porém ainda não foi apresentado como o trabalho pretende ser controlado, podendo ser considerado esse meio a principal forma de controle. Este controle se dá devido a plataforma IFTTT, que se denomina como uma plataforma que conecta aplicativos, com o intuito de melhorar a conexão dos mesmos uma vez que nem tudo que se encontra na internet tem uma conexão estável e correta.

O motivo real da plataforma ter sido escolhida é a possibilidade de conexão com o *Google Assistant*, ou seja, qualquer dispositivo com conexão com conta Google consegue utilizar o que for realizado pela plataforma, além de que o manuseio dentro do site é baseado em lógica Se/Então, ou seja, caso ocorra uma ação designada pelo IFTTT, ele resultará alguma consequência no Adafruit IO.

Para o manuseio na plataforma IFTTT basta seguir as ações a seguir:

1. Acesse o site <<https://ifttt.com/>> (Acessado em 09 de Novembro de 2018);
2. Crie sua conta clicando em “*Sign in*”, no canto superior direito da página (Recomenda-se utilizar a conta do Google pelo qual o dispositivo de controle utiliza, ou criar uma conta Google somente para o controle da residência);
3. Após a criação da conta clique em “*My Applets*”, presente no canto superior esquerda da tela;
4. Clique no botão de “*New Applet*”;
5. Clique no campo que está escrito “+this” em azul claro;
6. No campo de pesquisa que aparecerá digite “Google Assistant” e clique no ícone que aparecerá em baixo com o mesmo nome;
7. Clique no quadro com o nome de “*Say a simple phrase*”;
8. Nos campos de “*What do you want to say?*”, “*What’s another way to say it?*” e “*And another way?*” deverá ser colocado frases em inglês para o controle, como por exemplo caso queira ligar as luzes da sala o autor escolheu por colocar “*Turn room ON*”, “*Room on*” e “*Turn on the room*”;
9. No campo “*What do you want the Assistant to say in response?*” deve ser colocado o que o Google responderá ao seu comando, ou seja, uma frase dita para mostrar que o comando realizado foi aceito e realizado. Seguindo o exemplo do passo 8, escolheu a frase “*Room is on*”, para indicar que as luzes da sala estão acesas;

10. É possível escolher a linguagem que deverá estar os comandos, porém, como já mencionado, não tem Português e, então, escolheu-se o inglês por ser uma linguagem mais universal;
11. Clique no botão “+that”;
12. No campo de pesquisa digite “Adafruit” e clique no quadro que aparecerá em baixo com a mesma nomenclatura;
13. Clique no quadro “*Send data to Adafruit IO*”;
14. Agora será o elo de ligação entre o IFTT e o Adafruit IO, ou seja, a cada ação realizada dentro do Google Assistant terá efeito direto no *Feed* do MQTT e conseqüentemente em seu *Dashboard*, ou seja, a cada ação de voz realizada mudará o estado do interruptor criado na plataforma Adafruit IO. Para isto, basta escolher o nome do *Feed* desejado e no campo “*Data to save:*” colocar ON para quando quiser ligar e OFF para quando quiser desligar;
15. Agora basta refazer o procedimento para, como o exemplo dado foi de ligar, desligar o mesmo equipamento.

O maior empecilho encontrado por esta plataforma é que como foi escolhido o controle de voz para controle dos dispositivos, todas as ações devem ser feitas em inglês, gerando uma barreira entre a língua nativa brasileira e a língua de comando. Para contornar este problema, é possível encontrar dentro da plataforma Google Play, diversos aplicativos gratuitos de conexão com *brokers* MQTT, como por exemplo o aplicativo IoT MQTT Dashboard, desenvolvido para Android pela Nghia TH, que será necessário inserir a chave de acesso a conta do usuário, sendo possível obter clicando no botão amarelado, com uma figura de chave no centro, presente na Figura15 deste projeto.

### 3.3.4 Código

O código realizado foi dividido por partes para ser representado no projeto e conterà um anexo com o mesmo completo. As partes serão apresentadas a seguir:

1. Incluindo as Bibliotecas

Primeiro deve-se incluir 3 bibliotecas no código

```
//Controla as conexões com o MQTT
#include <Adafruit_MQTT.h>
#include <Adafruit_MQTT_Client.h>
//Controla as conexões de WiFi da ESP32
#include <ESP8266WiFi.h>
```

## 2. Definindo as Entradas e Saídas

A fim de tornar o programa mais prático e *Open Source*, são dados nomes as portas IO para que ao longo do processo não as tratemos como números e sim como nomes, portanto mais fácil o entendimento e torna mais difícil o erro.

```
//Definindo entradas e saídas
#define ledsala 12
#define ledcozinha 13
#define ledquarto 14
#define ledbanheiro 15
#define ledhall 2
#define ldr 0
int ldrValor = 0;
```

## 3. Conectando a Rede

Deve-se conectar a placa a sua rede local, bastando substituir “nome” pelo nome da sua rede, também entre aspas. O mesmo deve ser feito para a “senha”.

```
//Conectando a Rede de Internet
#define WIFI_SSID "nome"
#define WIFI_PASS "senha"
```

## 4. Conectando ao Broker MQTT

Após definir qual será sua rede WiFi, deve-se definir qual será sua rede MQTT, esse procedimento feito para a Adafruit IO pode ser feito também para qualquer outro *Broker* MQTT, portanto é um código que não se restringe somente a essa utilização, como já foi mencionado. Antes de definir deve-se acessar a chave amarela contida na Figura 15 para substituir no campo MQTT\_PASS.

```
#define MQTT_SERV "io.adafruit.com"
#define MQTT_PORT 1883
#define MQTT_NAME "usuário"
#define MQTT_PASS "chave"
```

## 5. Criando um cliente para o WiFi e para o Adafruit como variáveis globais

Posteriormente deve-se criar variáveis globais para cada feed contido em seu Dashboard do Adafruit IO, pode conter o mesmo nome assim facilita na hora de programar.

```
WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, MQTT_SERV, MQTT_PORT,
MQTT_NAME, MQTT_PASS);

Adafruit_MQTT_Subscribe sala = Adafruit_MQTT_Subscribe(&mqtt,
```

```

MQTT_NAME "/f/sala"); //sala = nome do feed
    Adafruit_MQTT_Subscribe cozinha = Adafruit_MQTT_Subscribe(&mqtt,
MQTT_NAME "/f/cozinha"); //cozinha = nome do feed
    Adafruit_MQTT_Subscribe quarto = Adafruit_MQTT_Subscribe(&mqtt,
MQTT_NAME "/f/quarto"); //quarto = nome do feed
    Adafruit_MQTT_Subscribe banheiro = Adafruit_MQTT_Subscribe(&mqtt,
MQTT_NAME "/f/banheiro"); //banheiro = nome do feed
    Adafruit_MQTT_Subscribe ligardesligar = Adafruit_MQTT_Subscribe(&mqtt,
MQTT_NAME "/f/ligardesligar"); //ligardesligar = nome do feed
    Adafruit_MQTT_Subscribe hall = Adafruit_MQTT_Subscribe(&mqtt,
MQTT_NAME "/f/hall"); //hall= nome do feed

```

## 6. Conectando ao WiFi

Finalizado o processo de definições do programa deve-se indicar a velocidade da comunicação serial, optado por ser 115200 nesse projeto, pois é a velocidade necessária para gravar na placa e para operar. Deve-se também conectar a placa ao wifi, agora de forma real.

```

Serial.begin(115200);

//Connect to WiFi
Serial.print("\n\nConectando ao WiFi... ");
WiFi.begin(WIFI_SSID, WIFI_PASS);
while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
}

Serial.println("Conectado");

```

## 7. Inscrevendo nos Feeds

Uma vez conectado na rede WiFi faz-se possível a comunicação da placa com o Adafruit IO, que será o próximo passo, conectar a NodeMCU com os Feeds do Adafruit IO

```

//Inscrever-se nos feeds
mqtt.subscribe(&sala);
mqtt.subscribe(&cozinha);
mqtt.subscribe(&quarto);
mqtt.subscribe(&banheiro);
mqtt.subscribe(&ligardesligar);

```

## 8. Definir as IO como entrada ou saída

Como os pinos contidos na NodeMCU pode ser definidos como entrada ou saída, deve-se denominar como cada pino já definido operará.

```
pinMode(LED_BUILTIN, OUTPUT);
pinMode(ledsala, OUTPUT);
pinMode(ledcozinha, OUTPUT);
pinMode(ledquarto, OUTPUT);
pinMode(ledbanheiro, OUTPUT);
pinMode(ledhall, OUTPUT);
pinMode(ldr, INPUT);
```

## 9. Conectar ou Reconectar ao MQTT

Nessa parte já foi configurado todo o programa e iniciará de forma real sua operação. E como todo o processo se baseia na conexão com o MQTT, deve-se iniciar por ele.

```
//Conecta e/ou Reconecta ao MQTT
MQTT_connect();

//Ler se o usuário enviou algum dado, ou
//esperar 5 segundos para uma nova informação
Adafruit_MQTT_Subscribe * subscription;
```

## 10. Controlando o LDR

Como o LDR não depende do MQTT, ele se faz de forma independente já pode colocar ele na lógica.

```
//Funcionamento do LDR
ldrValor = analogRead(ldrpin); //O valor lido será entre 0 e 1023
//se o valor lido for maior que 200, liga o led
if (ldrValor >= 200) digitalWrite(ledldr, HIGH);
// senão, apaga o led
else digitalWrite(ledldr, LOW);
```

## 11. Controlando os Feeds

Agora deve-se fazer de fato o controle de cada *Feed*, que funciona em um *loop*, onde se lê cada *Feed* e somente realizará a ação do último *Feed* acessado, impedindo que uma ação sobreponha a outra.

```
while ((subscription = mqtt.readSubscription(5000)))
{
  //Se tudo estiver OK...

  if (subscription == &sala)
  {
    //Salva o valor Serial na tela
    Serial.print("Sala: ");
    Serial.println((char*) sala.lastread);
```

```

//Se for "ON" liga a lâmpada
//Se não, desligue-a
if (!strcmp((char*) sala.lastread, "ON"))
{
    digitalWrite(ledsala, HIGH);
}
else
{
    digitalWrite(ledsala, LOW);
}
}

if (subscription == &cozinha)
{
    //Salva o valor Serial na tela
    Serial.print("Cozinha: ");
    Serial.println((char*) cozinha.lastread);

    //Se for "ON" liga a lâmpada
    //Se não, desligue-a
    if (!strcmp((char*) cozinha.lastread, "ON"))
    {
        digitalWrite(ledcozinha, HIGH);
    }
    else
    {
        digitalWrite(ledcozinha, LOW);
    }
}

if (subscription == &quarto)
{
    //Salva o valor Serial na tela
    Serial.print("Quarto: ");
    Serial.println((char*) quarto.lastread);

    //Se for "ON" liga a lâmpada
    //Se não, desligue-a
    if (!strcmp((char*) quarto.lastread, "ON"))
    {
        digitalWrite(ledquarto, HIGH);
    }
    else
    {
        digitalWrite(ledquarto, LOW);
    }
}

if (subscription == &banheiro)
{

```

```

//Salva o valor Serial na tela
Serial.print("Banheiro: ");
Serial.println((char*) banheiro.lastread);

//Se for "ON" liga a lâmpada
//Se não, desligue-a
if (!strcmp((char*) banheiro.lastread, "ON"))
{
    digitalWrite(ledbanheiro, HIGH);
}
else
{
    digitalWrite(ledbanheiro, LOW);
}
}

if (subscription == &ligardesligar)
{
    //Salva o valor Serial na tela
    Serial.print("Ligar/Desligar: ");
    Serial.println((char*) ligardesligar.lastread);

    //Se for "ON" liga a lâmpada
    //Se não, desligue-a
    if (!strcmp((char*) ligardesligar.lastread, "ON"))
    {
        digitalWrite(ledsala, HIGH);
        digitalWrite(ledcozinha, HIGH);
        digitalWrite(ledquarto, HIGH);
        digitalWrite(ledbanheiro, HIGH);
    }
    else
    {
        digitalWrite(ledsala, LOW);
        digitalWrite(ledcozinha, LOW);
        digitalWrite(ledquarto, LOW);
        digitalWrite(ledbanheiro, LOW);
    }
}
}
}

```

## 12. Conexão com o MQTT

Essa é a última parte do programa, onde já foi realizada toda a operação lógica, porém ela já foi citada na Etapa 9, porém lá somente como a ação e nessa etapa configuramos a ação já realizada.

```
void MQTT_connect()
{
  int8_t ret;
  // Stop if already connected
  if (mqtt.connected())
  {
    return;
  }

  Serial.print("Connectando ao MQTT... ");
  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) // conexão vai retornar 0 se estiver conectado
  {
    Serial.println(mqtt.connectErrorString(ret));
    Serial.println("Tentando reconectar ao MQTT em 5 segundos...");
    mqtt.disconnect();
    delay(5000);
    retries--;
    if (retries == 0)
    {
      //Só espera se estiver conectado
      while (1);
    }
  }
  Serial.println("Conectado ao MQTT!");
}
```

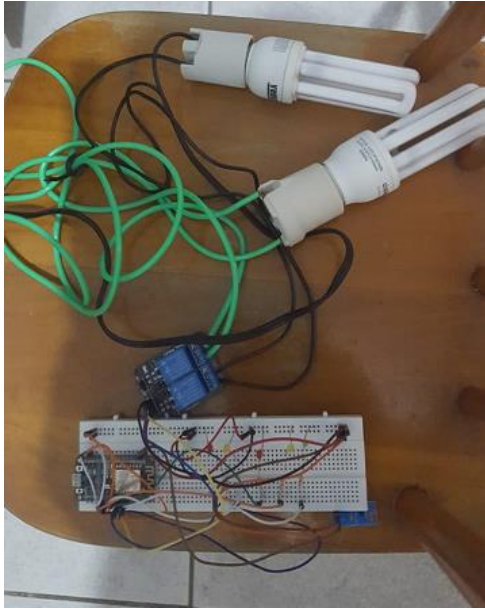
## 4 RESULTADOS

Neste capítulo estará descrito os resultados obtidos do projeto, abrangendo os erros e os acertos, assim como as melhorias, onde já é perceptível no código já demonstrado no capítulo 3, uma vez que está presente 4 cômodos da residência e a presença de um sensor sensível a luminosidade, o LDR, o que não é compatível com a ideia inicial do projeto, porém a adição dos cômodos no projeto foi para tornar ainda mais real e aplicável o protótipo desenvolvido por este trabalho, e a adição do LDR simula a luz externa de uma residência em função da luminosidade do ambiente, ou seja, a ideia é que ao anoitecer manter a área iluminada, podendo melhorar a segurança ou a economia na residência.

Porém, este capítulo será dividido em duas partes, uma primeira parte que demonstrará os problemas enfrentados, e/ou os erros encontrados, e a segunda parte que demonstrará os acertos do projeto e o funcionamento do mesmo. Em ambas as partes será proposto melhorias para o projeto, porém ainda não implementadas.

Antes de separar o capítulo nas partes descritas, faz-se válido a demonstração do protótipo com a foto retirada do mesmo, demonstrando a simples montagem e a melhoria realizada com relação ao já demonstrado pelo esquemático montado no *software* Fritzing, pois foram adicionados dois leds representando as luzes dos outros dois cômodos, onde não foram inclusos novos relés por falta de recursos do autor em adquirir um novo, e ao mesmo tempo por se tratar somente de uma demonstração. O LDR incluído também está perceptível, onde ele está conectado a um outro led representando, também, uma lâmpada. Mas vale ressaltar que a representação de um led no lugar de uma lâmpada se faz válido uma vez que tanto o led como o relé são acionados por sinais digitais, ou seja, o mesmo comando feito para controlar o led é feito para comandar e acionar o relé, portanto a representação é a mesma, e o led foi escolhido por emitir luz que é a mesma função de uma lâmpada residencial.

Figura 16 – Protótipo Real



Fonte: Próprio Autor.

#### 4.1 PONTOS A SEREM MELHORADOS

Como já mencionado foram detectados alguns pontos de melhoria durante a execução do projeto, nenhum que realmente tire a funcionalidade do protótipo, porém que para que seja implementado de fato precisam ser sanados, são eles:

1. O projeto foi executado tanto utilizando a plataforma Windows, quanto utilizando a plataforma Linux, porém houve divergência de funcionamento entre as duas plataformas com relação à instalação das bibliotecas. Ao se instalar a biblioteca do Adafruit IO como já mencionado por este trabalho, no Windows não se faz necessário realizar mais nenhuma ação, porém quando foi realizado o mesmo processo no Linux, fez-se necessário a instalação da biblioteca do MQTT de forma separada.

Para resolver este problema basta realizar os seguintes passos: Sketch → Incluir Biblioteca → Gerenciar Bibliotecas...

Na aba de pesquisa digite “Adafruit MQTT Library” e instalar a versão mais recente da biblioteca;

2. Outro problema percebido foi com relação ao incremento do LDR no projeto, uma vez que o código foi feito para renovar a assinatura com o MQTT de forma constante, o LDR só responde conforme for feito os comandos das lâmpadas, pois é quando o programa considera que é necessário realizar alguma ação.

Na forma que o código está descrito no capítulo anterior o problema foi parcialmente resolvido, pois o LDR responde antes da assinatura com o MQTT, porém o código não permite que o LDR e o MQTT façam ao mesmo tempo as suas devidas ações, é necessário realizar uma por vez;

3. Conforme já descrito no Capítulo 3, a área de *Dashboard* dentro da página do Adafruit IO é um meio de controle das lâmpadas, porém foi perceptível que, se ativado as lâmpadas pelo Google Assistant e as mesmas foram apagadas pelo site do Adafruit IO, ocorriam algumas vezes perda de conexão da placa de controle com o MQTT, fazendo necessário o reset manual da placa. Este problema ainda não foi estudado o motivo de estar ocorrendo, portanto ainda não existe uma solução até o devido momento;
4. Outro problema enfrentado foi o fato de que o IFTTT somente aceita comandos em inglês obrigatoriamente o celular tem que ter sua linguagem padrão em inglês. Nos testes foi realizado esta troca, porém mesmo assim o celular demorou a começar a reconhecer as falas de forma correta. Para facilitar os comandos, instalou-se o aplicativo Allo, que é um aplicativo da própria Google, que serve como uma plataforma de comunicação, onde tem a opção de contato o Google Assistant, ficando mais fácil para realizar os comandos do projeto.

Os erros como é perceptível não impedem o funcionamento do protótipo, com exceção de quando se faz necessário resetar a placa, porém ainda são erros e que precisam ser arrumados caso se queira implementar o projeto em algum domicílio.

## 4.2 ACERTOS

Como evidenciado, o projeto funcionou de maneira correta, os comandos de forma separada funcionaram todos de forma correta, independente do meio de controle. De forma mais detalhada, pode-se fazer o destaque com relação ao:

1. Um dos maiores acertos do projeto, que faz com que uma das propostas seja atingida, é o fato de que o projeto funcione com duas conexões de internet distintas, ou seja, a placa precisa estar conectada com somente uma rede de internet fixa e estável, e o usuário somente precisa ter alguma conexão com a internet e acesso a alguma das três formas de controle desenvolvida;
2. A comunicação entre todos os elementos do projeto funcionar foi, também, um sucesso, uma vez que a comunicação proposta de utilização do Google Assistant para o controle das ações foi totalmente aceita pelo Adafruit IO, que gera os dados

necessários de controle toda vez que o mesmo é realizado, possibilitando uma outra ação, gerar uma planilha de dados de acesso, permitindo um controle do usuário dos horários e dias em que os cômodos foram acessados, podendo como uma melhoria futura do projeto, criar um meio de controle das lâmpadas em virtude da taxa de uso, podendo ser feita por inteligência artificial, por exemplo. Ou seja, o projeto mesmo que seja somente uma iniciação no mundo da domótica ele gera recursos para que seja implementado melhorias diversas.

Figura 17 – Tabela de Acesso

	A	B	C	D	E	F	G	H	I	J	K	L
1	ON	June 28, 2018 at 06:06PM										
2	OFF	June 28, 2018 at 06:07PM										
3	ON	June 28, 2018 at 06:07PM										
4	OFF	June 28, 2018 at 06:07PM										
5	ON	June 28, 2018 at 06:08PM										
6	OFF	June 28, 2018 at 06:08PM										
7	ON	June 28, 2018 at 06:10PM										
8	OFF	June 28, 2018 at 06:10PM										
9	ON	June 28, 2018 at 06:10PM										
10	OFF	June 28, 2018 at 06:10PM										
11	ON	June 28, 2018 at 06:10PM										
12	OFF	June 28, 2018 at 06:10PM										
13	ON	June 28, 2018 at 06:10PM										
14	OFF	June 28, 2018 at 06:10PM										
15	ON	June 28, 2018 at 06:11PM										
16	OFF	June 28, 2018 at 06:11PM										
17	ON	June 29, 2018 at 02:01PM										
18	OFF	June 29, 2018 at 02:01PM										
19	ON	June 29, 2018 at 02:02PM										
20	OFF	June 29, 2018 at 02:02PM										
21	ON	June 29, 2018 at 02:02PM										

Fonte: Próprio Autor.

## 5 CONCLUSÃO

A área de automação residencial é um tema com muitas possibilidades de estudo, o presente trabalho apontou somente um caminho a ser tomado, ligando temas como domótica e indústria 4.0, além de permitir, por meio de um projeto *Open Source*, que qualquer entusiasta consiga ter uma introdução a sistemas embarcados, mesmo que o foco seja automação residencial, tornando uma das preocupações do projeto menos relevante, que era o preconceito social com relação a sistemas robotizados.

Como já destacado o objetivo de criar um ambiente *Open Source* foi atingido com êxito, assim como a construção de um protótipo de automação residencial de baixo custo, mesmo apresentando alguns pontos de melhorias como já citados no Capítulo 4.

Uma vez que os controles do projeto funcionaram a proposta de valor inicial do trabalho relacionado ao conforto para a vida domiciliar foi atingido, assim como proporciona um maior controle sobre quais dispositivos estão ligados ou não, podendo gerar economia para a residência, esse por sua vez também é uma proposta de valor do projeto, portanto novamente atingindo um objetivo proposto.

Com relação a segurança não se pode dizer que o trabalho concluiu o objetivo com êxito, uma vez que o LDR não funcionou da melhor forma possível, não conseguindo trabalhar de forma paralela aos comandos de voz, e esta foi a solução implementada para segurança, portanto esta proposta de valor não foi atingida totalmente, em virtude das suas falhas.

Paralelamente os objetivos secundários foram todos atingidos, uma vez que foi respeitado um capítulo inteiro somente para demonstrar o funcionamento teórico tanto dos conceitos aplicados no projeto quanto dos componentes aplicados no mesmo. Vale ressaltar que houve também uma introdução teórica sobre o tema como um todo, não limitando o leitor ou entusiasta, que queira reproduzir ou melhorar ou protótipo, a seguir somente o rumo que este trabalho tomou.

Um dos objetivos que o projeto mais pecou foi com relação a eficiência energética, pois foi realizado um estudo com embasamento no consumo da placa de controle (NodeMCU) em diversos modos, considerando modos de *sleep* para menor consumo, porém não foi aplicado nenhum de fato no projeto, o código trabalha de forma constante de assinaturas, somente sendo espaçado por intervalos de tempo pré-determinados, não caracterizando um modo de menor consumo estudado.

Porém com relação a aplicação de uma interface de fácil manuseio para o usuário, assim como aplicação de uma tecnologia comum para o usuário e altamente cotidiana foi atingido, pois o sistema utiliza dispositivos móveis que muitos possuem e de forma relativamente nova para a área, que é por meio de comando de voz utilizando o Google Assistant, simulando alguns dispositivos comercializados porém de custo alto, o que fugiria totalmente da proposta inicial do trabalho.

Concluindo, o projeto conseguiu atingir grande parte dos objetivos propostos, mesmo que com falhas pequenas, o protótipo tem sua operação realizada e testada, portanto pode-se concluir que a ideia inicial do projeto foi atingida, inclusive na parte de ser um projeto que visava permitir que qualquer entusiasta pudesse reproduzir ou melhorar o mesmo, o trabalho foi desenvolvido em forma de tutorial para facilitar o entendimento do leitor com este intuito. Como melhorias futuras, o protótipo deverá consertar os erros mencionados e atingir um controle maior sobre os dados obtidos, gerando uma residência inteligente de fato, onde é capaz de aprender e agir de acordo com as ações dos seus usuários.

## REFERÊNCIAS

- ALIEXPRESS. **Base de dados para captura de valores.** Disponível em: <pt.aliexpress.com>. Acesso em: 28 de out. 2018.
- ALVES, J. A; MOTA, J. **Casas inteligentes.** Lisboa: Centro Atlântico, 2003.
- AMARAL, G.; ARBIX, G.; LINS, L. M.; SALERNO, M. S; ZANCUL, E. **O Brasil e a nova onda de manufatura avançada:** O que aprender com Alemanha, China e Estados Unidos. Vol.36. no.3. São Paulo: Novos estud. CEBRAP. 2017.
- ARDUINO. **Placa de desenvolvimento.** Disponível em: <https://store-cdn.arduino.cc/usa/catalog/product/cache/1/image/520x330/604a3538c15e081937dbfbd20aa60aad/a/0/a000066\_featured\_4.jpg>. Acesso em: 28 de out. 2018.
- BARROS, M. **MQTT – Protocolos para IoT.** 2015. Disponível em: <https://www.embarcados.com.br/mqtt-protocolos-para-iot/>. Acesso em: 28 out. 2018.
- BOLZANI, C. A. M. **Desenvolvimento de um simulador de controle de dispositivos residenciais inteligentes:** Uma Introdução aos Sistemas Domóticos. 2004. 131 f. Dissertação de Mestrado (Mestre em Engenharia) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2004a.
- BOLZANI, C. A. M. **Residências inteligentes.** São Paulo: Livraria da Física. 2004b.
- BOLZANI, C. A. M. Residências inteligentes e o novo contexto do século XXI. 2012. Publicação da Editora do Administrador. **Revista Automatizar.** ano IV, n. 6. Dezembro/ Janeiro 2012.
- BOLZANI, C. A. M. **Análise de arquiteturas e desenvolvimento de uma plataforma para residências inteligentes.** 2010. 155 f. Tese (Doutorado em Engenharia Elétrica) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2010.
- BRASIL, Banco Nacional do Desenvolvimento; BRASIL. Ministério do Planejamento, Desenvolvimento e Gestão; BRASIL, Ministério da Ciência, Tecnologia, Inovações e Comunicações. **Internet das coisas:** um plano de ação para o Brasil. Brasília, 2018. 79 p.
- CURVELLO, A. **Apresentando o módulo ESP8266.** Disponível em: <https://www.embarcados.com.br/modulo-esp8266/>. Acessado em 28 de out. de 2018.
- DEVMEDIA. **Protocolo TCP/IP.** Disponível em: <https://www.devmedia.com.br/via-de-mao-dupla-com-websockets/28281>. Acesso em: 28 out.2018.
- EFETIVIDADE, S. **Arduino ATmega328.** Disponível em: <http://static.efetividade.net/img/arduinoatmega328webbypighixxx-d5u4ape-650px-70932.jpg>. Acesso em: 28 out.2018.
- FACCIONI FILHO, M. **Internet das coisas.** Palhoça : UnisulVirtual. 2016. 56 p.

FILIFEFLOP. **Base de dados para captura de valores.** Disponível em: <filife flop.com>. Acesso em: 28 de out. 2018.

FILIFEFLOP. **Módulo de relé de 2 canais.** Disponível em: <https://uploads.filife flop.com/2013/02/imagem003.png>. Acesso em 26 out. 2018.

FINEP. **Kevin Ashton:** entrevista exclusiva com o criador do termo “Internet das Coisas”. Disponível em: <http://finep.gov.br/noticias/todas-noticias/4446-kevin-ashton-entrevista-exclusiva-com-o-criador-do-termo-internet-das-coisas>. Acesso em: 26 out. 2018.

FISCHER, C. S. **America calling:** A social history of the telephone to 1940. Merkeley, CA: University of California Press, 1994. 424 P.

GARCIA, F. D. **Introdução aos sistemas embarcados e microcontroladores.** Disponível em: <https://www.embarcados.com.br/sistemas-embarcados-e-microcontroladores/>. Acesso em: 26 out. 2018.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **PNAD contínua TIC 2016:** 94,2% das pessoas que utilizaram a Internet o fizeram para trocar mensagens. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/20073-pnad-continua-tic-2016-94-2-das-pessoas-que-utilizaram-a-internet-o-fizeram-para-trocar-mensagens>. Acesso em: 13 out. 2018.

IFUTURETECH. **Placa de desenvolvimento NodeMCU.** Disponível em: <http://www.ifuturetech.org/ifuture/uploads/2017/07/AMICA-NODEMCU-ESP8266-LUA-CP2102-WIFI-DEVELOPMENT-MODULE-IOT-gujarat.png>. Acesso em: 28 out. 2018.

LEIDEN, C.; WILENSKY, M. **TCP/IP for dummies.** 6 ed. Indianapolis, Indiana: Wiley Publishing, 2009.

LIMA, T. **O melhor microcontrolador para seu projeto.** Disponível em: <https://www.embarcados.com.br/microcontrolador/>. Acesso em: 26 out. 2018.

MADEIRA, D. **IIOT – Industrial internet of things.** Disponível em: <https://www.embarcados.com.br/iiot-industrial-internet-of-things/>. Acesso em: 26 out. 2018.

MAZZAFERRO, J. A E. Indústria 4.0 e a qualidade da informação. **Soldagem e Inspeção**, São Paulo , v.23, n.1, 2018.

MERLIN, L. **Base de dados para captura de valores.** Disponível em: <leroyermerlin.com.br>. Acesso em: 28 de out. 2018.

MONK, S. **Programação com arduino:** começando com Sketches. Porto Alegre: Bookman, 2013. 148p.

NASSAR, V.; VIEIRA, M. L. H. O compartilhamento de informações no transporte público com as tecnologias RFID e NFC: uma proposta de aplicação. **Revista Brasileira Gestão Urbana.**, Curitiba, v.9, n.2, 2017.

OLIVEIRA, Sérgio de. **Internet das coisas com ESP8266, Arduino e Raspberry PI**. São Paulo: Novatec, 2017. 240 p.

PISA, P. **O que é IP?** Disponível em: <<https://www.techtudo.com.br/artigos/noticia/2012/05/o-que-e-ip.html>>. Acesso em: 26 out. 2018.

POTTER, E. G. **Simulador de ambientes inteligentes: foco em home care** para pessoas com idade avançada. 2007. 107 f. Dissertação (Mestrado em Engenharia Elétrica) – Centro Universitário FEI, São Bernardo do Campo, 2007.

ROCKHEINE. **Internet das coisas**. Disponível em: <<https://www.devmedia.com.br/via-de-mao-dupla-com-websockets/28281>>. Acesso em: 26 out. 2018.

SILVA, L. F. C; SILVA, A. C. da. **IoT: contexto geral, presente e perspectiva futura – Parte 1**. Disponível em: <<https://www.embarcados.com.br/iot-contexto-geral-perspectiva-parte-1/>>. Acesso em: 14 de out. de 2018.

SILVA, L. F. C; SILVA, A. C. da. **IoT: contexto geral, presente e perspectiva futura – Parte 2**. Disponível em: <<https://www.embarcados.com.br/iot-contexto-geral-perspectiva-parte-2/>>. Acesso em: 14 out. 2018.

SILVA, D. S. da, **Desenvolvimento e implementação de um sistema de supervisão e controle residencial**. 2009. 62 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Rio Grande do Norte, Natal, 2009.

SOUZA, F. **Arduino Uno**. 2013. Disponível em: <<https://www.embarcados.com.br/arduino-uno/>>. Acesso em: 28 out. 2018.

TAVARES, P. L. **Redes de sensores sem-fio**. Disponível em: <[https://www.gta.ufrj.br/grad/02\\_2/Redes%20de%20sensores/Redes%20de%20Sensores%20sem-fio.doc](https://www.gta.ufrj.br/grad/02_2/Redes%20de%20sensores/Redes%20de%20Sensores%20sem-fio.doc)>. Acesso em: 26 out. 2018.

VIDAL, V. **10 protocolos de IoT que você deveria conhecer**. Disponível em: <<https://www.profissionaisti.com.br/2017/11/10-protocolos-de-iot-que-voce-deveria-conhecer/>>. Acesso em: 26 out. 2018.

WATERS, R. **FT interview with Google co-founder and CEO Larry Page**. Disponível em: <<https://www.ft.com/content/3173f19e-5fbc-11e4-8c27-00144feabdc0#axzz3HfAZgM8I>>. Acesso em: 18 out. 2018.

## ANEXO A – Código do Programa Completo

```

//Controla as conexões com o MQTT
#include <Adafruit_MQTT.h>
#include <Adafruit_MQTT_Client.h>
//Controla as conexões de WiFi da ESP32
#include <ESP8266WiFi.h>

//Definindo entradas e saídas
#define ledsala 12
#define ledcozinha 13
#define ledquarto 14
#define ledbanheiro 15
#define ledhall 2
#define ldr 0
int ldrValor = 0;

//Conectando a Rede de Internet
#define WIFI_SSID "AndroidAP"
#define WIFI_PASS "aninhaaa"
//Conectando ao MQTT
#define MQTT_SERV "io.adafruit.com"
#define MQTT_PORT 1883
#define MQTT_NAME "Jabbinhathehutt"
#define MQTT_PASS "70aa83e596714f2b971161455e77c190"

//Criando WiFiClient e Adafruit_MQTT_Client como variáveis globais
WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, MQTT_SERV, MQTT_PORT, MQTT_NAME,
MQTT_PASS);

Adafruit_MQTT_Subscribe sala = Adafruit_MQTT_Subscribe(&mqtt, MQTT_NAME
"/f/sala"); //sala = nome do feed
Adafruit_MQTT_Subscribe cozinha = Adafruit_MQTT_Subscribe(&mqtt, MQTT_NAME
"/f/cozinha"); //cozinha = nome do feed
Adafruit_MQTT_Subscribe quarto = Adafruit_MQTT_Subscribe(&mqtt, MQTT_NAME
"/f/quarto"); //quarto = nome do feed
Adafruit_MQTT_Subscribe banheiro = Adafruit_MQTT_Subscribe(&mqtt, MQTT_NAME
"/f/banheiro"); //banheiro = nome do feed
Adafruit_MQTT_Subscribe ligardesligar = Adafruit_MQTT_Subscribe(&mqtt,
MQTT_NAME "/f/ligardesligar"); //ligardesligar = nome do feed
Adafruit_MQTT_Subscribe hall = Adafruit_MQTT_Subscribe(&mqtt, MQTT_NAME
"/f/hall"); //hall= nome do feed

void setup() {

//Serial.begin(9600);
Serial.begin(115200);

```

```

//Connect to WiFi
Serial.print("\n\nConectando ao WiFi... ");
WiFi.begin(WIFI_SSID, WIFI_PASS);
while (WiFi.status() != WL_CONNECTED)
{
  delay(500);
}

Serial.println("Conectado");

//Inscrever-se no feed ligardesligar
mqtt.subscribe(&sala);
mqtt.subscribe(&cozinha);
mqtt.subscribe(&quarto);
mqtt.subscribe(&banheiro);
mqtt.subscribe(&ligardesligar);

//Setando as IO's
pinMode(LED_BUILTIN, OUTPUT);
pinMode(ledsala, OUTPUT);
pinMode(ledcozinha, OUTPUT);
pinMode(ledquarto, OUTPUT);
pinMode(ledbanheiro, OUTPUT);
pinMode(ledhall, OUTPUT);
pinMode(ldr, INPUT);
}

void loop() {

  ldrValor = analogRead(ldr);
  if (ldrValor <= 200) digitalWrite(ledhall,HIGH);
  else digitalWrite(ledhall,LOW);

  //Conecta e/ou Reconecta ao MQTT
  MQTT_connect();

  //Ler se o usuário enviou algum dado, ou
  //esperar 5 segundos para uma nova informação
  Adafruit_MQTT_Subscribe * subscription;
  while ((subscription = mqtt.readSubscription(5000)))
  {
    //Se tudo estiver OK...

    if (subscription == &sala)
    {
      //Salva o valor Serial na tela
      Serial.print("Sala: ");
      Serial.println((char*) sala.lastread);
    }
  }
}

```

```

//Se for "ON" liga a lâmpada
//Se não, desligue-a
if (!strcmp((char*) sala.lastread, "ON"))
{
    digitalWrite(ledsala, HIGH);
}
else
{
    digitalWrite(ledsala, LOW);
}
}

if (subscription == &cozinha)
{
    //Salva o valor Serial na tela
    Serial.print("Cozinha: ");
    Serial.println((char*) cozinha.lastread);

    //Se for "ON" liga a lâmpada
    //Se não, desligue-a
    if (!strcmp((char*) cozinha.lastread, "ON"))
    {
        digitalWrite(ledcozinha, HIGH);
    }
    else
    {
        digitalWrite(ledcozinha, LOW);
    }
}

if (subscription == &quarto)
{
    //Salva o valor Serial na tela
    Serial.print("Quarto: ");
    Serial.println((char*) quarto.lastread);

    //Se for "ON" liga a lâmpada
    //Se não, desligue-a
    if (!strcmp((char*) quarto.lastread, "ON"))
    {
        digitalWrite(ledquarto, HIGH);
    }
    else
    {
        digitalWrite(ledquarto, LOW);
    }
}

if (subscription == &banheiro)
{

```

```

//Salva o valor Serial na tela
Serial.print("Banheiro: ");
Serial.println((char*) banheiro.lastread);

//Se for "ON" liga a lâmpada
//Se não, desligue-a
if (!strcmp((char*) banheiro.lastread, "ON"))
{
    digitalWrite(ledbanheiro, HIGH);
}
else
{
    digitalWrite(ledbanheiro, LOW);
}
}

if (subscription == &ligardesligar)
{
    //Salva o valor Serial na tela
    Serial.print("Ligar/Desligar: ");
    Serial.println((char*) ligardesligar.lastread);

    //Se for "ON" liga a lâmpada
    //Se não, desligue-a
    if (!strcmp((char*) ligardesligar.lastread, "ON"))
    {
        digitalWrite(ledsala, HIGH);
        digitalWrite(ledcozinha, HIGH);
        digitalWrite(ledquarto, HIGH);
        digitalWrite(ledbanheiro, HIGH);
    }
    else
    {
        digitalWrite(ledsala, LOW);
        digitalWrite(ledcozinha, LOW);
        digitalWrite(ledquarto, LOW);
        digitalWrite(ledbanheiro, LOW);
    }
}
}

void MQTT_connect()
{
    int8_t ret;
    // Stop if already connected
    if (mqtt.connected())
    {
        return;
    }
}

```

```
Serial.print("Connectando ao MQTT... ");
uint8_t retries = 3;
while ((ret = mqtt.connect()) != 0) // conexão vai retornar 0 se estiver conectado
{
  Serial.println(mqtt.connectErrorString(ret));
  Serial.println("Tentando reconectar ao MQTT em 5 segundos...");
  mqtt.disconnect();
  delay(5000);
  retries--;
  if (retries == 0)
  {
    //Só espera se estiver conectado
    while (1);
  }
}
Serial.println("Conectado ao MQTT!");
}
```