

LIGA DE ENSINO DO RIO GRANDE DO NORTE
CENTRO UNIVERSITÁRIO DO RIO GRANDE DO NORTE
CURSO DE ESPECIALIZAÇÃO EM REDES DE COMPUTADORES

RAFAEL FERNANDES DE CASTRO

INTERNET DAS COISAS:
UM ESTUDO DE CASO EM AUTOMAÇÃO RESIDENCIAL

NATAL/RN

2017

RAFAEL FERNANDES DE CASTRO

INTERNET DAS COISAS:
UM ESTUDO DE CASO EM AUTOMAÇÃO RESIDENCIAL

Trabalho de Conclusão de Curso apresentado ao Centro Universitário do RN, como requisito final para obtenção do título de especialista em Redes de Computadores.

Orientador: Prof. M.Sc. Aluizio Ferreira da Rocha Neto.

NATAL/RN

2017

Catálogo na Publicação – Biblioteca da UNI-RN
Setor de Processos Técnicos

Castro, Rafael Fernandes de.

Internet das coisas: um estudo de caso em automação residencial /
Rafael Fernandes de Castro. – Natal, 2017.
34 f.

Orientador: Prof. M.Sc. Alúzio Ferreira da Rocha Neto.
Monografia (Especialização em Redes de Computadores) – Centro
Universitário do Rio Grande do Norte.

1. Internet of Things (IoT) – Monografia. 2. Automação residencial –
Monografia. 3. Raspberry PI – Monografia. 4. Microcontroladores –
Monografia. 5. ESP8266 – Monografia. I. Rocha Neto, Alúzio Ferreira da.
II. Título.

RN/UNI-RN/BC

CDU 004.72

RAFAEL FERNANDES DE CASTRO

INTERNET DAS COISAS:
UM ESTUDO DE CASO EM AUTOMAÇÃO RESIDENCIAL

Trabalho de Conclusão de Curso apresentado ao Centro Universitário do RN, como requisito final para obtenção do título de especialista em Redes de Computadores.

Aprovado em: ____ / ____ / ____

BANCA EXAMINADORA

Prof. M.Sc. Aluízio Ferreira da Rocha Neto
Orientador

Professor Convidado
Membro

Professor Convidado
Membro

RESUMO

Este trabalho tem o objetivo de apresentar os conceitos de microcontroladores e Internet das coisas (Internet of Things – IoT) voltados para automação residencial, a fim de entender quais as finalidades desses conceitos e como aplica-los em soluções do cotidiano. Serão apresentados alguns aspectos da IoT, termo que descreve um cenário em que vários objetos do nosso dia a dia estão se conectando a Internet e se comunicando para prover algum serviço. Com esta comunicação entre os objetos podemos otimizar e facilitar tarefas, como controlar e integrar diferente tipos de equipamentos de uma residência. Normalmente, o custo de uma implementação de uma tecnologia dessa nas residências é alto. Isso motivou o desenvolvimento do projeto de automação residencial utilizando tecnologias de baixo custo, no caso o Raspberry Pi 3 e um Módulo WiFi ESP8266 NodeMcu ESP-12, além de um uma aplicação Web. Esse projeto possibilita o morador a controlar a iluminação, alarme e um som ambiente e monitorar uma câmera de segurança.

Palavras-chave: Internet of Things (IoT). Automação residencial. Raspberry PI. Microcontroladores. ESP8266.

ABSTRACT

This paper aims to present the concepts of microcontrollers and the Internet of Things (IoT) for residential automation, in order to understand the purposes of these concepts and how to apply them in advanced solutions. Some aspects of IoT will be presented, a term that describes a scenario where several objects of our daily life are connecting to the Internet and communicating to provide some service. With this communication between objects we can optimize and facilitate tasks, such as controlling and integrating different types of equipment in a home. Typically, the cost of implementing such a technology in homes is high. This motivated the development of a residential automation project using low cost technologies, such as Raspberry Pi 3 and an ESP8266 NodeMcu ESP-12 WiFi Module, as well as a Web application. This project allows residents to control lighting, alarm and monitor a security camera.

Keywords: Internet of Things (IoT). Home automation. Raspberry Pi. Microcontrollers. ESP8266.

LISTA DE FIGURAS

Figura 1 – Módulo com quatro relés.....	15
Figura 2 – Placa Raspberry Pi 3.....	17
Figura 3 – Placa ESP8266 NodeMcu ESP-12.....	17
Figura 4 – Sensor de umidade e temperatura DHT11	19
Figura 5 – Montagem do circuito	20
Figura 6 – Instalação do Raspbian	21
Figura 7 – Configuração do Raspberry Pi	22
Figura 8 – Tela de preferências da IDE Arduino.....	24
Figura 9 – Tela para a instalação do suporte ao ESP8266 na IDE Arduino	25
Figura 10 – Tela para as instalações das bibliotecas	25
Figura 11 – Configuração do aplicativo MQTT Buddy	27
Figura 12 – Criação dos botões.....	28
Figura 13 – Esquemático tomada paralela com relé	30

LISTA DE TABELAS E QUADROS

Tabela 1 – História da Internet das Coisas	10
Tabela 2 – Consumo das placas Raspberry PI	13
Tabela 3 – Comparativo de consumo de diversos dispositivos portáteis	13
Quadro 1 – Definições de comunicação com o MQTT	26
Quadro 2 – Função para se conectar ao Servidor MQTT	26
Quadro 3 – Trecho de código responsável por ligar ou desligar o relé	29
Quadro 4 – Trecho de código que obtém a temperatura e a humidade	31

SUMÁRIO

1 INTRODUÇÃO	8
1.1 OBJETIVOS	8
1.1.1 Geral	8
1.1.2 Específicos	9
1.2 ORGANIZAÇÃO DO TRABALHO	9
2 REFERENCIAL TEÓRICO	10
2.1 INTERNET DAS COISAS	10
2.1.1 História	10
2.1.2 Conceitos básicos	11
2.1.3 Aplicação	12
2.2 RASPBERRY PI	12
2.3 RASPIBIAN	14
2.3.1 <i>General Purpose Input Output (GPIO)</i>	14
2.4 RELÉS	14
2.5 ESP8266	15
3 IMPLANTAÇÃO DO PROJETO	17
3.1 COMPONENTES	18
3.1.1 Raspberry Pi	18
3.1.2 ESP8266 NodeMcu	18
3.1.3 Sensor DHT11	18
3.1.4 Modulo Relé	19
3.2 MONTAGEM	19
3.3 INSTALAÇÃO DO SISTEMA OPERACIONAL E APLICATIVOS	20
3.3.1 Instalação do sistema operacional no Raspberry Pi	20
3.3.2 Instalando o MQTT Mosquitto no Raspberry	22
3.3.3 Instalando Bibliotecas no IDE do Arduino	23
3.4 FUNCIONAMENTO DO SISTEMA	26
4 CONSIDERAÇÕES FINAIS	32
REFERÊNCIAS	33

1 INTRODUÇÃO

Automação residencial é uma tecnologia onde permite a integração entre diversos equipamentos motorizados e automatizados conversando em si, também conhecida como domótica. O termo vem da junção da palavra “domus” (casa) com “robótica” (controle automatizado de algo). Assim, permitindo ligar ou deligar todos os equipamentos com um só comando, e simplificando a vida diária das pessoas através do controle por smartphones ou tablets.

Com a evolução das tecnologias estamos entrando na era da Internet das Coisas (ou *Internet of Things* - IoT) onde qualquer objeto poderá estar conectado à Internet e com a possibilidade de centralizar todo o controle em um só lugar, buscando conforto, comodidade e segurança.

O preço para implantar uma tecnologia de automação residencial geralmente é alto. Para baratear essa implantação, este trabalho propõe o uso de hardwares de baixo custo, tais como: uma placa Raspberry Pi 3, uma placa de relés e um módulo WiFi ESP8266 NodeMcu ESP-12. Com esses dispositivos é possível controlar lâmpadas, sensores, câmeras, som e alarmes em uma aplicação web acessível a partir de um computador, smartphone ou tablet, tanto interno como externo à residência.

Esse trabalho mostra que essas ferramentas podem ser utilizadas para tal fim e com um custo bem mais acessível.

1.1 OBJETIVOS

1.1.1 Geral

Este trabalho tem como objetivo principal a apresentação de conceitos e as tecnologias necessárias para um projeto de automação residencial como o estudo de caso da utilização de um sistema web em um smartphone para controlar a iluminação, sensores, câmera de segurança e som.

1.1.2 Específicos

Para atingir o objetivo geral, os seguintes objetivos específicos foram seguidos:

- Compreender como funciona uma residência automatizada;
- Levantar os equipamentos a qual podemos manipular;
- Construir o projeto para automatizar esses equipamentos com o Raspberry Pi 3, uma placa de rele e um módulo WiFi ESP8266 NodeMcu ESP-12;
- Desenvolver uma aplicação web para controlar os equipamentos.

1.2 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado da seguinte forma: O capítulo 2 apresenta o referencial teórico onde apresenta todos os conceitos e tecnologias utilizadas para a elaboração do projeto. O capítulo 3 apresenta um estudo de caso para a elaboração do projeto. O capítulo 4 apresenta o resultado final do projeto.

2 REFERENCIAL TEÓRICO

2.1 INTERNET DAS COISAS

2.1.1 História

Segundo Rampim (2016) autora do livro *Internet das coisas sem mistério*, um dos principais conceitos sobre Internet das coisas (IoT) veio do Kevin Ashton diretor e co-fundador da empresa Auto-ID Center, que desenvolveu o projeto RFID um identificador de rádio frequência. Na apresentação do projeto do seu sistema em uma palestra para Procter & Gamble, Kevin colocou como título a expressão *Internet of Things* (Internet das Coisas). Após alguns anos a ITU (União internacional de Telecomunicações) publicou um relatório evidenciando o conceito de Internet das Coisas, sugerindo que poderia conectar os objetos do mundo, tanto de forma sensorial como inteligente, por meio da comunicação de tecnologias.

Ela fala que em 2009 muitos autores consideram o ano de nascimento da Internet das Coisas, onde segundo a Cisco *Internet Business Solutions Group* (IBSG) havia mais objetos conectados do que a própria população mundial (RAMPIM, 2016).

Abaixo uma tabela de História da Internet das Coisas retirada do livro.

Tabela 1 – História da Internet das Coisas

Ano	Fatos
1999	Kevin Ashton, em uma apresentação para P&G apresenta a nova ideia do sistema RFID na cadeia de suprimentos e liga a Internet ao contexto.
2003	Lançamento da EPC Network (EPCglobal), a RFID é evidenciada entre as tecnologias AutoID existentes.
2005	Mandato Walmart e US DoD. Primeiro relatório sobre IoT da ITU.
2008	Lançamento da Aliança IPSO para promover o uso do IP nas redes dos objetos inteligentes.
2009	A Internet das Coisas nasce. Pela primeira vez na História da Internet, havia mais coisas conectadas do que pessoas.

Fonte: Rampim (2016).

2.1.2 Conceitos básicos

Segundo Rampim (2016) não existe um conceito único para Internet das Coisas, cada especialista tem uma interpretação diferente. Ela cita dois conceitos de âmbito mundial sobre o IoT.

IoT sob a perspectiva da ITU (RAMPIM, 2016, p. 20):

Uma infraestrutura global para a sociedade da informação permitindo serviços avançados por meio-da-interligação das coisas (físicas e virtuais) baseadas na interoperabilidade das tecnologias de informação e de comunicação existente e em evolução.

Nota 1: por meio da exploração de capacidade de identificação, captura de dados, processamento e comunicação, a IoT faz pleno uso das coisas para oferecer serviços a todos os tipos de aplicações, garantindo o cumprimento dos requisitos de segurança e privacidade.

Nota 2: a partir de uma perspectiva mais ampla, a IoT pode ser compreendida como uma visão com implicações tecnológicas e sociais.

CASAGRAS: Conceito apresentado pelo projeto Coordination and Support Action for Global RFID-related Activities and Standardisation (RAMPIM, 2016, p. 20):

[...] uma infraestrutura de rede global, interligando objetos físicos e virtuais por meio da exploração de captura de dados e capacidades de comunicação. Essa infraestrutura inclui a Internet existente e em evolução, bem como os desenvolvimentos de rede. Ela oferecerá identificação de objetos específica, capacidade de sensoriamento e de conexão como base para o desenvolvimento de aplicações e serviços independentes cooperativos. Estes serão caracterizados por um elevado grau de captura autônoma de dados, transferência de eventos, conectividade e interoperabilidade de rede.

Em uma entrevista da Inovação em Pauta com Kevin Ashton, foi perguntado do que se trata a IoT, o Kevin Ashton respondeu (FINEP, 2017):

O fato de eu ter sido provavelmente a primeira pessoa a dizer “Internet das Coisas” não me dá nenhum direito de controle sobre como os outros usam a frase, obviamente. 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.

2.1.3 Aplicação

Entre diversas possibilidades existentes podemos citar alguns exemplos na automação residencial, como enviar um comando via smartphone para ligar ou desligar uma lâmpada, ativar sensores de presença até mesmo ligar uma cafeteira entre outras aplicações que são infinitamente possíveis.

2.2 RASPBERRY PI

O Raspberry Pi, fabricada no Reino Unido pela Fundação Raspberry Pi, foi lançada em 2012 como um microcomputador do tamanho de um cartão de crédito. Seu objetivo principal é incentivar o ensino científico da computação para crianças. Com o intuito de alcançar um público maior, o preço teria que ser bem baixo, para isso convenceram a empresa Broadcom a vender os processadores que são usados nas placas com um valor menor (MACIEL, 2014).

O Raspberry Pi é um projeto de hardware aberto onde a família de seu processador é ARM, sua versão atual está na 3, onde tem um poder de processamento de 1,2 GHz de 64 bits, integrado com interfaces WiFi e Bluetooth, 1GB de memória RAM, quatro portas USB, 40 pinos de entrada e saída e saída HDMI. Tem capacidade de decodificar vídeo H.264 Full-HD em tempo real (OLIVEIRA, 2017).

Segundo Oliveira (2017), o Raspberry Pi inicialmente foi proposto como uma solução de computador popular, utilizando uma versão do sistema operacional Linux, com interface gráfica e pacotes de aplicativos de código aberto. Com seu sucesso, abriu espaço para várias outras aplicações. Com seu tamanho reduzido, suas interfaces de entrada e saída, facilitam para várias aplicações.

Com o sistema operacional Linux, possibilita a integração com vários periféricos como câmera, teclado, mouses, impressoras, monitores, leitor biométrico, entre outros. Com ambiente de programação para diversos tipos de linguagem, incluindo interface gráfica e APIs para integrar com diversos tipos de dispositivos, o Linux possibilita também a disponibiliza de servidores de bancos de dados, servidores e protocolos de rede para as mais diversas aplicações (OLIVEIRA, 2017).

O consumo de energia do Raspberry Pi em relação a outros microcontroladores como o ESP8266 é mais elevado, mas comparado com

computadores pessoais como notebook e desktop o consumo é bem menor, apesar de sua placa ter vários periféricos incorporados à placa, como saída HDMI, entrada USB, WiFi e Bluetooth (OLIVEIRA, 2017).

Inicialmente o Raspberry Pi foi lançado em versões A+ e B+, onde a diferença era o número de periféricos integrados na placa, atualmente com novas versões a 2 e 3, onde aumentou o seu poder de processamento e periféricos incluído como o WiFi e Bluetooth integrado na versão 3, sem a necessidade de placas adicionais ocasionando mais consumo de energia (OLIVEIRA, 2017).

A tabela 2 apresenta o consumo das versões do Raspberry Pi e situações de Inicialização, ocioso, vídeo e sobrecarga:

Tabela 2 – Consumo das placas Raspberry Pi

Consumo (mA)	Pi1 (B+)	Pi2 B	Pi3 B	Zero
Inicialização	220	220	350	150
Ocioso	200	220	300	100
Vídeo	220	280	330	160
Sobrecarga	350	820	1340	350

Fonte: Rampim (2016).

A tabela 3 apresenta um comparativo entre os principais módulos para IoT, comparando placas de consumos extremamente baixo (Telos Motes) e BÇE nRF51822 a laptops.

Tabela 3 – Comparativo de consumo de diversos dispositivos portáteis

Módulo	Consumo			
	Sleep	Sem comunicação	Recebendo	Transmitindo
Telos Motes	15 μ W	3 mW	41 mW	41 mW (0 dBm)
BLE nRF51822	1,8 μ W	7 mW	39 mW	31,4 mW (0dBm)
ESP8266	33 μ W	49,5 mW	165 mW 184,8 mW	561 mW (17 dBM) 462 mW (15 dBM) 396 mW (13 dBM)
Arduino Nano	115 μ W	75 mW		
Raspberry Pi	-	1.150 mW 1.500 mW		

		1.650 mW		
Smartphone	150 mW	500 mW	750 mW	1.000 mW
Notebook	500 mW	14 W	14 W	14 W

Fonte: Rampim (2016).

2.3 RASPIBIAN

Distribuição Linux oficial suportada diretamente pelo site do Raspberry Pi. Ela pode ser baixada diretamente do site oficial (www.raspberrypi.org), disponível em duas versões completa e a *lite*, uma com interface gráfica e outra não, respectivamente. Uma das vantagens de usar a versão gráfica é que pode ser usado como computador pessoal e para utilizar como ambiente de desenvolvimento para aplicações (OLIVEIRA, 2017).

2.3.1 General Purpose Input Output (GPIO)

GPIO são portas lógicas que podem ser configuradas como entrada e saída de dados, por software, com interfaces digitais simples onde normalmente assumem valor 0 ou 1, representando nível de tensão baixo e alto na maioria dos casos (OLIVEIRA, 2017).

Um exemplo de aplicação de entrada para GPIO é ligar uma botoeira, utilizando um resistor para manter a entrada em baixa tensão (Terra), enquanto não é acionada. Ao acionar leva a tensão de entrada para nível alto, onde é recebido na entrada do microcontrolador. Como interface de saída o GPIO podem ser usadas para acionar dispositivos, acender leds ou acionar outros componentes (OLIVEIRA, 2017).

2.4 RELÉS

Utilizado para controlar dispositivos elétricos, como a corrente de saída das GPIOs é muito baixa não sendo possível acionar diretamente os dispositivos, a não ser leds ou dispositivos de baixíssima corrente (OLIVEIRA, 2017).

A figura 1 apresenta um módulo com quatro relés, ele vem com acoplamento óptico para evitar retorno do relé para a entrada GPIO, onde poderia queimar. Para

zerar a contracorrente induzida pela bobina ela conta com um diodo invertido em paralelo com a bobina K.

Figura 1 – Módulo com quatro relés



Fonte: Elaborado pelo autor (2017).

2.5 ESP8266

Produzido pela fabricante chinesa Espressif, o ESP8266 é um microcontrolador de 32 bits que inclui um núcleo microprocessado Tensilica L106. Sua frequência padrão é de 80 MHz, podendo chegar até a 160 MHz. Seu protocolo WiFi utiliza 20% da capacidade de processamento desse processador, os 80% ficam livre para a utilização de aplicações do usuário. Tem uma memória interna de 50 kB para dados programas já retirando o espaço necessário para o WiFi. Tendo 4 MB disponível para utilizar no programa principal, podendo ser atualizado em funcionamento chamado de OTA (*Over The Air*) e 512 kB onde só podem ser atualizados via cabo, em procedimento de atualização (OLIVEIRA, 2017).

O ESP8266 tem uma tensão nominal de funcionamento de 3,3 volts, seu consumo de energia é de 20 μ A no modo sleep cerca de 50mA conectado a um AP WiFi em modo de recepção. Podendo chegar até 170 mA em transmissão em

potência máxima de 17 dBm, sua interface WiFi pode atuar no modo AP, como cliente, no modo ad hoc ou WiFi Direct (OLIVEIRA, 2017).

O ESP8266 tem 17 interfaces GPIO de entrada e saída de dados, tem quatro interfaces de saída PWM e uma entrada analógica com 10 bits de precisão, por uma tensão de 3,3 volts onde é sua tensão de funcionamento (OLIVEIRA, 2017).

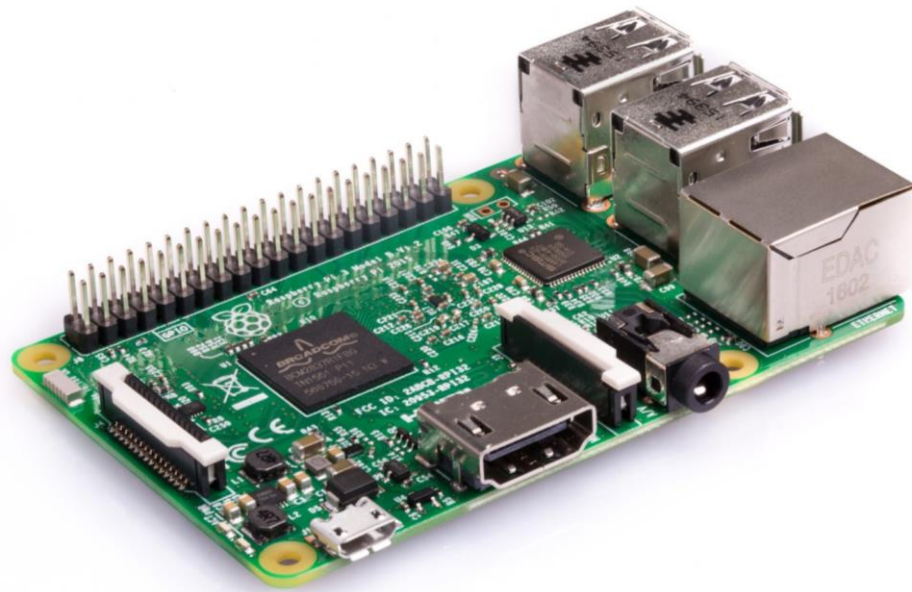
Existem vários módulos disponíveis no mercado para o microcontrolador ESP8266, a versão mais simples pode chegar a 1 dólar em sites chineses como Aliexpress (<https://www.aliexpress.com>), DealeXtreme (<http://www.dx.com>) entre outros (OLIVEIRA, 2017).

O módulo NodeMCU é o mais popular, a programação pode ser feita em linguagem LUA. Linguagem desenvolvida foi desenvolvida no Brasil por professores da PUC-Rio (Pontifícia Universidade Católica do Rio de Janeiro), é interpretada e tem uma sintaxe de fácil aprendizagem e integração, ela já vem integrada um conversor USB-Serial e regulador de tensão (OLIVEIRA, 2017).

3 IMPLANTAÇÃO DO PROJETO

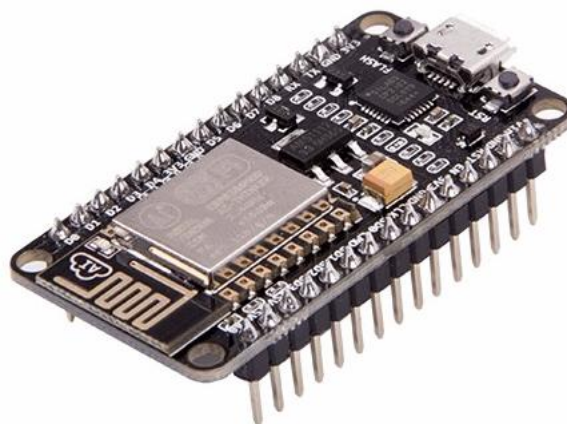
O projeto inicia com a análise das tecnologias IoT para implementar uma automação residencial. Uma das melhores opções foi utilizar o Raspberry Pi visto na Figura 2 e o ESP8266 NodeMcu visto na Figura 3, se comunicando através do protocolo MQTT.

Figura 2 – Placa Raspberry Pi 3



Fonte: Placa... (2017).

Figura 3 – Placa ESP8266 NodeMcu ESP-12



Fonte: Placa... (2018)

O projeto inicial foi para controlar as luzes de um quarto e monitorar a temperatura e umidade do mesmo através de um celular, demonstrando que as placas utilizadas de baixo custo podem sim ser implementadas na automação residencial.

3.1 COMPONENTES

Neste capítulo serão descritos os componentes utilizados para a construção do projeto.

3.1.1 Raspberry Pi

Para esse projeto alguns itens foram escolhidos. O primeiro deles é o Raspberry Pi 3 (Figura 3), escolhido pelo seu baixo custo e sua alta performance entre os modelos disponíveis para IoT, o mesmo dispõe de WiFi e Bluetooth integrado que para um projeto de automação isso é ideal, foi utilizado para ser instalado o broker Mosquitto para realizar a comunicação entre a placa ESP8266 NodeMcu ou qualquer outra aplicação que venha ser utilizado.

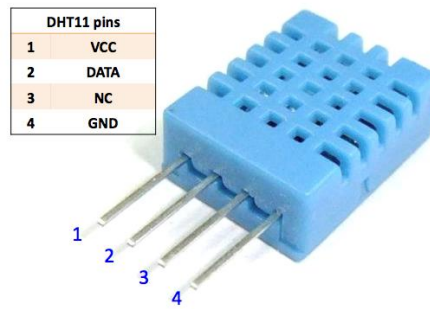
3.1.2 ESP8266 NodeMcu

Esse módulo foi escolhido pelo seu baixíssimo custo no mercado, pelo seu tamanho reduzido e por disponibilizar de placa WiFi integrada, utilizado para se comunicar com o módulo relé e o sensor DHT11.

3.1.3 Sensor DHT11

Utilizado sensor DHT11 (figura 4) para coletar a temperatura e umidade do ambiente, foi escolhido esse sensor por dispor da coleta das duas informações.

Figura 4 – Sensor de umidade e temperatura DHT11



Fonte: Sensor... (2018).

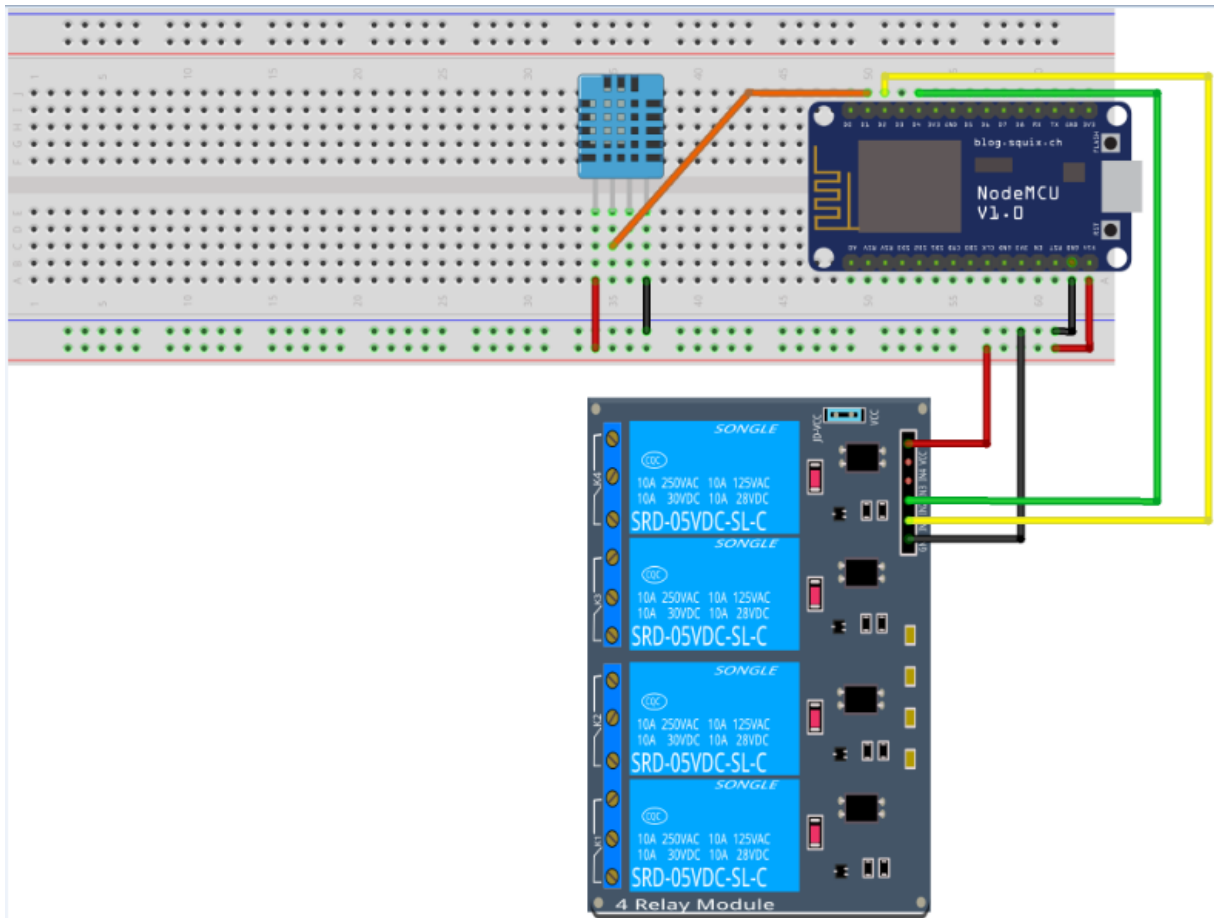
3.1.4 Modulo Relé

Utilizado o modulo relé (figura 1), para controlar as lampas através do ESP8266 NodeMcu.

3.2 MONTAGEM

Após a escolha dos micros controladores e componentes, iniciou-se a montagem, foi utilizado o protoboard para montar o circuito, exemplo de como ficou na figura 5.

Figura 5 – Montagem do circuito



Fonte: Elaborado pelo autor (2017).

3.3 INSTALAÇÃO DO SISTEMA OPERACIONAL E APLICATIVOS

Para o desenvolvimento deste projeto fez-se necessário instalação do sistema operacional e um broker no Raspberry Pi e instalações de bibliotecas para o ESP8266.

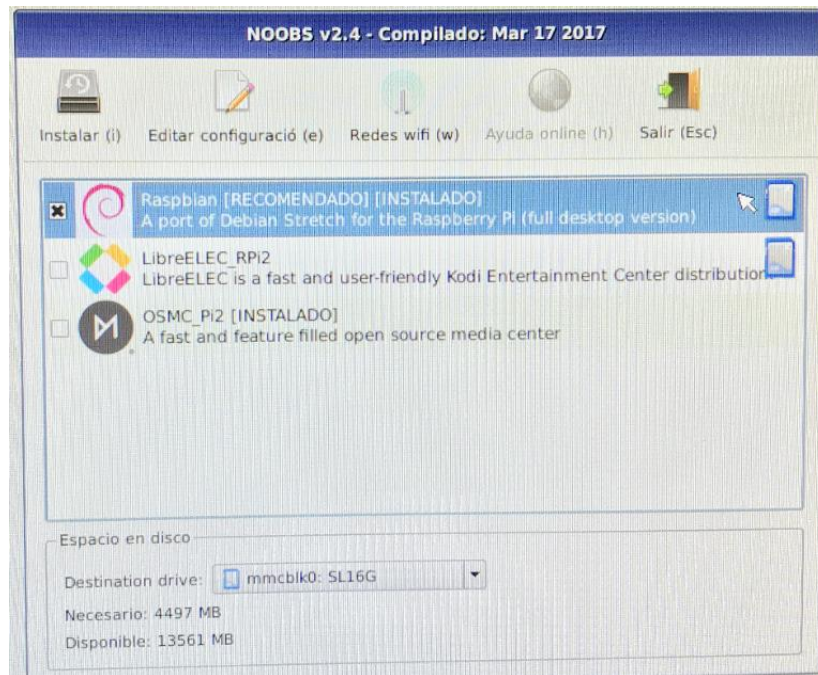
3.3.1 Instalação do sistema operacional no Raspberry Pi

Foi optado pelo sistema operacional Raspbian, onde encontra-se disponível para download no site do Raspberry Pi (<https://www.raspberrypi.org>). Existe duas formas de instalar o Raspbian, uma através do NOOBS, onde pode ser instalado outros sistemas operacionais simultâneos e outra instalando via própria imagem do Raspbian. Nesse projeto foi instalado via NOOBS. Após efetuar o download da imagem do NOOBS, foi utilizado o programa Win32 Disk Imager para extrair a

imagem para o cartão de memória, o programa pode ser baixado através do link (<https://sourceforge.net/projects/win32diskimager/>).

Após a extração, foi inserido o cartão de memória no Raspberry Pi, ao iniciá-lo foi escolhido o sistema operacional raspbian para instalação conforme figura 6.

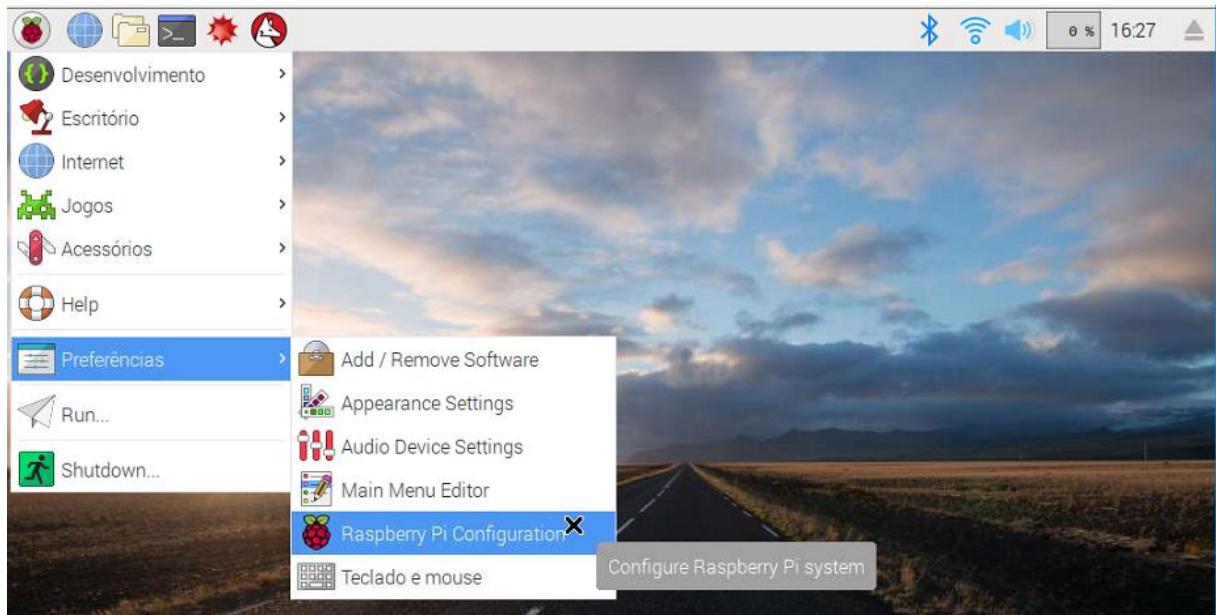
Figura 6 – Instalação do Raspbian



Fonte: Elaborado pelo autor (2017).

Após inicializar do Raspberry Pi foram realizadas algumas configurações iniciais conforme Figura 7 (Pressionando o botão inicial > Preferências > Raspberry Pi Configuration) como alterar a senha de usuário, pois a mesma vem com a senha padrão “raspberrry”, alterado a linguagem, alterado a linguagem do teclado, habilitado o SSH e VNC para acesso remoto.

Figura 7 – Configuração do Raspberry Pi



Fonte: Elaborado pelo autor (2017).

Após configurações iniciais foi realizado a atualização do sistema operacional via os comandos:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

3.3.2 Instalando o MQTT Mosquitto no Raspberry

Para realizar a comunicações entre os dispositivos IoT, foi utilizado o protocolo MQTT (Message Queue Telemetry Transport), como broker foi instalado o Mosquitto no Raspberry Pi, assim transformando ele como um servidor para enviar e receber e receber mensagens dos dispositivos IoT.

Depois de atualizar o sistema operacional com os comandos demonstrado acima, foi definido um IP fixo no Raspberry Pi, para evitar que o IP seja alterado e os equipamentos conectados ao servidor MQTT percam conexão, para isso podemos fixar no roteador ou no próprio Raspberry Pi acessando arquivo de configuração a seguir:

- interface {interface de rede que vai ser utilizada}
- static ip_address= {IP}/{máscara de rede}
- static routers= {IP do seu roteador}

- `static domain_name_servers= {IP do seu roteador} 8.8.8.8 8.8.4.4`

Para instalar o broker MQTT Mosquitto, foi utilizado os comandos abaixo:

```
$ sudo apt-get install mosquitto -y
$ sudo apt-get install mosquitto-clients -y
```

Para realizar a configuração do Mosquitto, foi acessado o arquivo `mosquitto.conf` e acrescentado as linhas a seguir e comentado (#) a linha que contém `(include_dir /etc/mosquitto/conf.d)`.

```
allow_anonymous false
password_file /etc/mosquitto/pwfile
listener 1883
```

A primeira linha é definida para não permitir que façam *publish* e *subscribe* sem autenticação, a segunda linha está informando qual o arquivo que vai conter os usuários e senhas, a terceira linha define qual a porta que o *broker* vai utilizar.

Para criar o usuário e senha, utiliza-se o comando a baixo:

```
$ sudo mosquitto_passwd -c /etc/mosquitto/pwfile {Nome de Usuário}
```

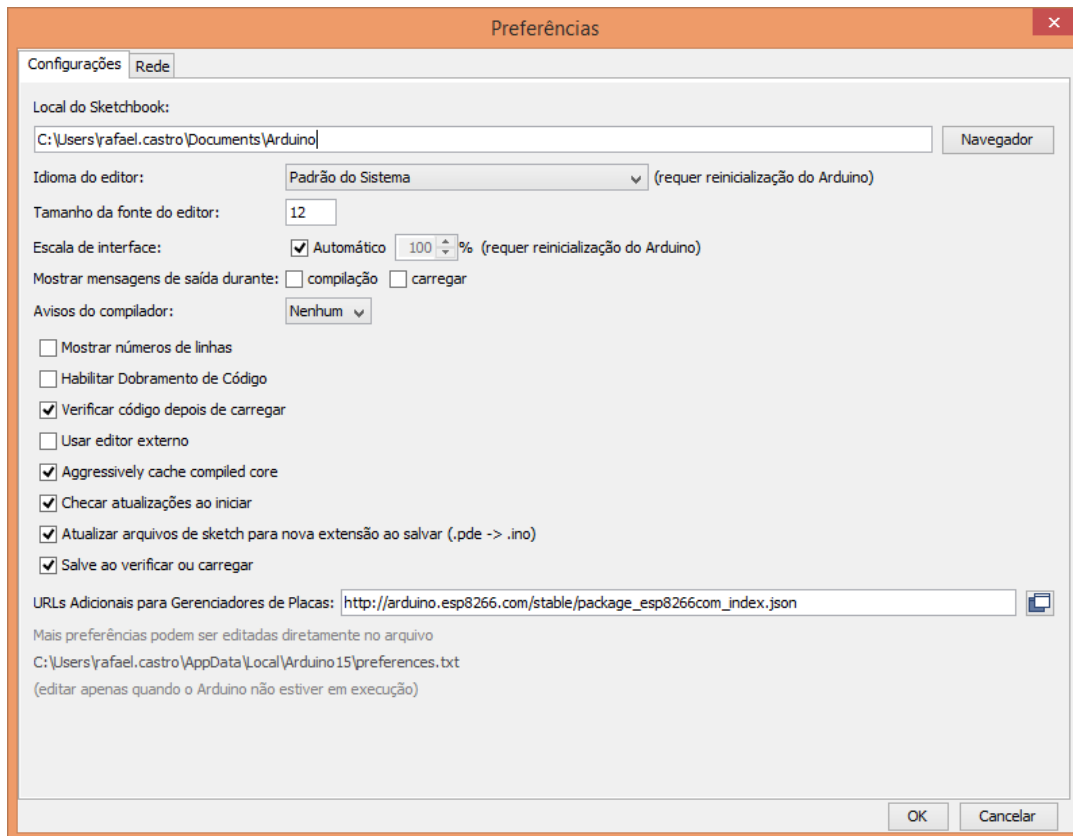
Após todas as configurações e criação de usuário é preciso reiniciar o Raspberry Pi e ao iniciar iniciar o broker Mosquitto ditando no terminal:

```
$ mosquitto
```

3.3.3 Instalando Bibliotecas no IDE do Arduino

Foi utilizado o IDE do Arduino (<http://www.arduino.cc>) para realizar a programação no ESP8266 NodeMcu ESP-12, para que o mesmo seja reconhecido é na IDE é necessário realizar a instalação da placa, para isso é preciso ir em arquivos e depois Preferências, será apresentado a tela da figura 8. Em seguida, adicionado ao campo URLs Adicionais de Gerenciadores de Placas a URL `<http://arduino.esp8266.com/stable/package_esp8266com_index.json>`.

Figura 8 – Tela de preferências da IDE Arduino



Fonte: Elaborado pelo autor (2017).

Depois, deve-se acessar o menu Ferramentas, depois placa e Gerenciador de Placas. Após isso deve-se procurar a placa *esp8266 by ESP8266 Community*, conforme mostra a Figura 9, e escolher a opção instalar, após a instalação é possível compilar e carregar o módulo ESP8266 NodeMCU.

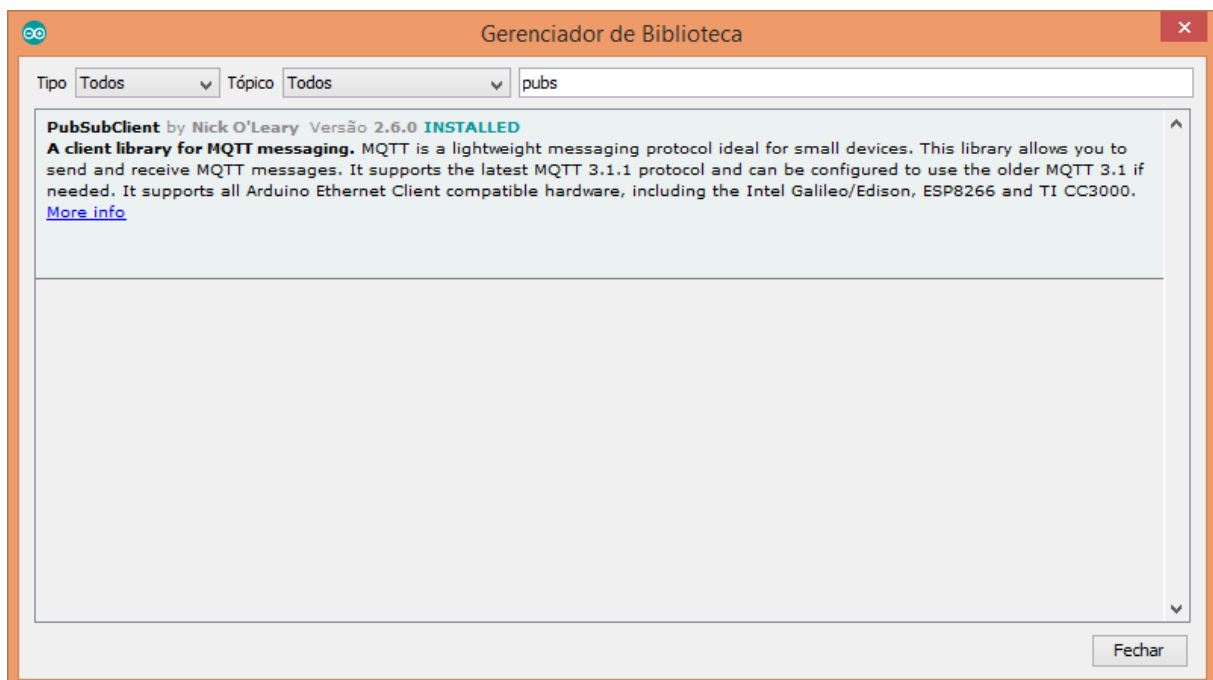
Para o projeto foi instalado as bibliotecas ESP8266WiFi, DHT e PubSubclient, para instalar é preciso ir no menu sketch, depois escolher a opção incluir biblioteca e gerenciar bibliotecas, conforme Figura 10. Para facilitar a busca, tem um campo de pesquisa. A primeira biblioteca é necessária para que o módulo ESP8266 se conecte via WiFi, a segunda usada para leitura dos dados do sensor DHT11 e a terceira biblioteca é para realizar a comunicação através do protocolo MQTT, que nesse projeto foi instalado um broker Mosquitto no Raspberry Pi.

Figura 9 – Tela para a instalação do suporte ao ESP8266 na IDE Arduino



Fonte: Elaborado pelo autor (2017).

Figura 10 – Tela para as instalações das bibliotecas



Fonte: Elaborado pelo autor (2017).

3.4 FUNCIONAMENTO DO SISTEMA

Para que tudo funcionasse foi necessário a integração entre os hardwares, foi criado um servidor *brocker* Mosquitto para se comunicar através do protocolo MQTT com os demais hardwares utilizando o *publish/subscriber* para trocar mensagem. Para a comunicação entre o cliente (ESP8266) / servidor (Raspberry) foi definido o código no quadro 1, IP do servidor e a porta de comunicação e criado uma função para se comunicar com o servidor conforme quadro 2:

Quadro 1 – Definições de comunicação com o MQTT

```
const char* SERVIDOR_MQTT = "10.0.0.105"; // IP do Servidor MQTT
int SERVIDOR_MQTT_PORTA = 1883; // Porta do Servidor MQTT
```

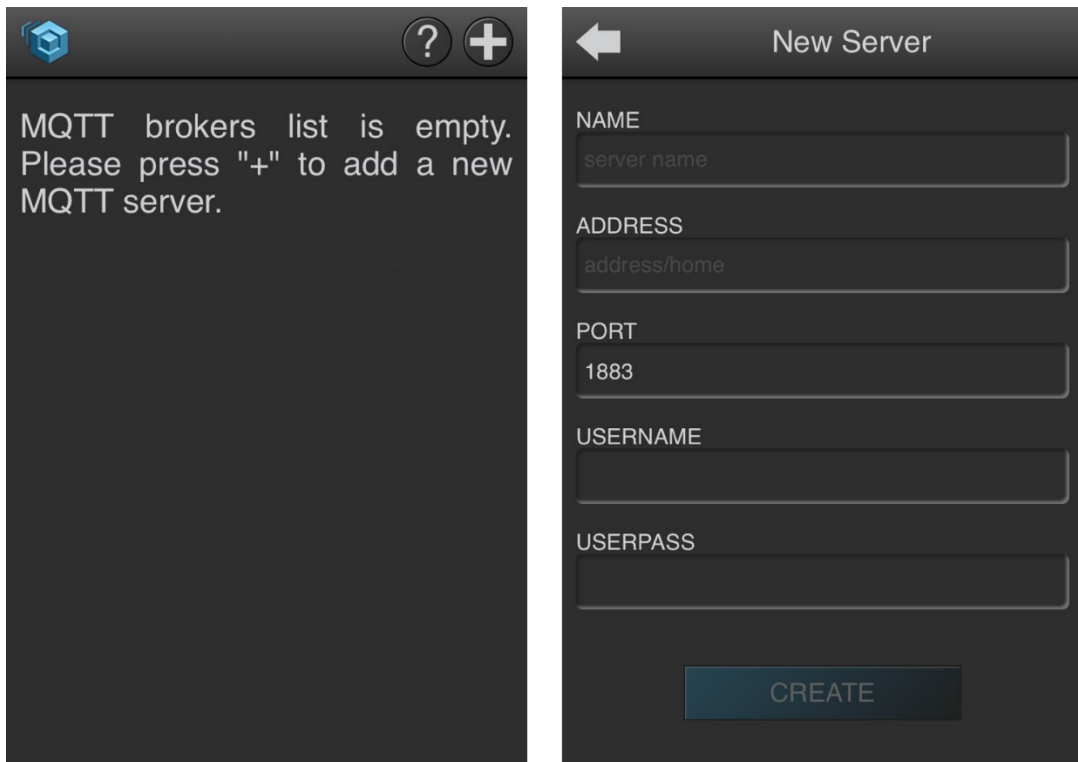
Fonte: Elaborado pelo autor (2017).

Quadro 2 – Função para se conectar ao Servidor MQTT

```
void MQTT() {
  MQTT.setServer(SERVIDOR_MQTT, SERVIDOR_MQTT_PORTA);
  MQTT.setCallback(mqtt_callback);
}
```

Fonte: Elaborado pelo autor (2017).

A comunicação aplicativo cliente, foi utilizado o aplicativo MQTT Buddy, disponível para IOS, o mesmo faz a comunicação com o broker mosquitto instalado no Raspberry Pi, que por sua vez faz comunicação com ESP8266 para o acionamento do rele e coleta da temperatura. Para configurar clicamos no mais no canto superior direito da aplicação e configuramos com os dados do servidor conforme Figura 11.

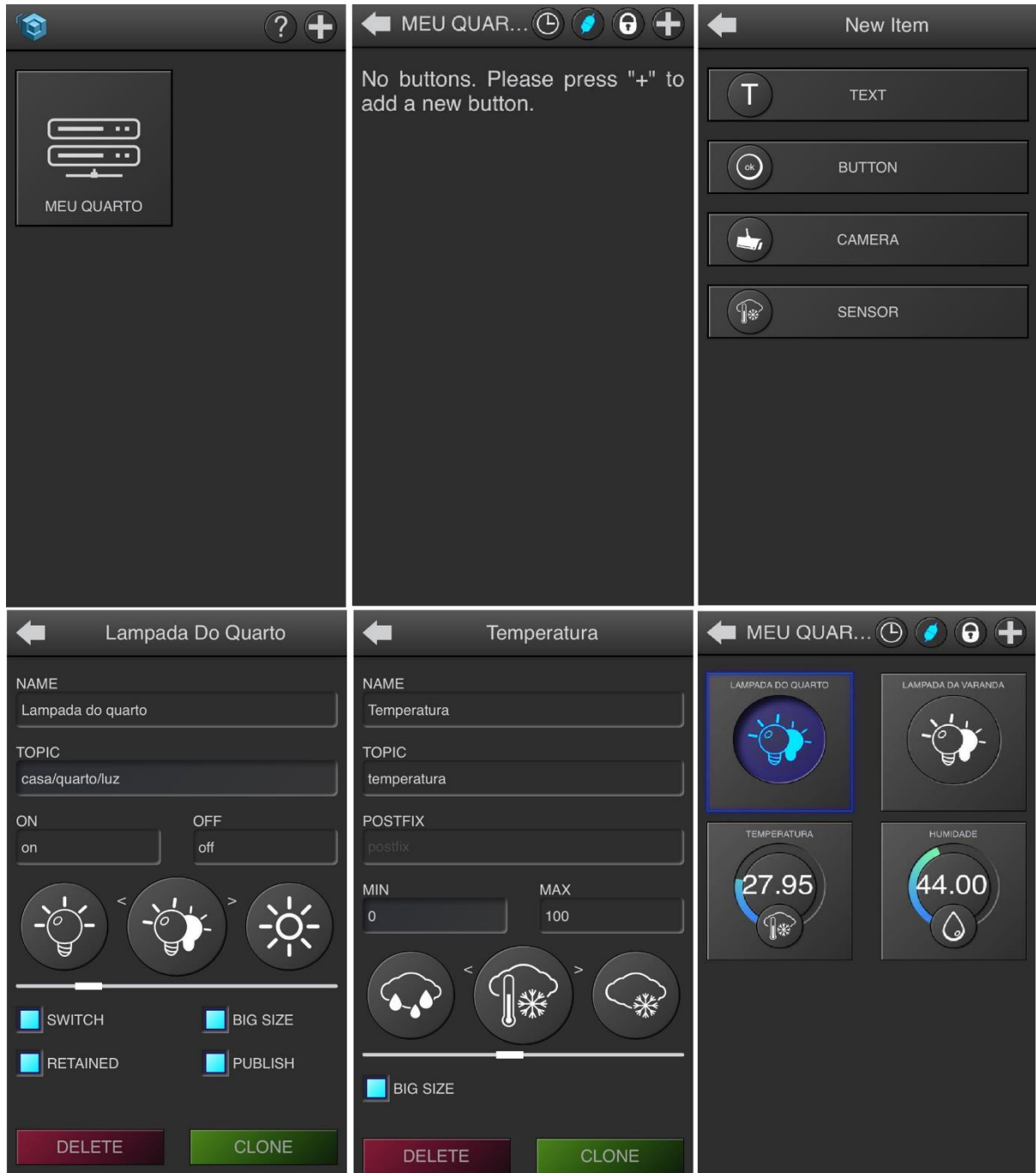
Figura 11 – Configuração do aplicativo MQTT Buddy

Fonte: Elaborado pelo autor (2017).

Aposto a configuração foi criado os botões para acionamento e monitoramento de temperatura, conforme figura 12.

Ao acionar o botão “lâmpada do quarto”, o mesmo envia uma mensagem “on” ou “off” para o servidor MQTT que por sua vez envia para ESP8266 que interpreta a mensagem e aciona o relé, o trecho do código responsável por ligar e desligar os relés pode ser visualizado no quando 3.

Figura 12 – Criação dos botões



Fonte: Elaborado pelo autor (2017).

Quadro 3 – Trecho de código responsável por ligar ou desligar o relé

```

void mqtt_callback(char* topic, byte* payload, unsigned int length) {
    String mensagem;
    for (int i = 0; i < length; i++) {
        char c = (char)payload[i];
        mensagem += c;
    }
    Serial.println("Tópico => " + String(topic) + " | Valor => " + String(mensagem));
    if (mensagem == "on") {
        digitalWrite(D4, 1);
    } else if (mensagem == "off") {
        digitalWrite(D4, 0);
    }
    if (mensagem == "on2") {
        digitalWrite(D5, 1);
    } else if (mensagem == "off2"){
        digitalWrite(D5, 0);
    }
    Serial.flush();
}

```

Fonte: Elaborado pelo autor (2017).

Na figura 13 pode ser observado como ficou a instalação do relé a rede elétrica. Para que o acionamento seja realizado tanto pela aplicação como pelo interruptor, foi utilizado o esquema semelhante ao acionamento de lâmpada paralelo com interruptor de três polos (*three-way*), só que em vez de utilizar outro interpor é utilizado o relé como pode ser observado no esquemático na figura 13, o neutro vai direto para lâmpada e o fase vai para polo do meio do interruptor, na lâmpada o fase chega do relé onde o fio sai do pino do meio, chamado de comum, o pino a esquerda é chamado de normalmente aberto e a sua direita normal mente fechado, quando acionado o relé muda de um estado para o outro.

Existe também uma opção de verificar a temperatura e a humidade do ambiente, para obtenção dos dados foi utilizado os códigos contido conforme demonstrado no quadro 4, o mesmo faz a leitura do sensor e envia os dados para o servidor através do protocolo MQTT via Publish.

Quadro 4 – Trecho de código que obtém a temperatura e a humidade

```

//Leitura de Humidade
float h = dht.readHumidity();

// Leitura da Temperatura como Celsius
float t = dht.readTemperature();

// Verifique se alguma leitura falhou (para tentar novamente).
if (isnan(h) || isnan(t)) {
  Serial.println("Falha ao ler o sensor DHT11!");
  return;
}

// Calcula o índice de calor em Celsius
float hic = dht.computeHeatIndex(t, h, false);

String
payload = "";
payload += hic;
payload += " ";
String
payload2 = "";
payload2 += h;
payload2 += " ";

if (MQTT.connected()){
  Serial.print("Envia o payload: ");
  Serial.println("Temperatura: " + payload);
  Serial.println("Humidade: " + payload2);

//Realizara o Publish
if (MQTT.publish(topic, (char*) payload.c_str())) {
  Serial.println("Publish Temperatura ok");
}
else {
  Serial.println("Publish Falhou");
}
if (MQTT.publish(topic2, (char*) payload2.c_str())) {
  Serial.println("Publish Humidade ok");
}
else {
  Serial.println("Publish Falhou");
}
}
}

```

Fonte: Elaborado pelo autor (2017).

4 CONSIDERAÇÕES FINAIS

O propósito principal deste trabalho foi de apresentar um sistema de automação residencial de baixo custo, no decorrer do desenvolvimento foram levantadas todas as informações necessárias para entendimento e a implantação automação residencial. Todos os objetivos foram alcançados com êxito.

Um dos pontos de maior relevância, foi o baixo custo de implementação, onde foi gasto para a implementação do projeto em média R\$ 370,00 onde podendo ser reduzido para aproximadamente R\$ 95,00 substituindo o Raspberry Pi por um servidor nas nuvens, onde existe servidos pagos e também gratuitos, realizado um teste como Cloud MQTT (<https://customer.cloudmqtt.com>), o mesmo apresentou uma performance muito boa, podendo ser acessado tanto interno como externo, outro ponto a se destacar no projeto foi o uso do ESP8266 NodeMcu, seu custo benefício baixo e com a performance muito boa, seu custo aproximado é de R\$ 55,00 no Brasil, pode-se encontrar em sites como Aliexpress (<https://pt.aliexpress.com>) ou Deal Extreme (<http://www.dx.com>) por aproximadamente US \$3,50 convertendo para o Real aproximadamente 11,51. O mesmo disponibiliza WiFi integrado, o que facilitou na implementação do projeto, onde não foi preciso passar cabo de rede até o modulo, por seu custo ser tão baixo, pode ser utilizado em cada tomada um modulo se comunicando via WiFi utilizado o protocolo MQTT para se comunicar com o servidor Raspberry Pi.

Por fim, pode-se considerar que o projeto, em seu objetivo inicial, de desenvolver um sistema de automação residencial de baixo custo foi atendido.

REFERÊNCIAS

BALDENGINEER, MQTT Tutorial for Raspberry Pi, Arduino, and ESP8266. Disponível em: <<https://www.baldengineer.com/mqtt-tutorial.html>>. Acesso em: 24 out. 2017.

FINEP. Disponível em: <<http://finep.gov.br>>. Acesso em: 24 ago. 2017.

MACIEL, Vinicius. A Raspberry Pi e o Linux. **Embarcados**, 2014. Disponível em: <<https://www.embarcados.com.br/raspberry-pi-e-o-linux/>>. Acesso em: 7 set. 2017.

OLIVEIRA, Sergio. **Internet das coisas com ESP8266, Arduino e Raspberry PI**. São Paulo: Novatec, 2017.

PLACA ESP8266 NodeMcu ESP-12. Disponível em: <<https://statics3.seeedstudio.com/seeed/img/2017-03/qluwTVU7FQlvaC8dZy6x2JaM.jpg>>. Acesso em: 19 jan. 2018.

PLACA Raspberry Pi 3. 2017. Disponível em: <<https://www.raspberrypi.org/app/uploads/2017/05/Raspberry-Pi-3-1-1619x1080.jpg>>. Acesso em: 19 jan. 2018.

PUBSUBCLIENT sample for ESP8266 Arduino. **GITHUB** Disponível em: <<https://gist.github.com/igrr/7f7e7973366fc01d6393> >. Acesso em: 26 out. 2017

RAMPIM, Renata. **Internet das Coisas sem mistérios**. São Paulo: Netpress Books, 2016.

RASPBERRY PI: controlando o mundo com gpio. Disponível em: <<https://drive.google.com/drive/folders/0B8jg1T6usCJzekx2MmQ5ZVBxdlk>>. Acesso em: 22 ago. 2017.

REVISTA ELETRÔNICA BRIDGE ÁUDIO. Disponível em: <https://bridgeaudio.wordpress.com/2015/01/12/conceito-de-automacao-residencial-domotica>. Acesso em: 23 ago. 2017.

SENSOR de umidade e temperatura DHT11. Disponível em: <http://learning.grobotronics.com/images/Tutorials/DHT11_Pins.png>. Acesso em: 19 jan. 2018.

TECMUNDO. Disponível em: <<https://www.tecmundo.com.br/casas/9907-automacao-residencial-a-tecnologia-invade-a-sua-casa.htm>>. Acesso em: 23 ago. 2017.