



UNIVERSIDADE FEDERAL DE OURO PRETO
ESCOLA DE MINAS
COLEGIADO DO CURSO DE ENGENHARIA DE
CONTROLE E AUTOMAÇÃO



GUILHERME RAFAEL PEREIRA DA SILVA

IMPLANTAÇÃO DE UM SISTEMA DE AUTOMAÇÃO PREDIAL UTILIZANDO
PLATAFORMA EM NUVEM

MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Ouro Preto, 2018

GUILHERME RAFAEL PEREIRA DA SILVA

**IMPLANTAÇÃO DE UM SISTEMA DE AUTOMAÇÃO PREDIAL UTILIZANDO
PLATAFORMA EM NUVEM**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para obtenção de Grau em Engenheiro de Controle e Automação.

Orientador: Prof. M.Sc. Diógenes Viegas Mendes Ferreira

Coorientador: Prof. Dr. Luiz Fernando Rísoli Alves

Ouro Preto, 2018
Escola de Minas – UFOP
Novembro/ 2018

S586i

Silva, Guilherme Rafael Pereira da.

Implantação de um Sistema de Automação Predial Utilizando Plataforma em Nuvem [manuscrito] / Guilherme Rafael Pereira da Silva. - 2018.

71f.: il.: color; grafs; tabs.

Orientador: Prof. MSc. Diógenes Viegas Mendes Ferreira.

Coorientador: Prof. Dr. Luiz Fernando Rísoli Alves.

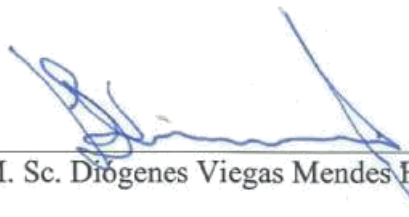
Monografia (Graduação). Universidade Federal de Ouro Preto. Escola de Minas. Departamento de Engenharia de Controle e Automação e Técnicas Fundamentais.

1. Internet da Coisas. 2. Automação Residencial. 3. MQTT. 4. ESP32. I. Ferreira, Diógenes Viegas Mendes. II. Alves, Luiz Fernando Rísoli. III. Universidade Federal de Ouro Preto. IV. Título.

CDU: 681.5

Catálogo: ficha.sisbin@ufop.edu.br

Monografia defendida e aprovada, em 28 de novembro de 2018, pela comissão avaliadora constituída pelos professores:



Prof. M. Sc. Diógenes Viegas Mendes Ferreira – Orientador



Prof. Dr. Luiz Fernando Rispoli Alves – Co-Orientador



Profa. Dra. Karla Boaventura Pimenta Palmieri – Professora Convidada



Prof. Dr. Paulo Marcos de Barros Monteiro – Professor Convidado

AGRADECIMENTO

Agradeço a Deus e a duas grandes senhoras, Esmenaide que durante todo seu tempo conosco buscou o melhor para seus familiares, e Lucas que sempre reforça suas orações para o futuro de seus filhos e netos.

Agradeço a minha família que ajudaram de forma imensurável, agradeço a força dos meus pais por terem confiado em um jovem, sair de casa, e buscar o mundo com meros 14 anos. Agradeço a família Braga que se formou, após minha demandada de Barbacena. E que tive o prazer de “apadrinhar” como minha segunda casa, o acolhimento sempre foi imenso, responsáveis momentos de felicidade, ponto de encontro de todas as famílias e todos os amigos. Agradeço de maneira especial ao meu irmão Marcel, que do início dessa aventura se apresentou como o maior parceiro para enfrentar esse mundo repleto de desafios, da maneira que pode foi simplesmente inexplicável, essa conquista é sua, sem sua força nada seria possível.

Agradeço a Francieli, que iluminou minha jornada em Ouro Preto. Proporcionando os melhores momentos que eu poderia ter nessa terrinha. “O nosso santo realmente bateu...”. Agradeço a todos os moradores da grandiosa República Pureza, que conviveram comigo nos últimos 4 anos, compartilhando vitórias e nos momentos de dificuldade. A vocês sempre serei grato por todo tempo que convivemos e que vamos conviver.

“... Pois eu sou da vila dos tigres, e a Pureza eu herdei...”.

Agradeço a todo o time Gerdau, pelos ensinamentos e pelo convívio. De maneira especial ao Robertinho, que me mostrou que as pequenas coisas podem sim, gerar grandes impactos. Aos amigos do Carro, vocês fizeram períodos de incerteza se transformarem em momentos inesquecíveis. “... Ninguém pode ter sucesso, se não começar arrumando sua própria cama ...”

Agradeço aos professores e aos grandes amigos feitos durante esse tempo na gloriosa Escola de Minas, que ensinaram e ajudaram a provar a filosofia que o com “Mente e a persistência”, nossos sonhos não tem limites. Ao fim dessa fase, não poderia deixar de agradecer as pessoas que foram de suma importância para conquista dessa vitória, dentre eles todos os moradores, e ex-moradores da República Toca do Bacanal, que acolheram um menino e formaram um homem. Aos amigos e professores da ETE, meu eterno respeito. Para alcançar as vitórias, devemos começar por uma base extremamente sólida. Com toda certeza digo e repito, vocês fizeram parte dos melhores anos da minha vida. Longe de casa, todos vocês foram minha família. Se essa vitória chegou, devo a todos vocês.

RESUMO

Desde os tempos primórdios o homem vem buscando maneiras de tornar suas atividades mais eficientes, gerando maior resultado, da maneira mais ágil e com o menor custo possível seja ele energético ou financeiro. A automação residencial vem ganhando espaço na vida das pessoas, com objetivo de automatizar as atividades nas residências, fornecendo aos seus usuários a capacidade de: monitorar, analisar, atuar e interagir com seus dispositivos, sendo o crescimento da área devido à implantação de sistemas inteligentes que usam tecnologias advindas do conceito da Internet das Coisas. Tecnologia essa que permite que os dispositivos estabeleçam troca de informações que possibilitem melhora em suas funções, acarretando em melhorias nas vidas de seus usuários. Diante disso, esse trabalho vem unir à Internet das Coisas, campo em constante crescimento e com boas perspectivas de mercado a automação predial. O presente trabalho foi desenvolvido com o objetivo de controlar lâmpadas e eletrodomésticos, além de monitorar dados como: temperatura e umidade de maneira remota através de um sistema em nuvem utilizando o protocolo de comunicação de máquinas-para-máquinas chamado MQTT realizando uma interação com o Kit de desenvolvimento eletrônico ESP32. Ao final do período de desenvolvimento, foi possível concretizar um protótipo funcional, além de documentação para futuras pesquisas e projetos a serem realizados na área automação residencial e internet das coisas.

Palavras chave: ESP32, automação, automação residencial, MQTT, IOT.

ABSTRACT

Since primordial times, man has been looking for ways to make his activities more efficient, generating greater results, in the most agile way and with the lowest possible cost, be it energetic or financial. Residential automation is gaining space in people's lives, aiming to automate activities at homes and provide users with the ability to: monitor, analyze, act and interact with their devices. The constant growth of the area is due to the deployment of intelligent systems, which uses technologies derived from the IoT (Internet of Things) concept. This technology allows the devices to establish connections with each other, making it possible to exchange information, which improve its functions and consequently the life of their users. Therefore, this work adds to the IoT, a field that is constantly growing and has great market perspectives regarding the domotic automation field. The presente work was developed aiming the control of lamps and electronic housing utensils, besides monitoring data such as: temperature and humidity in a remote way through a cloud computing system using the machine-machine protocol MQTT along with the electronic development kit ESP32. At the end of the development, it was possible to achieve a functional prototype, besides the documentation to future researches and projects on home automation and IoT.

Keywords: ESP32, automation, home automation, MQTT, IOT.

LISTA DE ABREVIATURAS

AC	<i>Alternating Current</i> (Corrente Alternada)
ADC	<i>Analog-to-Digital Converter</i> (Conversor Analógico para Digital)
CI	Circuito Integrado
CLP	Controlador Lógico Programável
DC	<i>Direct Current</i> (Corrente Contínua)
ES	<i>Embedded System</i> (Sistema Embarcado)
GPIO	<i>General Purpose Input/Output</i> (Portas programáveis de entrada e saída de dados)
IDE	<i>Integrated Development Environment</i> (Ambiente de Desenvolvimento Integrado)
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IOT	<i>Internet of Things</i> (Internet das Coisas)
MQTT	<i>Message Queue Telemetry Transport</i>
M2M	<i>Machine-to-Machine</i> (Máquinas para Máquinas)
PDC	Painéis de Comando Elétricos a Relés
VCA	Tensão em Corrente Alternada
VCC	Tensão em Corrente Contínua

LISTA DE FIGURAS

Figura 1 - KIT ESP32	16
Figura 2 - IDE Arduino	18
Figura 3 - Diagrama Esquemático de um Relé	19
Figura 4 - Módulo Relé	19
Figura 5 - Modelo de publicações e assinaturas p/ MQTT.....	21
Figura 6 - MQTT Dash	24
Figura 7 - MQTT Dashboard.....	25
Figura 8 - Usuários Finais e Aplicações.....	27
Figura 9 - Estrutura do IOT	28
Figura 10 - Crescimento do Mercado IOT	29
Figura 11 - Divisão do Mercado IOT.....	29
Figura 12 - Parcela dos Países	30
Figura 13 - Esquema de Pinos KIT ESP32	34
Figura 14 - Modelo Ilustrativo do Sistema.....	35
Figura 15 - Configuração Inicial.....	36
Figura 16 - Loop	37
Figura 17 - Função de Retorno	38
Figura 18 - Esquema de Ligação do Protótipo	40
Figura 19 - Tela Inicial CloudMQTT.....	41
Figura 20 - Tela Plan MQTT.....	41
Figura 21 - Definição do Data Center	42
Figura 22 - Informações da Instância.....	42
Figura 23 - Tela Inicial MQTT Dash.....	43
Figura 24 - Configuração do Cliente.....	44
Figura 25 - Menu Inicial	45
Figura 26 - Inserção de Elementos de Gatilho.....	46
Figura 27 - Criação de Elemento Switch.....	47
Figura 28 – Pré-Montagem.....	48
Figura 29 - Pré-Montagem 2	48
Figura 30 - Montagem Final.....	49
Figura 31 - Arquitetura com Banco de Dados	52

LISTA DE TABELAS

Tabela 1 - Características Principais ESP32.....	16
Tabela 2- Exemplo de Serviços de Nuvem	22
Tabela 3 - Pacotes <i>CloudMQTT</i>	23
Tabela 4 - Comparativo de Kits.....	33
Tabela 5 - Funções Firmware	34

Sumário

1	INTRODUÇÃO	11
1.1	Motivação	12
1.2	Objetivos	13
1.2.1	Objetivos Gerais	13
1.2.2	Objetivos Específicos	13
1.3	Estrutura	13
2	REVISÃO BIBLIOGRÁFICA.....	14
2.1	Automação	14
2.2	Sistemas Embarcados.....	14
2.2.1	Microcontroladores	15
2.2.2	Microprocessadores.....	15
2.3	ESP32.....	15
2.3.1	Periféricos Avançados	16
2.4	Arduíno.....	17
2.5	Relé	18
2.6	MQTT.....	19
2.6.1	O que é?	19
2.6.2	Funcionamento	20
2.6.3	Segurança	21
2.6.4	Broker.....	22
2.6.4.1	CloudMQTT	22
2.7	Aplicativo Cliente MQTT.....	23
2.8	Internet das Coisas	25
2.9	Automação Residencial	30
2.9.1	Trabalhos Relacionados	31
3	ESTUDO DE CASO	33
3.1	Firmware	35
3.2	Esquema Elétrico do Projeto.....	39

3.3	Configuração da Instância do <i>Broker</i>	41
3.4	Aplicativo	43
3.5	Protótipo Final e Testes	48
4	RESULTADOS.....	51
5	CONSIDERAÇÕES FINAIS	52
5.1	Sugestões para Trabalhos Futuros.....	52
	REFERÊNCIAS BIBLIOGRÁFICAS	54
	APÊNDICES	58
	Código Utilizado no Arduino	58

1 INTRODUÇÃO

Não é fácil compreender o momento de descoberta da Automação Industrial. Teoricamente, para ela existir, deve-se ter uma indústria com processos cabíveis de serem controlados e automatizados. Assim, é possível marcar o início da automação por volta do século XVIII na Inglaterra, de forma simultânea ao desenvolvimento da máquina a vapor. Já no século XIX, com o descobrimento de novas fontes energéticas como petróleo e energia elétrica, a indústria tem grande crescimento, principalmente nos EUA e Europa. Em meio ao processo de expansão industrial, os relés são criados. Esse período de descobertas é conhecido como a II Revolução Industrial (SILEVIRA; LIMA, 2003).

No final do século XX, com a indústria já instalada, novas demandas são criadas como: produção em massa, linhas de montagem, etc. Para essa finalidade os relés já enfrentavam problemas devido a necessidade de grandes áreas físicas para instalação de um painel de controle (PDC). Tais painéis possuíam alta complexidade de manutenção e desenvolvimento. Com isso em 1968 foi criado pela *Bedford Associates* o primeiro controlador lógico programável (CLP), conhecido por MODICON. Tal dispositivo permitiu que as indústrias colocassem maior complexidade e também permitiu que fosse obtido melhores resultados em seus processos (SCHNEIDER ELETRIC, 2018).

O campo do CLP foi aberto pela indústria automobilística, e conseqüentemente, todas as outras indústrias aderiram ao dispositivo. Grande parte do seu crescimento foi alavancada pela facilidade de programação, em linguagens que se aproximavam das antigas instalações utilizadas pelos eletricitistas. Anos mais tarde, tais elementos tornaram-se mais complexos, dotados de maior capacidade de processamento e com novos recursos, aumentando assim o conjunto de possibilidades para implantação desses dispositivos na indústria (LIMA, 2003).

Com a consolidação da automação aplicada a meios fabris, os engenheiros e pesquisadores começaram a buscar soluções que pudessem ser aplicadas em outros segmentos como: predial, rural, etc. Diferentemente das indústrias, tais mercados possuíam demandas específicas, como: flexibilidade de utilização, capacidade de interação, tamanho físico, consumo energético e custo financeiro. Para suprir essas necessidades, ampliou-se a utilização de sistemas embarcados para soluções de automação, tecnologia essa que já se mostrava capaz de suprir as novas demandas (TEZA, 2002).

Dentro dos sistemas embarcados, uma esfera que vem ganhando espaço é o do desenvolvimento de ambientes inteligentes, pela utilização de dispositivos que tenham tecnologias vindas da internet das coisas (do inglês, *Internet of Things* - IOT). Segundo Mandula et al. (2015) o objetivo principal desta tecnologia é gerenciar e controlar os dispositivos que estão ao nosso redor, gerando assim maior eficiência, saúde, segurança e conforto em nossas atividades diárias.

Segundo Columbus (2017), espera-se um crescimento no mercado global de IOT que acarretará em uma movimentação de 457 bilhões de dólares em 2020, com 75 bilhões de dispositivos conectados à internet em 2025. Diante das expectativas de crescimento do mercado, e da gama de opções de aplicação das tecnologias emergentes junto ao IOT, nesse trabalho será apresentado o desenvolvimento de um sistema de automação aplicado ao controle de dispositivos elétricos, e o monitoramento de dados, temperatura e campo-magnético através de uma placa ESP32, conectada à internet, sem fio por meio do protocolo *Message Queue Telemetry Transport* (MQTT), permitindo a interação do usuário via aplicativo instalado em um *smartphone* com sistema operacional Android.

1.1 Motivação

Já é reconhecido e difundido o conceito de casas inteligentes (do inglês, *Smart Homes*), que de maneira expressiva vem mudando a vida e o cotidiano das pessoas, permitindo um maior controle e maior conexão do ambiente residencial com seus usuários (YANG; LEE W.; LEE H., 2018).

As tecnologias aplicadas em residências, são cada vez mais presentes na vida das pessoas. Devido a propagação dos dispositivos inteligentes (do inglês, *Smart Devices*), dotados da capacidade de monitorar, controlar e processar dados. Soluções antes aplicáveis apenas em indústrias agora são levadas a outros ambientes, devido a evoluções tecnológicas como: miniaturização de sensores, aumento da capacidade de processamento e utilização de computação em nuvem.

A utilização de soluções oriundas, da união das tecnologias apresentadas, resulta na área conhecida como IOT, área essa que vem como ferramenta para facilitar a vida dos usuários, não apenas em indústrias e residências, como também nas áreas de segurança, saúde, vendas, marketing etc.

As possibilidades de atuação para os ingressantes na área de automação sejam elas em mercados já consolidadas como o residencial, ou em novos negócios, justificam o desenvolvimento do presente trabalho.

1.2 Objetivos

1.2.1 Objetivos Gerais

Desenvolver um sistema de automação para controle de dispositivos em residências, utilizando plataforma na nuvem.

1.2.2 Objetivos Específicos

- Fazer uma revisão de literatura sobre trabalhos já desenvolvidos na área de automação predial;
- Fazer uma revisão de literatura sobre a utilização de soluções IOT;
- Fazer uma revisão de literatura sobre o protocolo aplicado;
- Desenvolver um sistema automatizado para controle de cargas que utilizem tensão alternada, via internet;
- Fornecer documentação para o Laboratório de Automação Predial da Universidade Federal de Ouro Preto - UFOP, para futuras pesquisas dos alunos sobre aplicações IOT;
- Fornecer material para projetos futuros que utilizem tecnologias como: MQTT, IOT e ESP32;

1.3 Estrutura

Este trabalho está dividido em cinco capítulos, já incluindo esta introdução, onde é apresentado um breve contexto histórico dos rumos da automação, partindo do CLP até aplicações IOT. No Capítulo 2 é apresentada uma revisão bibliográfica das ferramentas que foram utilizadas no trabalho. No capítulo 3 é apresentada a forma com que essas ferramentas foram aplicadas no projeto, a metodologia de desenvolvimento do protótipo. No Capítulo 4 apresentam-se os resultados que foram alcançados com o protótipo. No último capítulo, apresentam-se as considerações finais e sugestões para pesquisas futuras.

2 REVISÃO BIBLIOGRÁFICA

2.1 Automação

CLPs e microcontroladores são largamente utilizados em projetos de automação (PALMIERE, 2016). Um principiante pode ter dúvidas sobre qual utilizar em seus projetos, tal resposta depende de suas demandas. No presente trabalho aborda-se a implementação de um projeto voltado a automação predial.

Para o mesmo, optou-se em utilizar um microcontrolador, pois embora o CLP possua grande modularidade e facilidade de programação, o mesmo possui um alto custo financeiro, fato esse que restringe sua utilização (PALMIERE, 2016).

2.2 Sistemas Embarcados

É possível encontrar sistemas embarcados em quase todos os dispositivos eletrônicos existentes, suas aplicações abrangem diversas áreas, de eletrodomésticos, carros, equipamentos hospitalares e até mesmo nossas roupas (RODRIGUES; PEDRO; TEDESCO, 2013).

A designação de sistema embarcado, pode ser compreendida como um computador voltado para uma determinada aplicação ou aplicações específicas, ao contrário dos computadores de uso geral como *Desktops* e *Notebooks*. Essa definição abrange desde processadores de 8-bits até processadores 64-bits com múltiplos núcleos. Dados históricos apontam o AGC (*Apollo Guidance Computer*), como o primeiro sistema embarcado (do inglês, *embedded system* – ES) desenvolvido, tal sistema estava a cargo de controlar a espaçonave Apollo (EMBARCADOS, 2013).

Segundo Mehi (2011), um sistema eletrônico pode ser considerado um sistema embarcado quando tem por finalidade uma determinada tarefa, e por causa disso interage com o mundo externo de forma contínua, através de sensores, atuadores, e *softwares*. Tais sistemas também possuem características particulares que os diferenciam dos computadores de uso geral, como: memória, capacidade de processamento, periféricos, entradas e saídas disponíveis para acesso e interação com o mundo externo.

É comum encontrar estudantes ou até mesmo profissionais da área elétrica / eletrônica que não sabem definir com certeza onde enquadrar os Microcontroladores e Microprocessadores, e distinguir suas diferenças e finalidades.

2.2.1 Microcontroladores

Para Rodrigues, Pedro e Tedesco (2013) Microcontroladores podem ser designados como processadores que incorporam “n” funções agrupadas em um único circuito integrado (CI). Diferentemente dos Microprocessadores que por serem aplicados em computadores de uso geral, possuem maior capacidade de processamento.

2.2.2 Microprocessadores

Podem-se distinguir os Microprocessadores dos outros sistemas embarcados, pois independente da grande capacidade de processamento, esses sistemas não possuem estrutura completa para formação de um sistema mínimo, necessitam de periféricos externos como: memória e unidade de entrada e saída, apenas assim será possível constituir um sistema mínimo. Microcontroladores por outro lado possuem integrados no mesmo chip memória e periféricos para interação com o ambiente (NERYS, 2016).

2.3 ESP32

Circuito integrado combo, desenvolvido pela empresa *Expressif System*, empresa de origem chinesa, número um do mundo na venda de CIs para IOT (EXPRESSIF SYSTEMS, 2018). A ESP32 possui periféricos que auxiliam no desenvolvimento de soluções IOT como: *Wi-fi*, *Bluetooth Low Energy (BLE)* e sensor capacitivo já embutidos, com capacidade de trabalhar em modos de baixo consumo, fatos esses que ajudam no desenvolvimento realizado pelo usuário, evitando custo com outros componentes. Suas características a enquadram como um microcontrolador.

A placa utilizada, já apresenta eletrônica embarcada para programação. Dessa forma o único adereço necessário para conecta-la ao computador é um cabo mini – USB, que tem por finalidade alimentar a placa e também servir como meio físico para trocar informações.

Seu *firmware* é de desenvolvimento ágil, em grande parte devido a gama de ambientes de desenvolvimento integrado (do inglês, *integrated development environment* – IDE) compatíveis com a plataforma. Foi definido pela IDE do Arduíno, devido ao grande número de matérias disponíveis, e a grande comunidade de desenvolvedores que podem fornecer suporte durante a execução de um projeto.

Na tabela 1 e na figura 1 apresentam-se respectivamente as características e a imagem da placa.

Tabela 1 - Características Principais ESP32

Características Principais	
<i>Wi-fi:</i>	802.11 B/G/N - 2.4GHz
Modos de Operação:	<i>Client/ Access Point/ Station + Access Point</i>
Microprocessador:	Dual Core Xtensa 32-Bit LX6
<i>Clock:</i>	80 até 240 MHz
Tensão de Alimentação:	3.3 VDC
SRAM:	512 KB
ROM:	448 KB
<i>Flash:</i>	32 MB
IO máx:	12 mA
GPIO:	36 (PWM / I2C e SPI)
<i>Bluetooth:</i>	4.2 BLE

Fonte: KOYANAGI, 2017

Figura 1 - KIT ESP32



2.3.1 Periféricos Avançados

Dentre os recursos chamados de periféricos avançados disponíveis no kit da ESP32, optou-se por utilizar dois deles, sendo: sensor de temperatura e sensor de efeito Hall.

- Sensor de Temperatura: Um sinal de tensão é fornecido, segundo a temperatura verificada pelo sensor, que opera na faixa entre -40°C até 125°C. O valor fornecido é convertido para uma palavra digital, através de um conversor analógico para digital (do inglês, *analog-to-digital converter* – ADC). Os valores coletados pelo sensor, devido a fatores de construção e também a interferência de circuitos como

CI que opera o *Wi-fi*, este sensor é indicado para verificar mudanças de temperatura. Para outros tipos de aplicação ele deverá ser calibrado antes de ser utilizado;

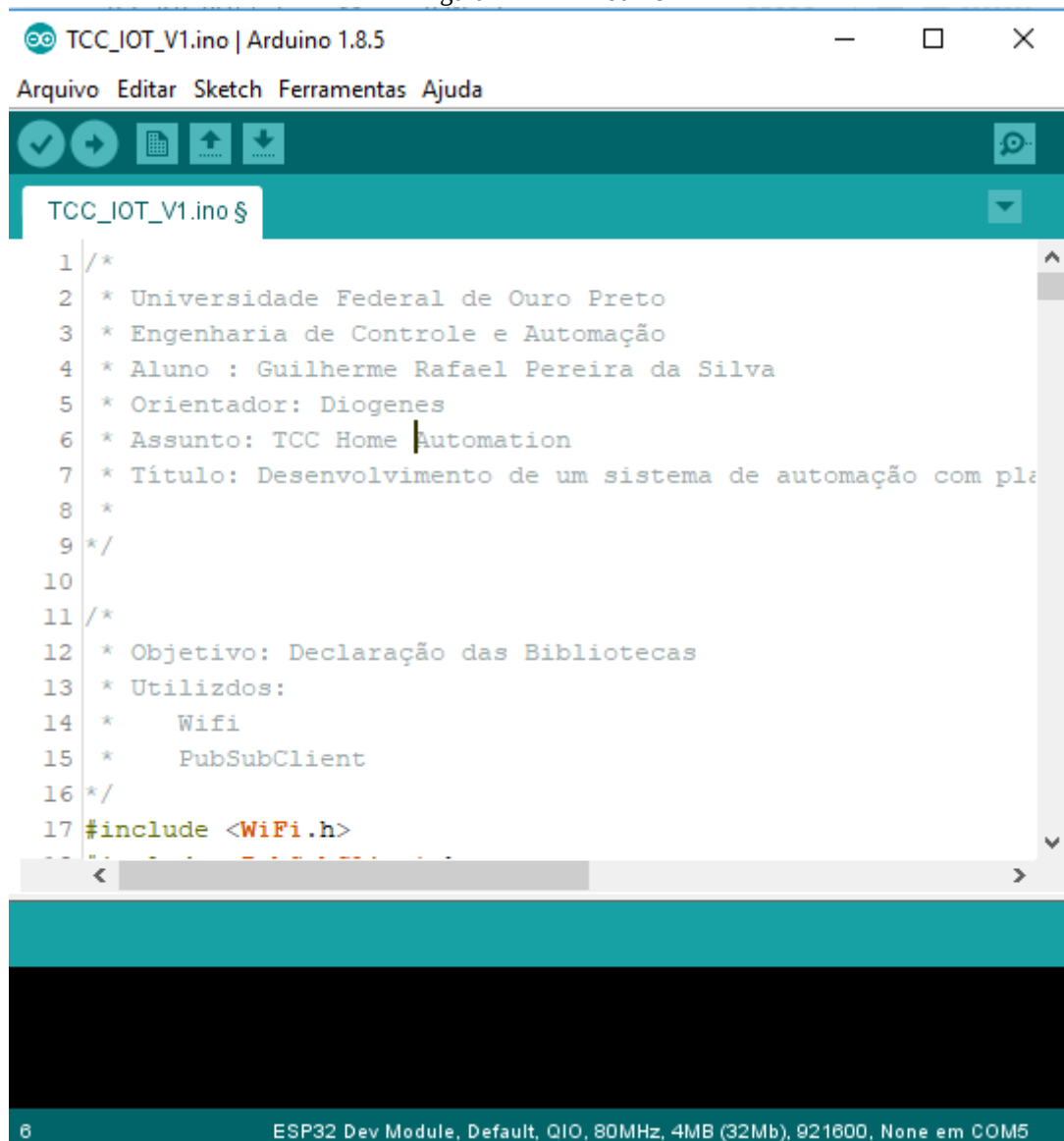
- Sensor de Efeito - Hall: O módulo integrado para medição de campo magnético é baseado no conceito do *N-carrier* resistor. A exposição do sensor a um campo magnético gera um nível de tensão sobre o resistor, que é convertido para uma palavra digital pelo ADC, antes de tal conversão o sinal pode passar por um pré-amplificador de baixo ruído para depois ser convertido (ESPRESSIF SYSTEMS, 2018). Na presente pesquisa, não foi possível encontrar a faixa de operação de tal sensor, mesmo assim optou-se em mantê-lo devido ao caráter acadêmico do experimento.

2.4 Arduíno

Plataforma livre criada com o objetivo de tornar mais eficiente o desenvolvimento de protótipos, sendo eles, *hardware e software*. Possui kits com capacidade de ler dados oriundos de botões, sensores e atuarem em saídas após processamento de dados. Para programar os kits dessa plataforma, o usuário pode utilizar a IDE nativa que possui linguagem própria. Destaca-se o grande número de material já existente para consulta, devido ao fato do ecossistema já ser largamente difundido. Ao passar dos anos os kits vem sofrendo mudanças para acompanhar as novas tendências tecnológicas (ARDUINO, 2018).

O projeto que é apresentado por esse documento utiliza a IDE do Arduíno para desenvolver o *software* de controle, a tela padrão do sistema é apresentada na figura 2.

Figura 2 - IDE Arduino

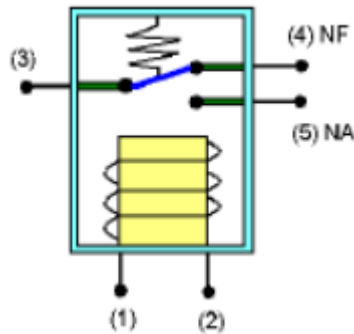


2.5 Relé

Relés são dispositivos eletromecânicos que tem por finalidade a operação de cargas em corrente alternada (do inglês, *alternating current* – AC), sua construção isola fisicamente os sinais de controle dos sinais de potência. (SOUZA,2009). No mercado existem variações desse dispositivo que o tornam mais inteligente, auxiliando dessa maneira seu usuário durante a implementação de sistemas com demandas mais elaboradas.

Na figura 3 é apresentado o esquemático de funcionamento de um relé semelhante ao utilizado no projeto.

Figura 3 - Diagrama Esquemático de um Relé



Fonte: SOUZA, 2009

O acionamento do contato móvel do relé representado pelo número (3) ocorrerá no momento em que for circulada corrente elétrica pela bobina, representado nos contatos (1) e (2). O controle da bobina é feito com sinais que podem variar em diferentes valores, o projeto utilizará um elemento controlado por um sinal de 5 VCC. Quando excitada, a bobina criará um campo magnético deslocando assim o contato (3), para um dos dois contatos auxiliares: (4) ou (5). Esses contatos podem acionar cargas de até 250 VAC/10A.

Deve-se ressaltar que não existe ligação física dos dois lados do dispositivo, fato esse que isola os sinais de controle dos sinais de potência. Na figura 4 é apresentada a imagem do kit relé utilizado.

Figura 4 - Módulo Relé



2.6 MQTT

2.6.1 O que é?

Protocolo de comunicação assíncrono, de máquinas-para-máquinas (do inglês, *machine-to-machine* - M2M), que opera sobre o protocolo TCP/IP, direcionado à aplicações IOT, baseado no sistema de troca de mensagens do tipo cliente servidor por publicações (do inglês, *publish*) e assinaturas (do inglês, *subscriber*), criado pelas empresas IBM e Eurotech em 1999. Largamente utilizado desde sua invenção, possui grande utilidade para realizar conexões entre dispositivos remotos, onde pequenos pacotes são necessários devido a limitação de banda disponível para tráfego de dados. O protocolo

pode ser empregado em aplicações de diversas áreas, que possuam demandas semelhantes as apresentadas nesse documento. Além dos pontos citados, tal protocolo beneficia aplicações onde se faz necessário encaminhar informações para múltiplos usuários (MQTT.ORG, 2018).

O protocolo possui como características:

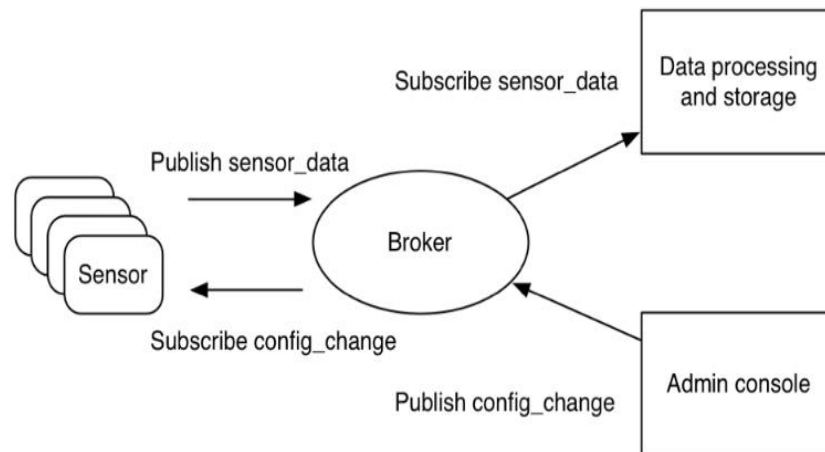
- Mensagens padronizadas no formato publicações e assinaturas, fato que auxilia na distribuição de um pacote para clientes múltiplos;
- Mecanismo de envio de mensagens para os dispositivos em caso de desconexão de maneira anormal;
- Baixa sobrecarga na transmissão, para reduzir o tráfego de rede mudanças no protocolo são minizadas (BANKS; GUPTA, 2014).

2.6.2 Funcionamento

O padrão de publicações e assinaturas utilizado nesse protocolo funciona da seguinte maneira: quando uma unidade da rede deseja receber algum dado, ela o subscreve, assim é feita uma requisição para outro elemento dessa mesma rede que tem por finalidade gerenciar as publicações e assinaturas do sistema. Na rede MQTT esse elemento é chamado de *broker*, que pode ser definido como um intermediador no processo de comunicação. Para elementos que necessitam realizar alguma publicação, o processo também é realizado mediante utilização do *broker*, no caso o dado não será solicitado e sim encaminhado para o intermediador. O padrão de publicações e assinaturas não é inédito, o mesmo já era utilizado em redes *Fieldbus* (BARROS, 2015). Na figura 5 é apresentado o modelo de funcionamento do protocolo utilizado no projeto.

Para distinguir as mensagens trocadas através desse protocolo, é utilizado o conceito de tópicos. Elementos presentes em uma mesma rede podem encaminhar diversos tópicos para o *broker* e, tais tópicos, podem ser assinados de maneira conjunta ou não por outros elementos da rede (BARROS, 2015).

Figura 5 - Modelo de publicações e assinaturas p/ MQTT



Fonte: YUAN, 2017

2.6.3 Segurança

Existem métodos para aumentar a segurança do sistema. A forma com que o cliente se conecta para subscrever ou publicar no *broker* é feita por via do TCP, pode-se criar camada de segurança por usuário e também é possível utilizar criptografias como SSL/TLS. Além desses métodos, é necessário que o usuário configure a qualidade do serviço desejado (QoS, de *Quality of Service*), a configuração serviço é feita entre cliente e o *broker*, com isso é possível existir níveis diferentes entre publicações e subscrições, segundo a demanda da aplicação (BARROS, 2015).

- Existem três tipos de qualidade que podem ser aplicados neste tipo de protocolo:
 - No máximo 1 (QoS 0): O elemento que envia a mensagem, não tem a obrigação de armazenar a mensagem para futuro reenvio. Mesmo assim, falhas podem ocorrer na transmissão. Esse nível de qualidade é indicado em sistemas onde a falha na recepção de uma informação não causa danos ao sistema;
 - No mínimo 1 (QoS 1): É certo que a mensagem será entregue pelo menos uma vez, dados duplicados podem ocorrer na transmissão. Nesse nível ocorre a verificação apenas de quem recebe a mensagem;
 - Exatamente 1 (QoS 2): O nível apresenta garantias de que a mensagem será entregue uma vez, a verificação é de envio e recebimento é feita de forma dupla.

Enquanto toda a operação não for confirmada a mensagem será mantida no sistema. Esse nível é indicado para operações onde a falha na transmissão de um dado possa causar danos ao sistema (BANKS; GUPTA, 2014; BARROS, 2015).

2.6.4 Broker

Existem diversos serviços de *broker* disponíveis no mercado, cada um possui suas especificidades de aplicação. Na tabela 2 é apresentado um resumo de dois exemplos de serviços.

Tabela 2- Exemplo de Serviços de Nuvem

Serviços	
Mosquitto	Pertencente a Fundação Eclipse, o <i>broker</i> que pode ser implementado com o protocolo MQTT das versões 3.1 e 3.1.1. Sua utilização vai de grandes servidores, até placas que necessitam operar com baixo consumo de energia (MOSQUITTO, 2018).
Adafruit IO	O Adafruit IO é um sistema que permite uma melhor análise de dados, a ferramenta é de ágil desenvolvimento e permite conexões sem a necessidade de programas complexos. O sistema tem a capacidade de tratar protocolos, dentre eles o MQTT e o REST (ADAFRUIT, 2015).

2.6.4.1 CloudMQTT

Devido as necessidades do projeto, foi optado pela utilização do serviço *CloudMQTT*, que possui serviços livres de pagamento e serviços pagos. A opção sobre um plano é feita segundo a demanda necessária.

Na tabela 3 é exibido um resumo com os planos disponíveis e suas características:

Tabela 3 - Pacotes *CloudMQTT*

Plano	Característica	Custo Mensal em Dólares
<i>Power Pug</i>	Até 10000 conexões, sem limitações artificiais, suporte por e-mail e telefone.	299
<i>Loud Leopard</i>	Até 1000 conexões, sem limitações artificiais, suporte por e-mail.	99
<i>Keen Koala</i>	Até 100 conexões, sem limitações artificiais, suporte por e-mail.	19
<i>Humble Hedgehog</i>	Até 25 conexões, limite de tráfego em 20Kbit/s, 3 bridges e suporte por e-mail.	5
<i>Cut Cat</i>	Até 5 conexões com o limite de tráfego de 10Kbit/s.	Grátis

Fonte: *CloudMQTT*, 2018

Devido a demanda do projeto, gerado por um caráter de demonstração acadêmica. Optou-se pelo pacote *Cut Cat* que é isento de pagamento. Outros serviços de *Broker* como o Mosquitto são totalmente livres de pagamento.

Os dados da instância criada para o projeto estão alocados em um centro de dados (do inglês, *datacenter*) da multinacional de tecnologia *Amazon*, situado no estado da Virginia do Norte nos Estados Unidos. Durante o processo de criação de uma instância, é possível definir o local onde serão alocadas as informações, a disponibilidade de locais é diferente segundo o plano selecionado pelo usuário.

2.7 Aplicativo Cliente MQTT

Em caráter experimental foi definido pela utilização de dois aplicativos diferentes para controle do sistema através do *smartphone*, devido a disponibilidade de aparelho, os dois aplicativos apresentados pertencem a plataforma Android. Mas como já apresentado, o sistema pode ser utilizado em softwares que funcionem sobre outros sistemas operacionais como o IOS. As telas dos dois aplicativos são apresentadas nas figuras 6 e 7.

Figura 6 - MQTT Dash

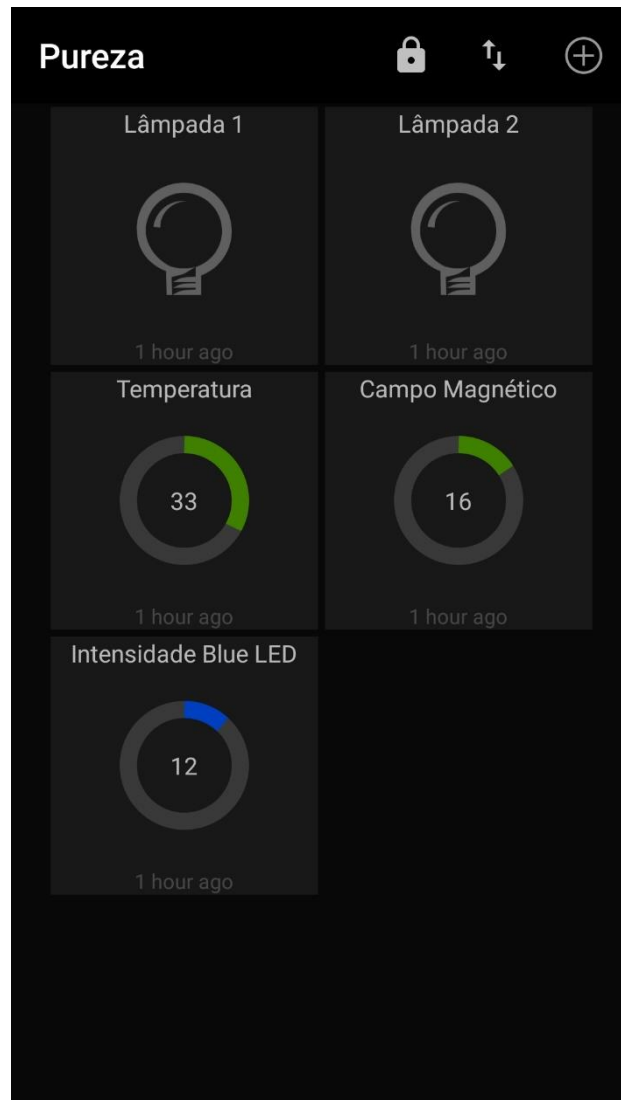
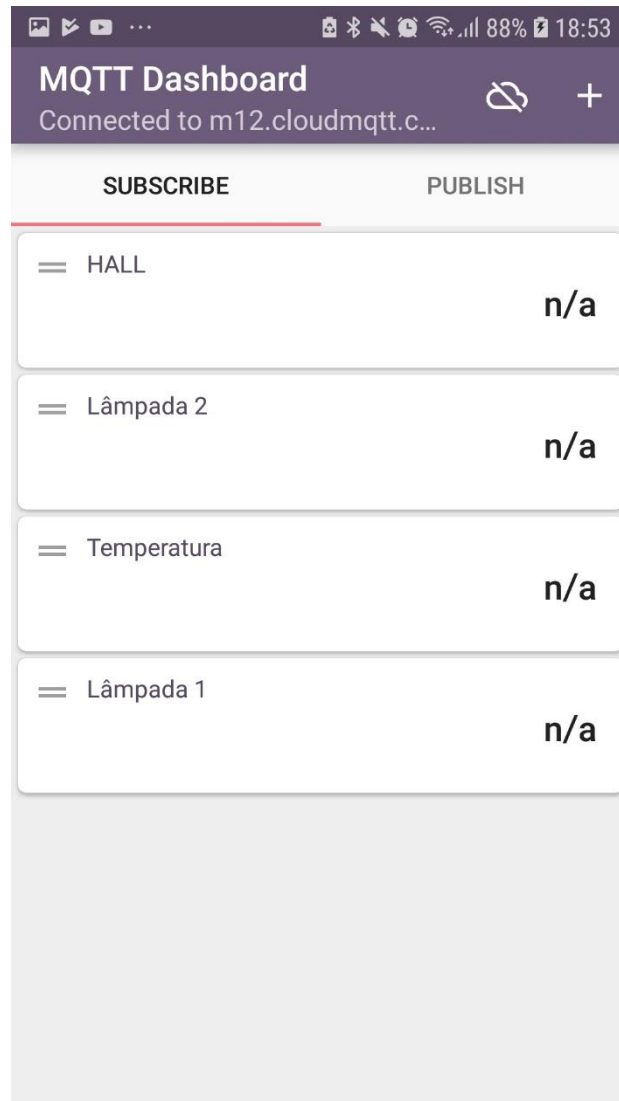


Figura 7 - MQTT Dashboard



2.8 Internet das Coisas

Com o avanço da tecnologia, a maneira como o ser humano interage com o mundo irá mudar, a relação com o mundo *online* não será restringida apenas por *desktops*, *notebooks*, *smartphones* etc. O desenvolvimento do IOT mostra que a interação do homem, será com todos os dispositivos que o rodeiam, sejam eles em casa ou no trabalho (GUBBI et al., 2013).

E comum encontrar dificuldade entre leigos e pessoas da área, quando questionadas sobre o que é IOT. Esta dificuldade advém da complexidade do sistema, o rápido avanço e o grande número

de possibilidades que essa tecnologia abrange aliado à ausência de definição completa por órgãos de padronização (SUMMARY, 2015).

Minerva, Biru e Rotondi (2015, p 10.) apresentam o primórdio do IOT, sendo criado por Sarma e Brock. Na época a identificação por rádio frequência (do inglês, *radio frequency identification tag* - RFID) era utilizada para rastrear e cargas em meio ao transporte, principalmente em meio ao mercado de empresas logísticas ou de suprimentos. Sarma e Brock foram os primeiros a colocar as informações dos produtos na internet através de *link* com as TAG RFID, tal feito foi de extrema importância para as novas operações das indústrias.

Além de empresas logísticas, mercados como: pagamentos, controle animal, pedágios, etc. Também utilizaram tecnologias sem fio semelhantes ao RFID como o da comunicação por proximidade de campo (do inglês, *Near Field Communication* – NFC). O crescimento destas tecnologias foi alavancado em grande parte pela definição de protocolos, padrões e barateamento da implantação (MINERVA; BIRU; ROTONDI, 2015; VIEGAS, 2017).

Segundo Viegas (2017), não existe um censo comum da primeira vez que foi utilizado o termo Internet das Coisas, sabe-se apenas que ela ocorreu na segunda metade da década de 90. O conceito da tecnologia surgiu devido a necessidade de interação de dados coletados por sensores e atuadores presentes em nosso dia-a-dia, realizando monitoramento de forma autônoma. Fornecendo ao ser humano informações de valor, para tomada de decisões em momentos específicos com maior eficiência possível.

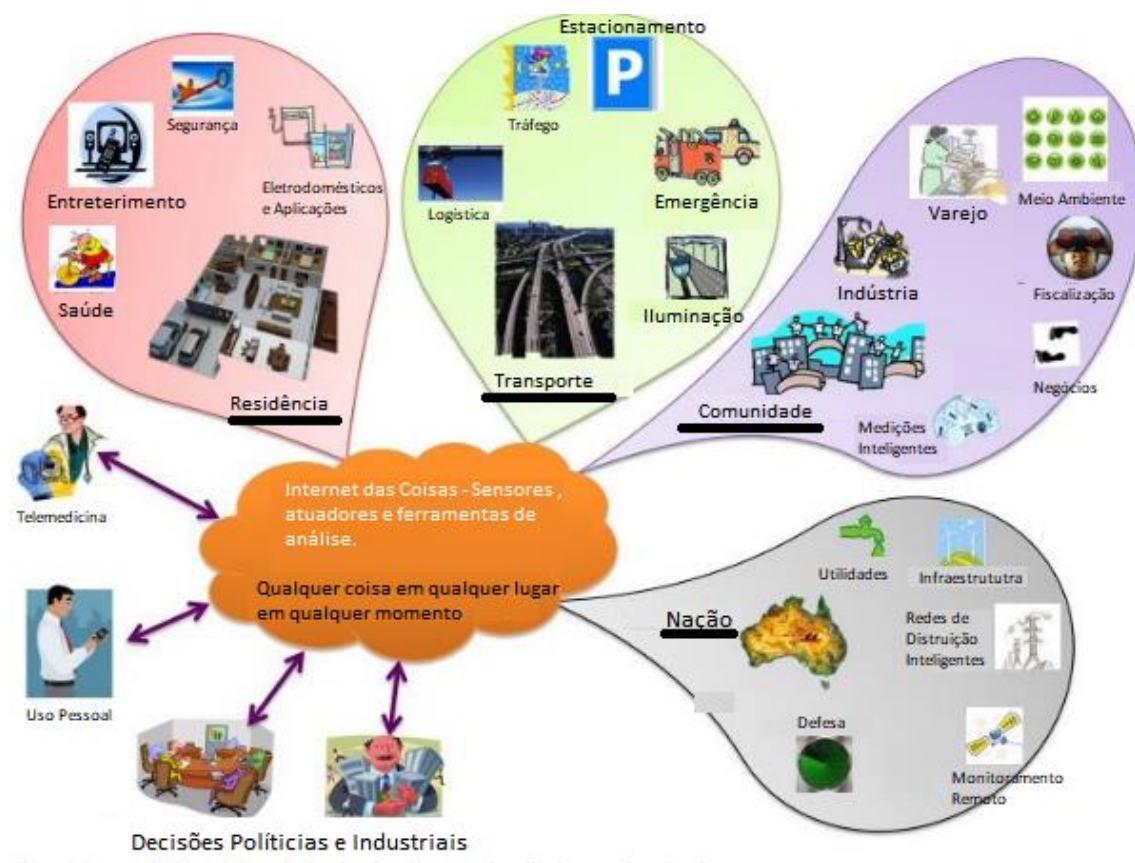
Com a crescente de dispositivos popularmente conhecidos como inteligentes (do inglês, *smart*), embarcados com Microcontroladores cada vez mais potentes. Tem - se uma mudança de objetivo, os *smart devices* tendem a ganhar capacidade não só de encaminhar dados para a internet, como obtém a capacidade de tomar decisões segundo as informações coletadas sem a necessidade de interação do homem. Caso necessário eles próprios eles se conectam a outros dispositivos para descobrir mais dados sobre o meio em que estiverem aplicados (VIEGAS, 2017). Com essas possibilidades aparecem protocolos de comunicação como; máquinas-para-máquinas, humanos-para-humanos e máquinas-para-humanos. O projeto apresentado por esse documento utiliza um protocolo de máquinas-para-máquinas.

Pereira (2017) apresenta a ideia de IOT de maneira diferente, ele a descreve como uma derivação de conceitos oriundos da computação ubíqua, como: coisas que pensam (do inglês, *things that think*) e ambientes inteligentes (do inglês, *ambient intelligence*) etc.

Para que ocorra um real crescimento do IOT, é necessário que os padrões da computação como mude, de maneira que todos os dispositivos possam se conectar, com computadores, *smartphones* etc. Tornando assim o ambiente em um ecossistema inteligente (GUBBI et al., 2013).

Na figura 8 é apresentado, um exemplo de sistema inteligente possível a partir da aplicação do IOT.

Figura 8 - Usuários Finais e Aplicações



Fonte: Adaptado de GUBBI et al., 2013

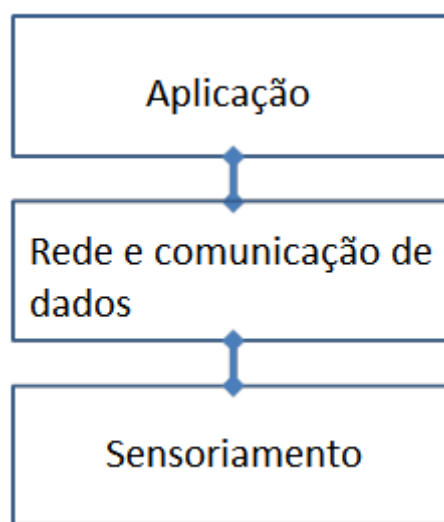
Os campos de aplicação apresentados na figura 8 podem ser traduzidos pelas palavras de Pereira (2017) “A Internet das Coisas permite que pessoas e coisas possam se conectar a qualquer hora, em qualquer lugar, com qualquer coisa, utilizando qualquer caminho, rede ou serviço”.

Pesquisas realizadas pelo Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE) identificaram mais de 140 padrões e projetos relevantes para o desenvolvimento de um padrão universal para a tecnologia do IOT (MINERVA; BIRU; ROTONDI, 2015).

Para buscar um padrão global no desenvolvimento dessa tecnologia a IEEE, criou em 2014 uma iniciativa para unir esforços entre indústria, governos e academia na busca da criação das bases do IOT, baseando-se em três pilares: mercado, tecnologia e padrões (IEEE, 2015).

O projeto P2413, criado pela IEE com o objetivo de gerar um padrão para o IOT, descreve a estrutura do IOT em três camadas, conforme pode ser observado na figura 9 (IEEE, 2015):

Figura 9 - Estrutura do IOT



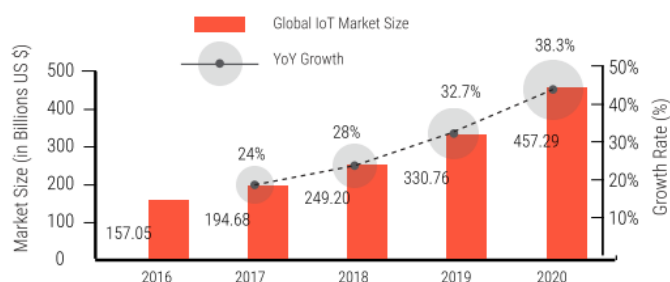
Fonte: Adaptado do MINERVA, BIRU E ROTONDI, 2015

Minerva, Biru e Rotondi (2015, p 10.) descrevem o IOT como: “uma rede de itens com sensores embarcados, que são conectados à internet”. Já o relatório apresentado pelo Instituto de Engenheiros Eletricistas e Eletrônicos (2015) retrata de maneira mais profunda como uma rede de sensores, atuadores e objetos inteligentes, sendo eles da indústria ou de nosso cotidiano, e da mesma maneira que Gubbi (2015) cita que o grande volume de dados gerados por esses elementos terão que ser processados e apresentados aos usuários da maneira mais clara possível, assim tornando mais eficiente as atividades executadas. Viegas (2017) apresenta que existem diferentes definições para o IOT, entre os órgãos de padronização do mundo.

O possível impacto dessa nova tecnologia pode ser verificado segundo as palavras do ex-diretor executivo da Samsung BK. Yoon “A internet das coisas tem o potencial de transformar nossa sociedade, economia e a forma que nós vivemos” (GROWTHENABLER, 2017).

Segundo dados do relatório Growthenabler (2017), existem boas expectativas para o crescimento do número de dispositivos e da movimentação financeira entorno desse mercado, conforme apresentado na figura 10.

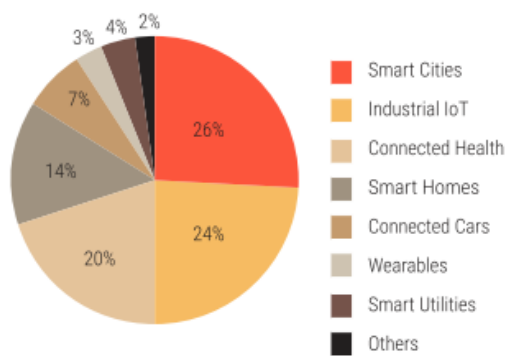
Figura 10 - Crescimento do Mercado IOT



Fonte: GROWTHENABLER, 2017

Na figura 11, podem-se verificar os três pilares do crescimento do IOT, sendo: cidades inteligentes, indústria e tecnologias médicas.

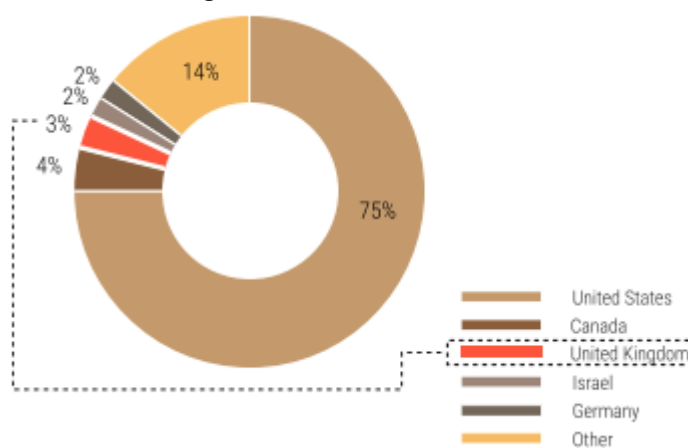
Figura 11 - Divisão do Mercado IOT



Fonte: GROWTHENABLER, 2017

Segundo o gráfico do ranking dos países que possuem maior desenvolvimento no segmento de IOT, verifica-se a predominância dos países de primeiro mundo, que já possuem infraestrutura tecnológica operante para suportar esses tipos de soluções. Como fato de grande relevância, destaca-se, a ausência de potências como China e Índia, tais informações podem ser vistas na figura 12 (GROWTHENABLER,2017).

Figura 12 - Parcela dos Países



Fonte: GROWTHENABLER, 2017

2.9 Automação Residencial

Automação Residencial, conhecido também por domótica pode ser apresentada como a porta de entrada para inserção de tecnologias em uma residência, prédio, comércio etc. De maneira que essa tecnologia possa fornecer, maior segurança, comodidade, conforto com o menor custo possível para seu usuário (PEREIRA, 2017).

Mantenha a temperatura de seu ambiente agradável durante todo o ano, partindo de informações como temperatura externa do imóvel, número de pessoas presentes, períodos do dia. Ilumine seu jardim durante a noite, monitore os parâmetros da água de sua piscina etc. A automação predial surge como forma de integrar todos os sistemas de forma a proporcionar uma vida mais confortável, segura, aos seus usuários usando de forma mais eficiente os recursos energéticos, acarretando no menor custo possível (LEVITON, 2013).

Além de fatores físicos do imóvel, a capacidade de instalação de sensores sem fio, com capacidade de processamento. Permite aos usuários monitoramento de dados fisiológicos, fato que permite aplicações de técnicas de telemedicina, diminuindo assim a necessidade de estadia em hospitais (GUBBI et al., 2013). Embora de extrema utilidade, especialistas apontam que os sistemas de telecomunicações deverão prover estruturas para operação das aplicações de IOT, estrutura essa ocasionada pelo aumento no volume de dados através das redes (PEREIRA, 2017).

A evolução do número de dispositivos *smart* ao redor do mundo possibilitou um grande aumento no tráfego de informações através da internet, informações essas geradas pelas próprias pessoas e também informações de sensores e atuadores espalhados pelo mundo que possibilitaram a automação de atividades diárias (PAVITHRA; BALAKRISHNAN, 2015).

Conforme informação coletada no gráfico da figura 11, os sistemas de automação voltados para residências inteligentes, vão ficar com uma fatia de 14% de todo o mercado de IOT.

Para Piyare (2013), a tecnologia nos fornecerá um novo conceito e uma vasta área para desenvolvimento voltado para automação predial, que poderá proporcionar aos seus usuários espaços inteligente, possibilitando maior conforto.

2.9.1 Trabalhos Relacionados

Após pesquisa realizada, pode-se destacar alguns projetos encontrados que utilizam tecnologia semelhante, ou possuem determinada característica que se aproxime do projeto que é apresentado por esse documento, como:

Morais H. (2017), retratou o desenvolvimento de um sistema de automação para residências, utilizando um controlador lógico programável. O projeto apresenta robustez devido a utilização de sistemas direcionados para indústrias em uma aplicação menos crítica, por outro lado o mesmo apresenta elevado custo financeiro e alta complexidade de integração da solução.

Para critério de comparação, foi utilizado um projeto de automação residencial/predial, utilizado como tema de monografia para obtenção do grau de Engenheiro de Controle e Automação na Universidade Federal de Ouro Preto. Luz (2018) apresentou em seu trabalho um projeto de automação de baixo custo, que tinha por finalidade o acionamento de cargas de maneira remota. O controle do sistema era feito por meio de um aplicativo desenvolvido para plataforma Android, para gerenciamento do *hardware*, foi utilizado um kit Arduíno Mega, juntamente com um módulo HC-06 para comunicação *bluetooth*. Segue alguns pontos relacionando o projeto estudado, com a implementação apresentada por esse documento.

- Padrão para comunicação sem fio *bluetooth*, embora amplamente difundido para aplicações remotas devido ao baixo consumo e disponibilidade de dispositivos no mercado, possui restrições quanto ao tráfego de dados e de maneira mais relevante o seu alcance. O módulo utilizado no projeto é do tipo *bluetooth* 2.0 + EDR, que possui alcance de cerca de 10 metros e limite e tráfego de 3Mbits/s (ELETROGATE, 2018). Tal limitação não permite ao usuário mobilidade para operar seu sistema de locais longe da base controladora. Além da distância, esse meio não é indicado para aplicações onde existem vários “clientes”, na aplicação apresentada existe um limite de sete clientes para o servidor, pela topologia de rede chamada Piconet. Para aplicações com mais conexões

se faz necessário a alteração de topologia e a implementação de códigos mais elaborados para realizar o gerenciamento e tráfego de informações.

O MQTT apresenta vantagens comparadas ao Bluetooth no que diz respeito a:

- Possibilidade de implementação de vários clientes para um mesmo servidor;
- Possibilidade de acionamento e monitoramento a distância, para o usuário atuar sobre o sistema será necessário além de um cliente previamente configurado a conexão com a internet, dessa forma o usuário poderá estar em qualquer lugar do mundo.
- A placa utilizada como central para processamento apresenta características inferiores de capacidade de processamento, e recursos, no que tange aplicações IOT por um preço superior ao ESP32. Após pesquisa realizada no dia 01/10/2018, foi constatado um custo superior médio de R\$30,00 do Arduíno para a ESP32.
- O aplicativo utilizado para gestão do *software* pode apresentar falhas devido a implementação, o projeto apresentado por esse documento, pode ser controlado por qualquer aplicativo cliente MQTT, já testado e disponível no mercado. Fato esse que também não restringe o sistema operacional utilizado pelo usuário, seja ele: IOS, Android etc.
 - Devido ao fato de utilizar MQTT, o sistema de automação apresentado por esse documento permite a conexão com qualquer outro dispositivo móvel, ou aplicação web etc., que possa receber a implementação de algum código MQTT. Dessa forma, possibilitando que o usuário expanda sua possibilidade de aplicações.

3 ESTUDO DE CASO

O projeto apresentado por esse documento, consiste no desenvolvimento de um sistema para controle e gerenciamento de cargas AC, e também para o monitoramento de variáveis como temperatura, e intensidade de campo magnético.

Como dispositivo central do projeto, foi utilizado um kit de desenvolvimento ESP32. Quando proposta a ideia do protótipo, foi realizado uma busca do kit que apresentasse o maior custo benefício, se levado em conta o preço e os recursos disponíveis do mesmo. Na tabela 4 é apresentado um comparativo entre eles.

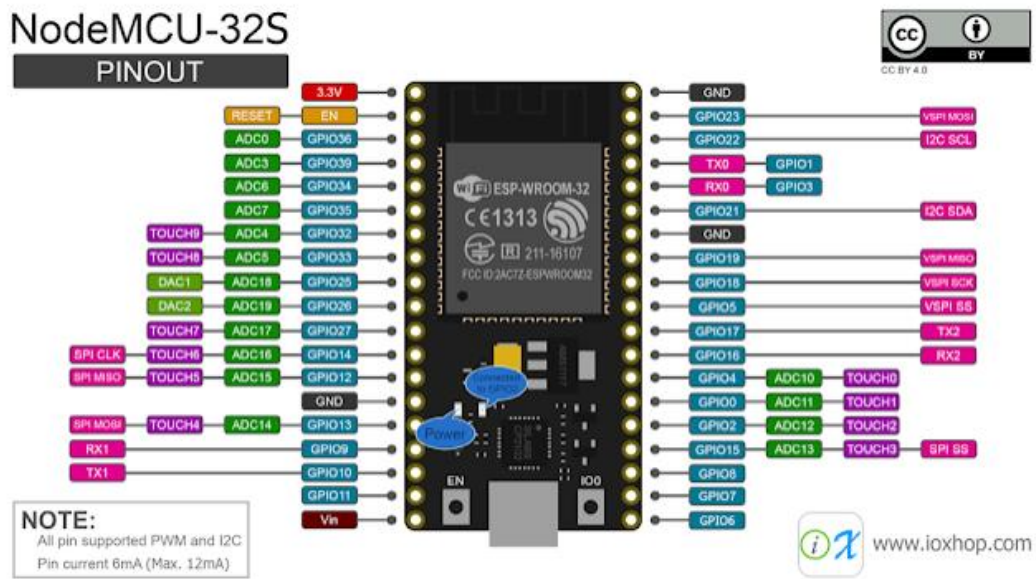
Tabela 4 - Comparativo de Kits

	ESP32	ESP8266	ARDUINO UNO R3
Cores	2	1	1
Arquitetura	32 bits	32 bits	8 bits
Clock	160MHz	80MHz	16MHz
WiFi	Sim	Sim	Não
Bluetooth	Sim	Não	Não
RAM	512KB	160KB	2KB
FLASH	16Mb	16Mb	32KB
GPIO	36	17	14
Interfaces	SPI / I2C / UART / I2S / CAN	SPI / I2C / UART / I2S	SPI / I2C / UART
ADC	18	1	6
DAC	2	0	0

Fonte: Adaptado de KOYANAGE,2017

Dos recursos disponíveis apresentados pelas placas, foram considerados de maneira prioritários, os padrões de comunicação, seguido por quantidade de canais ADC e depois memória RAM. Na figura 13 é apresentada a disposição dos terminais no kit utilizado.

Figura 13 - Esquema de Pinos KIT ESP32



Fonte: Adaptado de KOYANAGE,2017

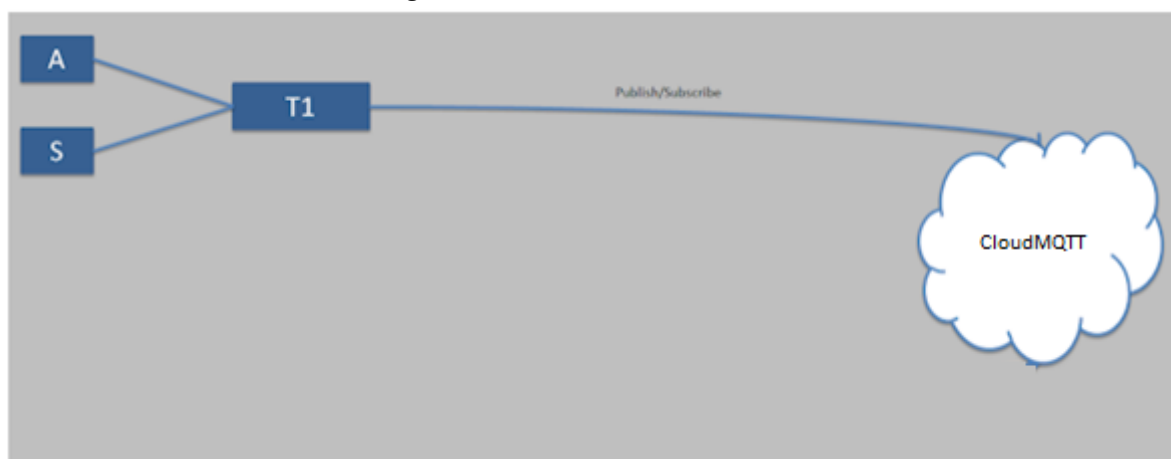
Devido ao propósito experimental do protótipo, ficou definido da implementação de cinco funções, que são apresentadas na tabela 5.

Tabela 5 - Funções Firmware

Tipo	Descrição
Atuador	Relé 1
Atuador	Relé 2
Atuador	Dimmer LED Azul
Sensor	Temperatura
Sensor	Efeito Hall

Dentre as possíveis kits, optou-se por a arquitetura semelhante a apresentada na figura 14. Na tabela 5 é exibido os cargas que o sistema irá monitorar ou atuar.

Figura 14 - Modelo Ilustrativo do Sistema



Fonte: Adaptado de KOYANAGE,2018

3.1 Firmaware

O *firmware* utilizado na ESP32 pode ser compreendido em basicamente três estruturas distintas, sendo elas:

- Configurações Iniciais: Bloco onde é realizada toda a configuração inicial do programa, sendo ela das variáveis, e periféricos utilizados;
- Loop: Nesse bloco é feita a verificação das conexões necessárias para funcionamento e também é implementada a estrutura de atualização dos dados dos sensores segundo o estouro do timer, que ocorre a cada 4 segundos;
- Função Retorno: A operação dessa função ocorre sem a necessidade de chamada da função durante o loop principal ou qualquer outro bloco ou função utilizada no código. Quando algum dado é publicado no *broker* a função verifica o tópico e realiza sua tratativa segundo código implementado.

O código integral que foi utilizado se encontra nos apêndices.

Nas figuras 15 e 16 e 17 são apresentados os fluxogramas de funcionamento dos blocos três blocos.

Figura 15 - Configuração Inicial



Figura 16 - Loop

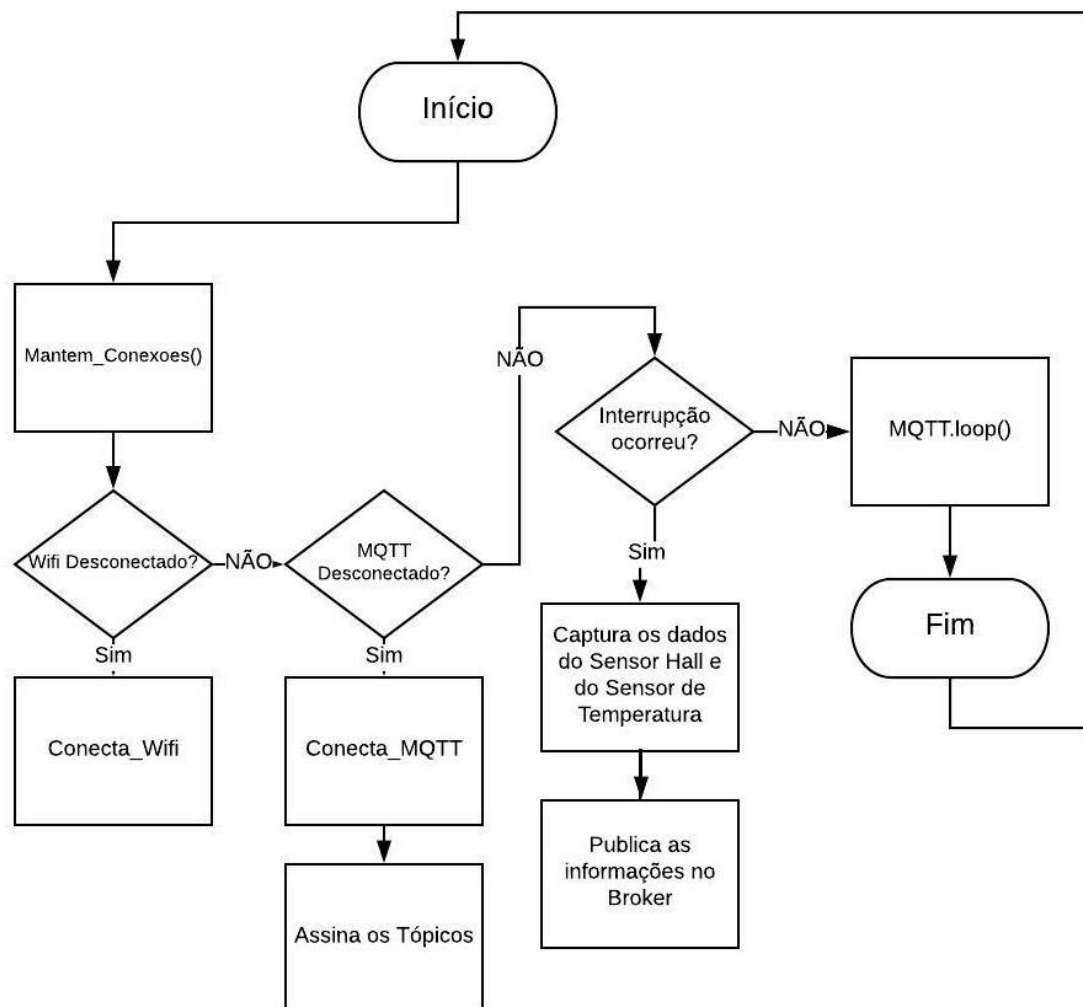
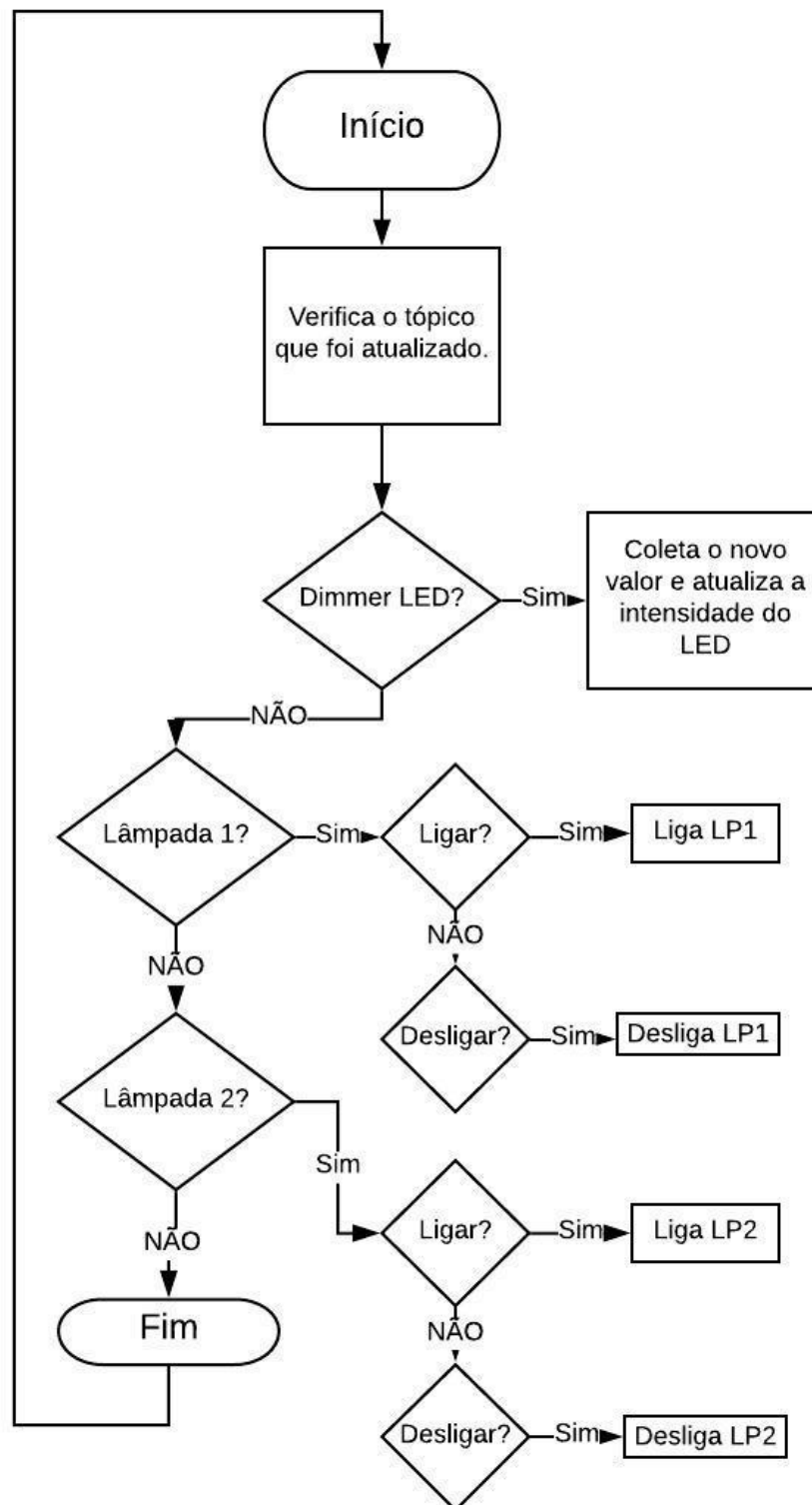


Figura 17 - Função de Retorno



3.2 Esquema Elétrico do Projeto

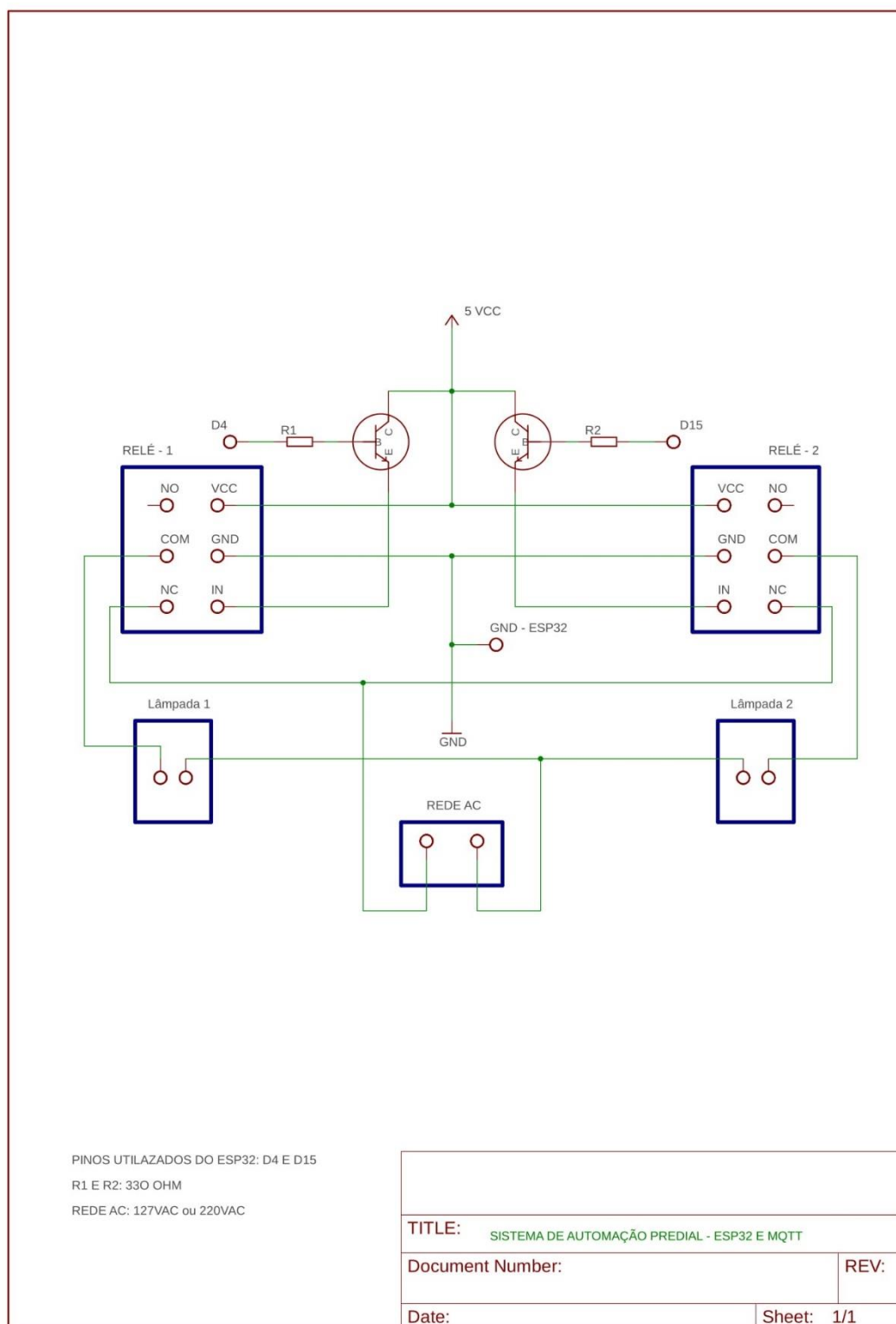
Para o acionamento de cargas AC, foi necessário utilizar um Relé que era controlado pelo sinal encaminhado pelo ESP32. Devido a limitação de corrente e tensão de saída do ESP32 foi necessário utilizar um circuito *driver* com um transistor BJT para acionar o Relé.

Informações retiradas da folha de dados (do inglês, *datasheet*) apresentam que a corrente máxima de saída do pino do ESP32 é de 12mA (EXPRESSIF SYSTEM, 2018), corrente essa inferior a mínima exigida pelo Relé JQC – 3FF – S – Z, como a tensão de saída do ESP32 também não atende os requisitos solicitados para o acionamento do relé, se fez necessário a criação de um circuito driver com um transistor BJT BCC337 da configuração NPN que permitisse uma corrente superior a mínima exigida para comutação dos contatos do relé com um nível de tensão solicitado de 5VDC, para o presente projeto a corrente máxima fornecida pelo circuito driver era de 100mA, corrente máxima permitida pelo transistor.

Na figura 18 é exibido o esquema de ligação do protótipo com os relés e o circuito driver utilizando os transistores, destaca-se o fato que a alimentação do circuito driver e dos relés foi feita por outra fonte de 5VDC, fato esse que acarretou na necessidade de ligar os sinais de terra da ESP com o da fonte externa.

Deve-se ter atenção a ligação das cargas AC, pois eventuais erros de ligação podem ocasionar em acidentes.

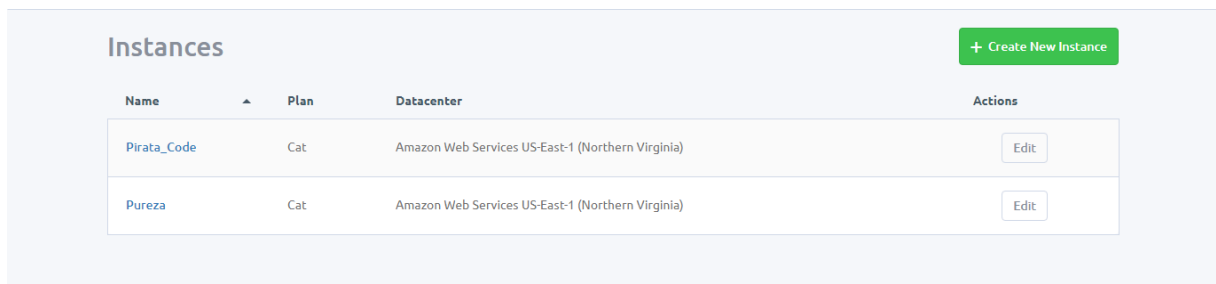
Figura 18 - Esquema de Ligação do Protótipo



3.3 Configuração da Instância do *Broker*

Segundo informações já apresentadas neste documento, o *broker* selecionado para centralizar as informações do sistema foi o *CloudMQTT*, nesta secção será apresentado a maneira de criar uma instância para utilização. Na figura 19 é apresentada a tela inicial do site, para criar uma nova instância deve-se selecionar o botão na cor verde, situado no campo superior direito da imagem.

Figura 19 - Tela Inicial CloudMQTT



Na figura 20, é apresentada a tela inicial de configuração onde é possível definir o nome da instância, o plano que será utilizado. No presente projeto foi optado por utilizar um plano grátis com alguns recursos disponíveis, o determinado plano já possui características suficientes para atender as demandas do projeto.

Figura 20 - Tela Plan MQTT

Progress bar: Plan (active), Region, Configure (Dedicated plans only), Confirm

Select a plan and name - Step 1 of 4

Name: Laboratório de Automação Predial

Plan: Cute Cat (Free)

Tags:

Tags are used to separate your instances between projects. This is primarily used in the project listing view for easier navigation and access control.

Tags allow admins to manage team members access to different groups of instances.

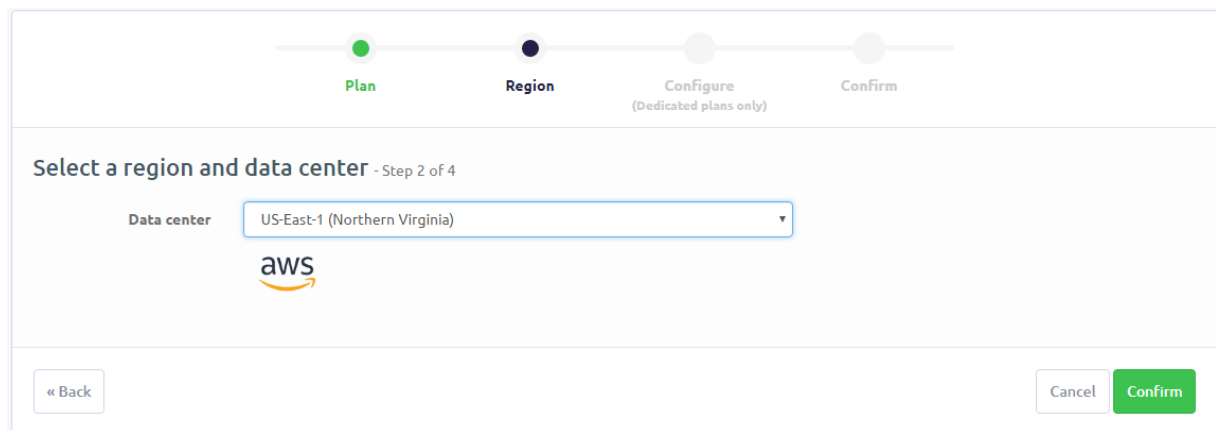
Plan preview: Cute Cat

See the plan page to learn about the different plans.

Buttons: Cancel, Select Region

Na figura 21, é apresentado o menu onde é definido o *datacenter* onde será alocado nossa instância. Para o pacote em questão existe a possibilidade de alocação em dois locais, sendo eles Virgínia do Norte nos Estados Unidos ou Irlanda, para o presente projeto optou-se na utilização do *datacenter* localizado no território americano.

Figura 21 - Definição do Data Center



Plan Region Configure (Dedicated plans only) Confirm

Select a region and data center - Step 2 of 4

Data center US-East-1 (Northern Virginia)

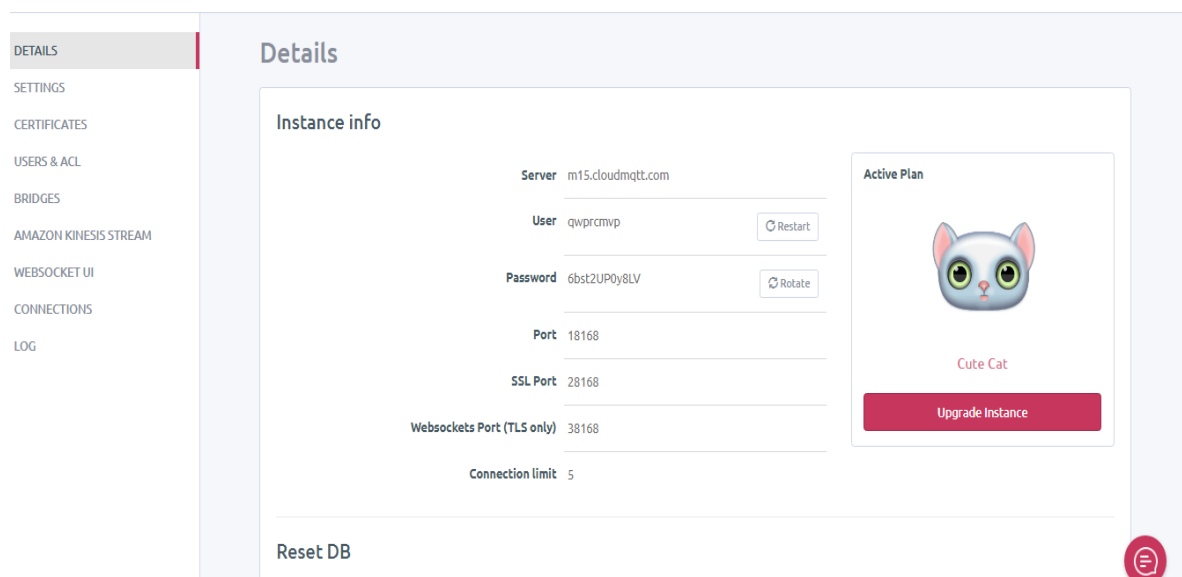
aws

« Back Cancel Confirm

Após a conclusão do passo da imagem 21, as informações necessárias para implementação da instância em uma aplicação são apresentadas na figura 22, como: servidor, usuário, senha, porta etc.

Os dados para acesso são utilizados na configuração do *broker* no *firmware* e também no aplicativo.

Figura 22 - Informações da Instância



DETAILS

SETTINGS

CERTIFICATES

USERS & ACL

BRIDGES

AMAZON KINESIS STREAM

WEBSOCKET UI

CONNECTIONS

LOG

Details

Instance info

Server m15.cloudmqtt.com

User qwprcmvp Restart

Password 6bst2UP0y8LV Rotate

Port 18168

SSL Port 28168

Websockets Port (TLS only) 38168

Connection limit 5

Active Plan

Cute Cat

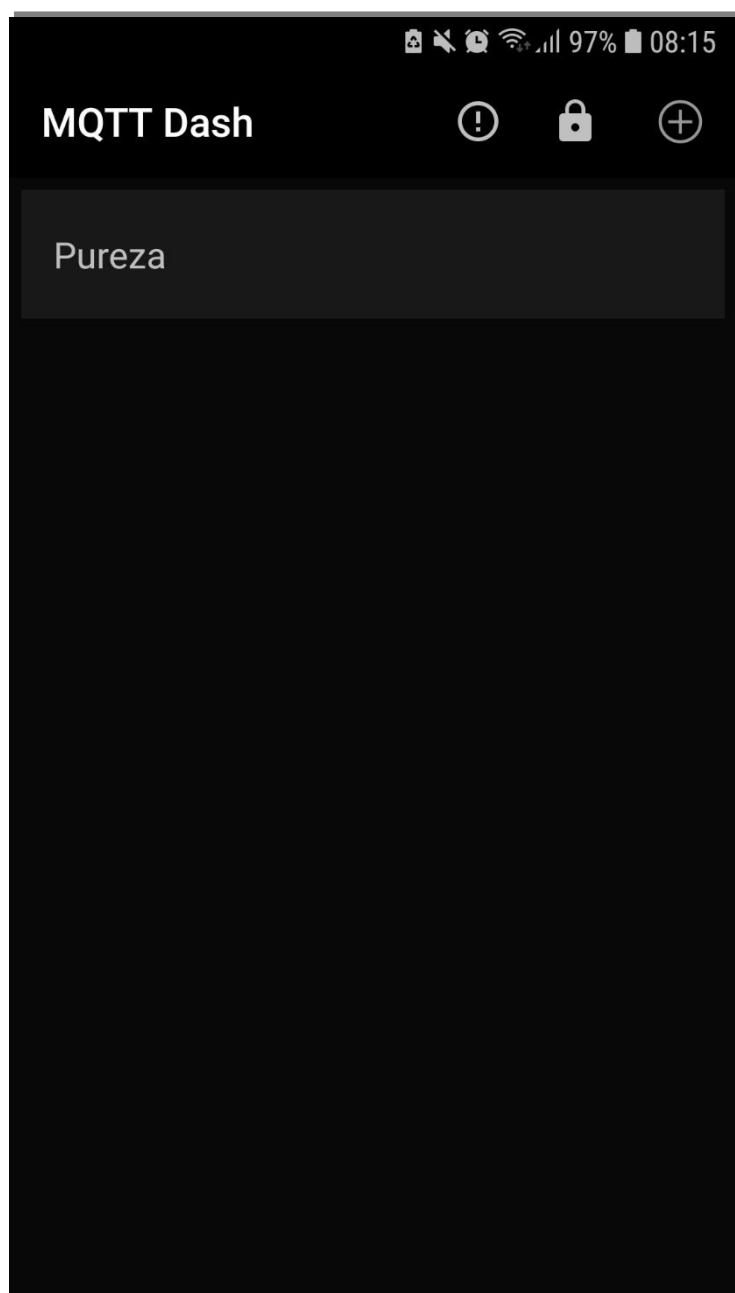
Upgrade Instance

Reset DB

3.4 Aplicativo

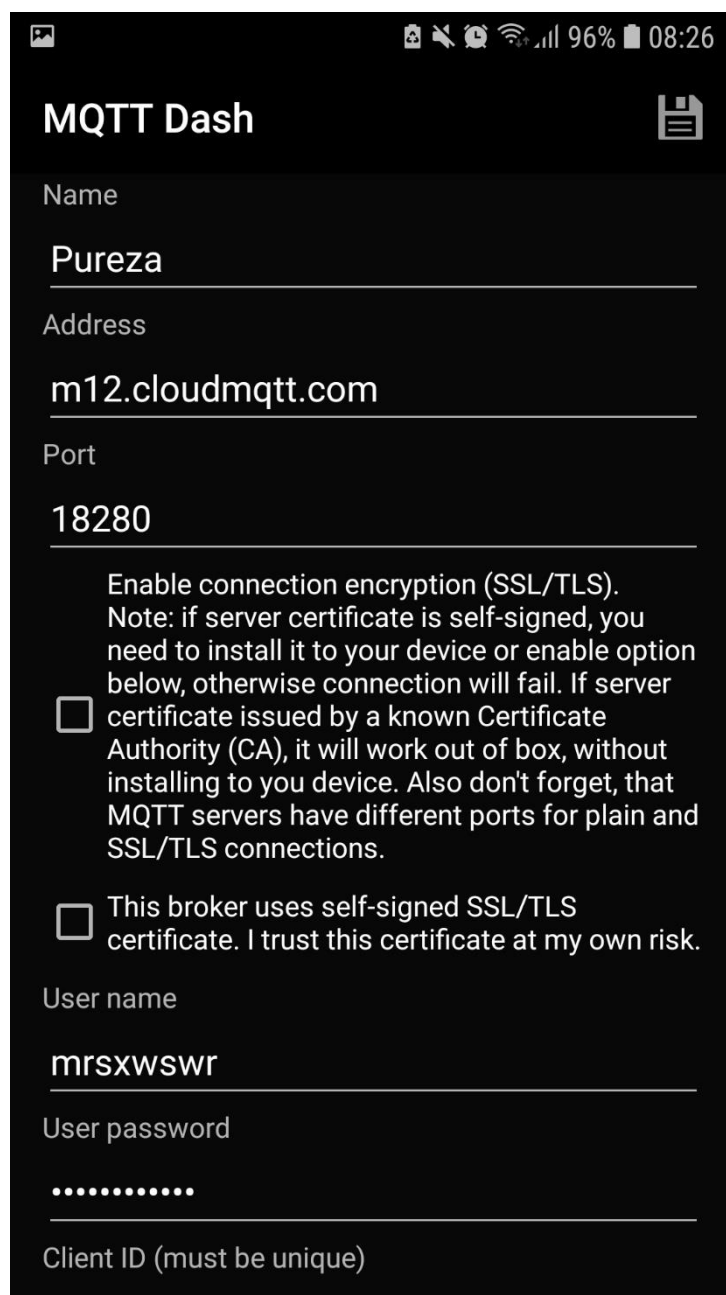
Para gerenciar o sistema, conforme já visto em secções anteriores optou-se pela utilização de serviços de cliente MQTT genéricos, é solicitada apenas a configuração do mesmo para operar junto ao sistema de automação. Nesta secção é apresentado os passos para configuração do aplicativo MQTT Dash, que opera sobre a plataforma Android. Na figura 23 é mostrada a tela inicial do aplicativo.

Figura 23 - Tela Inicial MQTT Dash



Na figura 23 é apresentada a tela inicial do aplicativo, nela é possível verificar todos os clientes criados. O aplicativo permite que eu crie “n” clientes em “n” *brokers* distintos, ficando todos disponíveis para acesso na presente tela. Para criar um novo cliente deve-se clicar no primeiro símbolo no canto superior direito da tela. A tela de configuração do cliente pode ser vista na figura 24.

Figura 24 - Configuração do Cliente



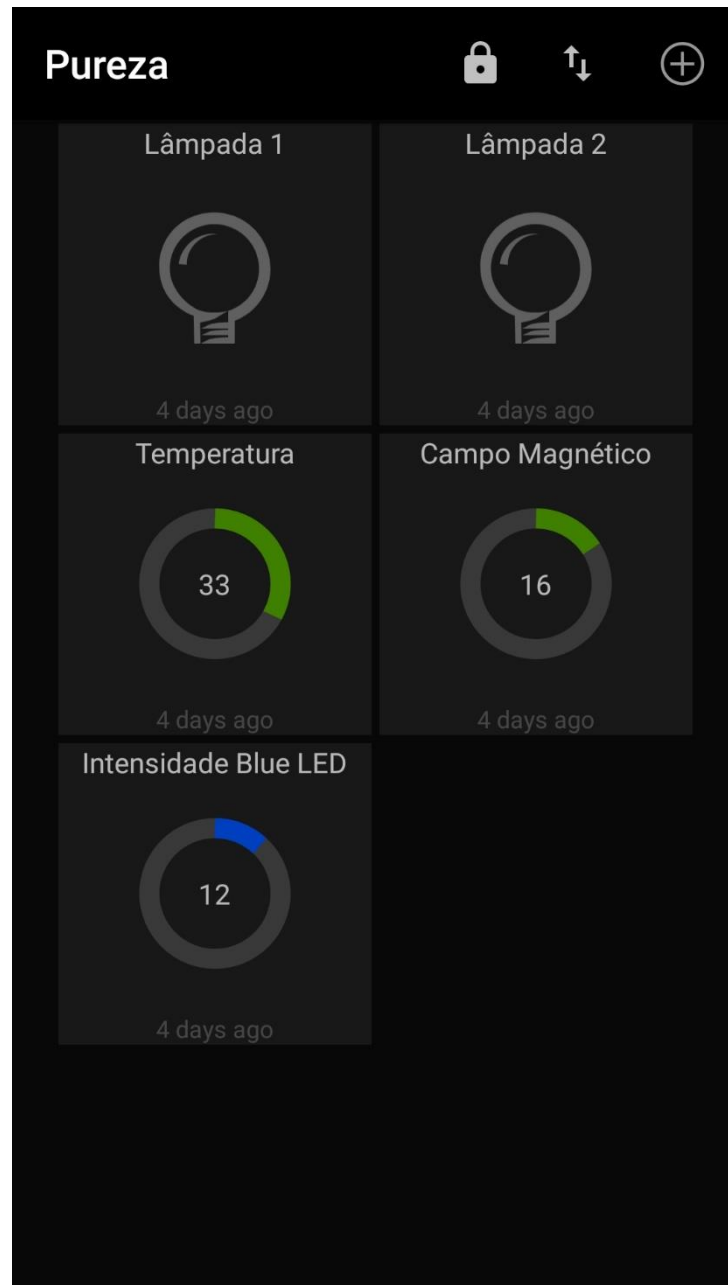
The screenshot displays the 'MQTT Dash' configuration interface. At the top, the status bar shows various icons and '96% 08:26'. The app title 'MQTT Dash' is in the top left, and a save icon is in the top right. The configuration fields are as follows:

- Name:** Pureza
- Address:** m12.cloudmqtt.com
- Port:** 18280
- SSL/TLS Options:**
 - ☐ Enable connection encryption (SSL/TLS). Note: if server certificate is self-signed, you need to install it to your device or enable option below, otherwise connection will fail. If server certificate issued by a known Certificate Authority (CA), it will work out of box, without installing to you device. Also don't forget, that MQTT servers have different ports for plain and SSL/TLS connections.
 - ☐ This broker uses self-signed SSL/TLS certificate. I trust this certificate at my own risk.
- User name:** mrsxwswr
- User password:** (masked with dots)
- Client ID (must be unique):** (empty field)

Os passos básicos para configuração do cliente: definição de um nome, que será exibido na tela do aplicativo. Após deverá ser indicado o endereço do servidor, e sua porta de acesso. Alguns

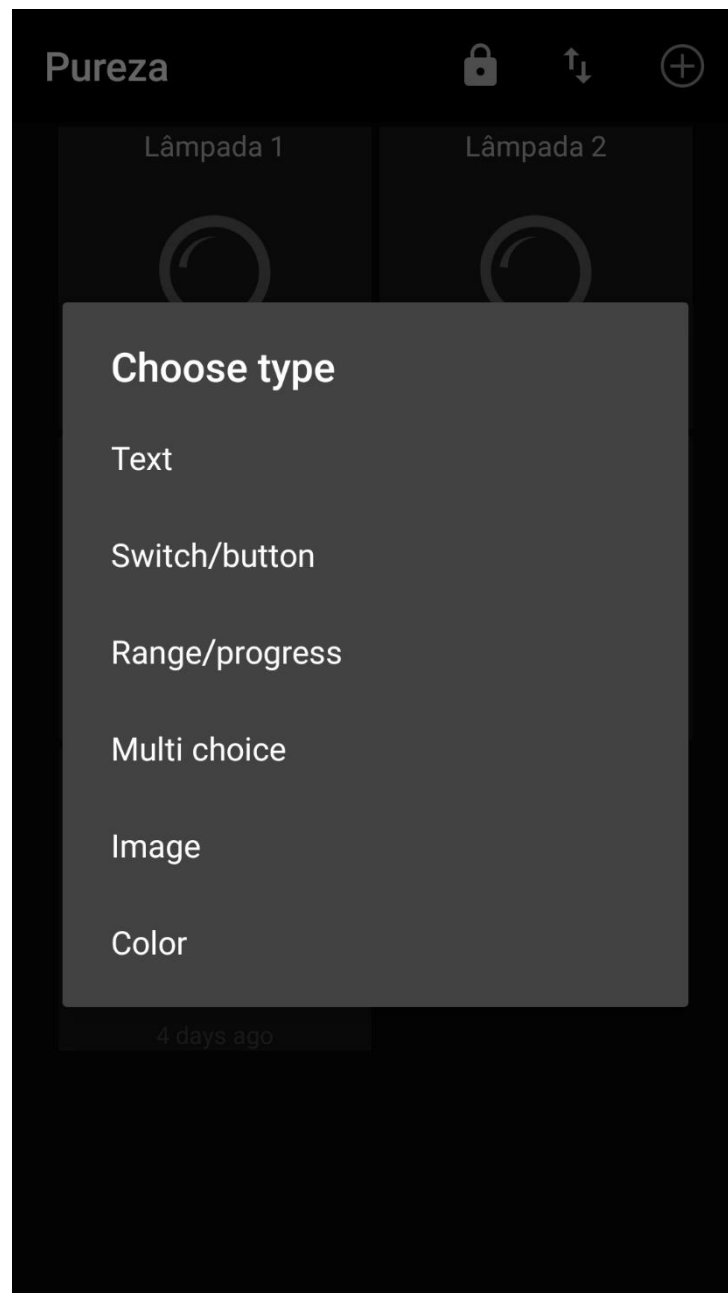
brokers implementam camada de segurança pela utilização de usuário e senha, como o *CloudMQTT*, no caso esses dados deverão ser informados. Na figura 25 é exibido a tela de menu, onde é possível criar os elementos para atuarem sobre os tópicos.

Figura 25 - Menu Inicial



Para criar um elemento, deve-se clicar no primeiro ícone semelhante ao sinal de adição. Presente no canto superior direito da figura. Dessa forma será apresentada uma tela igual a figura 26.

Figura 26 - Inserção de Elementos de Gatilho



Neste aplicativo é disponibilizada ao usuário uma série de gatilhos com diferentes comportamentos, eles são apresentados na figura 26. Neste projeto foram utilizados os elementos chave/botão (do inglês, *switch/button*) e variação/progresso (do inglês, *range/progress*). Na figura 27 são apresentados os requisitos básicos para configuração de um desses elementos citados, no caso o *switch/button*.

Figura 27 - Criação de Elemento Switch

The screenshot shows the 'MQTT Dash' app interface on a mobile device. At the top, the status bar shows icons for signal, Wi-Fi, battery (93%), and time (09:01). The app title 'MQTT Dash' is in the top left, and a save icon is in the top right. The form has the following fields and options:

- Name:** Lâmpada 1
- Topic (sub):** LAMP_1_STATE
- Extract from JSON path:** A text area with a placeholder explaining JSON path extraction and a link to <https://github.com/jayway/JsonPath/blob/master/README.md>.
- Enable publishing:** A checked checkbox.
- Topic (pub):** LAMP_1_STATE, with a note: 'keep empty if the same as sub'.
- Update metric on publish immediately:** A checked checkbox, with a note: '(do not wait for incoming message to update visual state)'. Below this is a paragraph explaining payload and icons for On/Off states.
- On/Off labels:** 'On LIGAD' and 'Off DESLI'.

No menu de criação de um elemento de disparo, deverá ser informado um título para o elemento, também deverá ser especificado o tópico e a opção selecionada na função específica, sendo ela *publish* ou *subscriber*. Também deverão ser identificados os títulos que vão realizar os disparos para ligado ou desligado, e o padrão de segurança que será utilizado.

Após a configuração do cliente e de todos os seus gatilhos, o usuário poderá interagir com o sistema.

3.5 Protótipo Final e Testes

Após elaboração de todas as demandas, e do circuito como um todo, optou-se pela aplicação do projeto em um painel didático que futuramente poderia ser utilizado para auxiliar alunos, profissionais e pessoas que se interessam pela área de automação. Nas figuras 28, 29 e 30 são apresentadas as etapas da montagem.

Figura 28 – Pré-Montagem

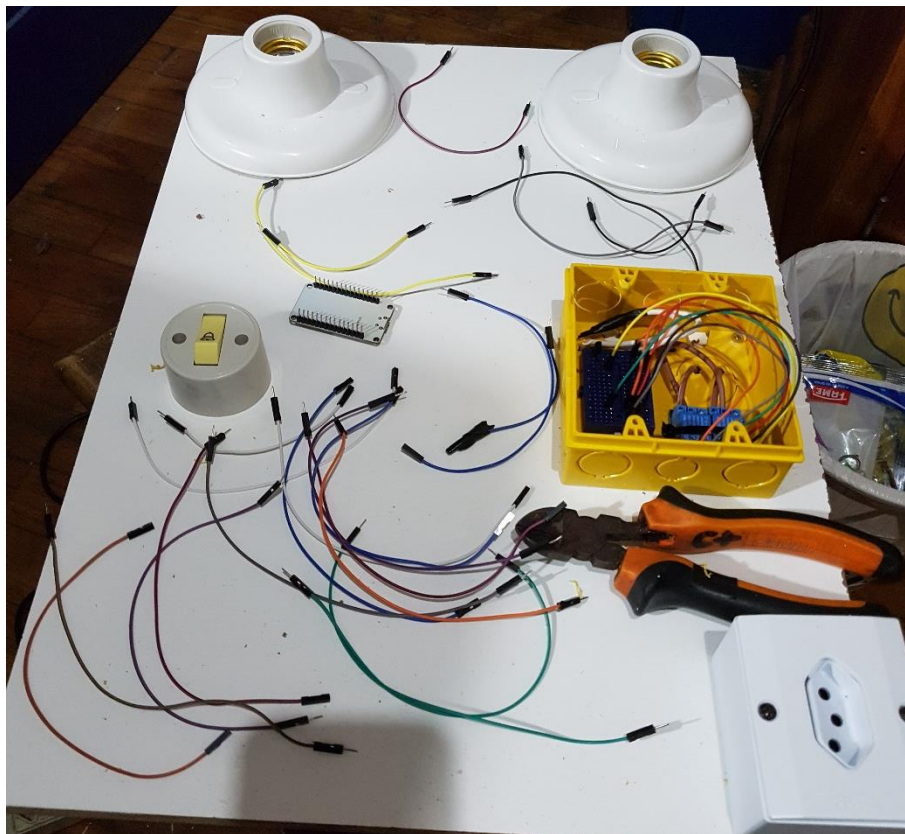


Figura 29 - Pré-Montagem 2



Figura 30 - Montagem Final



No final do desenvolvimento do painel de testes, foi optado por colocar um interruptor para aproximar da melhor forma ao que um profissional pudesse encontrar no mundo real. O interruptor funcionava como um gatilho para alteração do tópico, enviando um sinal em nível de controle apenas para o ESP32, o sinal era interpretado e logo depois uma ação era encaminhada pelo kit. Como a função do interruptor é o de funcionar como um simples botão, o mesmo pode ser conectado em qualquer pino cabível de funcionar como entrada do ESP32. O código apêndices apresenta a forma de configuração do mesmo.

Para aplicação do projeto em um ambiente real, fazem-se necessárias alterações na forma de trabalho conjunta do interruptor com o sistema de automação, pois nesse caso o interruptor deveria alterar o estado da lâmpada e no projeto deveria ser acrescentado um sensor para reconhecer o estado da carga, tal sensor encaminharia a informação para o ESP que trataria e alteraria o tópico segundo o estado da lâmpada. Essa alteração solucionaria algum eventual contratempo com o sistema devido a falha da rede.

No presente projeto não foi optado de se utilizar o sistema de automação de forma conjunta com um interruptor paralelo conforme explicado anteriormente, devido a necessidade de acréscimo do sensor. Dado o caractere acadêmico, de mostrar a estrutura e o formato de montagem optou-se pela arquitetura do interruptor atuando apenas em nível de controle, para implementação futuras pode-se verificar a possibilidade de integrar ao sistema sensores de tensão AC.

Durante os testes realizados, foi possível encontrar resultados de funcionamento conforme o esperado no que diz respeito ao tempo de resposta do sistema, devido à utilização da rede de dados móveis, esta que foi optada de forma a demonstrar a capacidade do sistema de operar com uma rede encontrada em grande parte das cidades do país, e também devido ao fato do IOT estar em crescente junto as novas redes móveis desenvolvidas.

4 RESULTADOS

De maneira conjunta ao desenvolvimento físico do protótipo, foi possível realizar um comparativo com outro trabalho já apresentado na Universidade Federal de Ouro Preto, levantando assim pontos que foram melhorados devido à utilização das tecnologias ESP32 e MQTT, esses dois sistemas permitem aos usuários um advento na solução de novos problemas utilizando IOT.

Foi realizado um levantamento do estado da arte da IOT, apresentando seu histórico e as projeções de mercado, fato esse de extrema relevância para futuros profissionais da área de Engenharia de Controle e Automação, que terão que aprender os novos paradigmas e as novas soluções possíveis com o crescimento desse campo, todo esse documento poderá ser utilizado pelos alunos, pesquisadores e interessados na área para estudo e desenvolvimento de novas tecnologias para automação residencial, predial etc.

A presente monografia poderá servir de base para novas pesquisas na área de automação predial a serem realizadas pelos usuários do Laboratório de Automação Predial da Universidade Federal de Ouro Preto, que utilizem MQTT, serviços em nuvem e IOT.

Ao final do desenvolvimento desse protótipo, foi possível implementar com sucesso o sistema de automação residencial, utilizando um ESP32 e o serviço de *broker CloudMQTT*, controlado este através de um aplicativo utilizando a plataforma Android. O sistema permitiu o controle de 2 cargas AC, a dimerização de 1 carga em corrente contínua (do inglês, *direct current* – DC) e também o acompanhamento da variação de duas variáveis analógicas com um intervalo de atualização de 4s.

Durante a fase de integração do painel didático onde foi implementado o projeto duas situações de problema foram encontradas, e até o fim da execução do presente relatório não foi possível encontrar a solução. Durante os testes de falha de conexão, ocorreram de maneira aleatória falha na reconexão, ou reconexão com alteração do tópico lâmpada 1 de forma indevida.

Verificou-se que o problema ocorreu devido ao modelo da montagem, mesma sendo realizada em um *protoboard*, fator esse que em alguns momentos causou ruído ao sistema. A opção por montagem desse tipo foi realizado devido ao caráter acadêmico e de pesquisa.

Para aplicação como produto deve-se optar pela elaboração de uma PCB para acoplar o sistema, fator esse que possa auxiliar na blindagem contra ruídos.

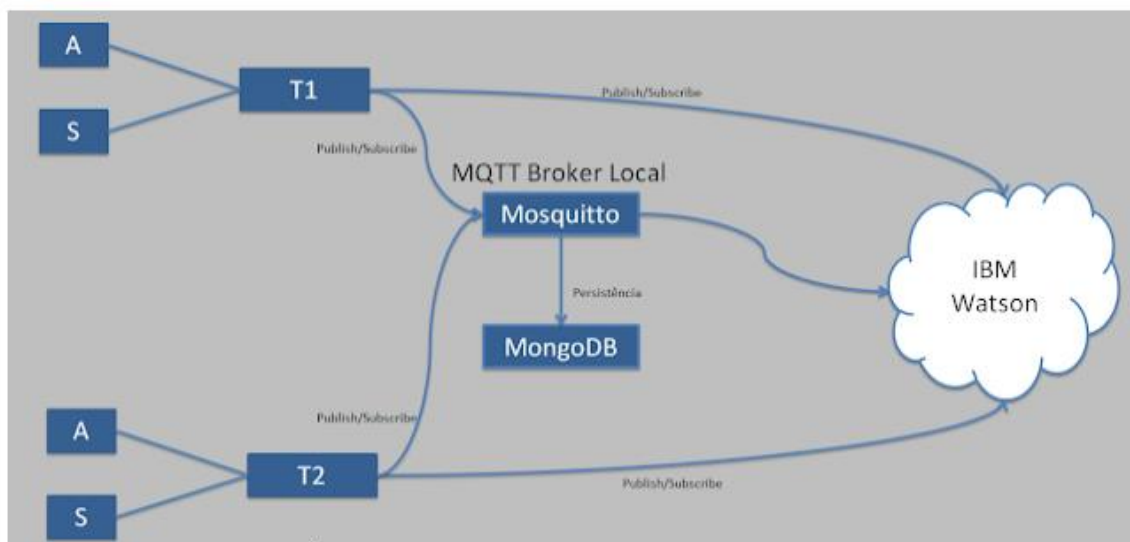
5 CONSIDERAÇÕES FINAIS

Ao final do período de desenvolvimento do presente trabalho, pode-se verificar que os objetivos foram alcançados conforme esperado, com eventuais falhas devido ao método de montagem. Na secção 3 e 4 foram apresentadas possibilidades de como solucionar os problemas de montagem e também como deve-se modificar o projeto para aplicação real. Novas possibilidades de aplicação apareceram durante o processo de pesquisa, algumas delas foram aplicadas e outras estão destacadas como sugestão para trabalhos futuros, como redundâncias e novos possíveis mercados.

5.1 Sugestões para Trabalhos Futuros

Poderá ser considerado o desenvolvimento de um sistema com soluções que permitam a integração com outros aplicativos, softwares ou páginas web, de forma semelhante à apresentada por Pereira (2017). Além dessa integração, poderá ser aplicado um sistema com redundância, que permitam a utilização de banco de dados, fato esse que auxilia na tratativa e de operações críticas e eventuais falhas da rede. Arquitetura sugerida é apresentada na figura 28. Deve ser ressaltado que a inserção de redundância do sistema acarreta em aumento dos custos e da complexidade do sistema, a mesma se justificando para operação de dados críticos.

Figura 31 - Arquitetura com Banco de Dados



Fonte: KOYANAGE, 2018

Com o advento do número de dispositivos conectados à internet coletando e compartilhando dados, cresce a preocupação quanto à possibilidade de um eventual ataque malicioso, fato esse que entregaria dados a pessoas e empresas sem a permissão do usuário, acarretando em risco

ao usuário. De forma a solucionar esse problema, pode-se optar pela elaboração de um sistema IOT atrelado a tecnologia *Blockchain* tecnologia essa que é explorado em áreas como o de criptomoedas, essa tecnologia traz como vantagem a possibilidade da realização de aplicações complexas, com camadas de autenticação e controle de dispositivos. Além do que a descentralização oferecida por essa solução é de extrema relevância para aumentar a segurança e a integridade da informação que é trafegada pela internet (MONACO, 2018).

Para novas pesquisas, poderá ser considerado outros mercados para aplicação de IOT, como o mercado de marketing onde a utilização de tecnologia como *Beacons* vem ganhando espaço para criação de campanhas e análise de comportamento de clientes, segundo sua localização (MORAIS, M., 2017). E também deve ser levada em conta a aplicação de tecnologias oriundas do IOT no campo, visto ser uma área onde é crescente a necessidade de tecnologia para melhoramento da gerencia e produção de cultivos de produtos e controle de animais, tais informações se encontram na entrevista dada pelo diretor de tecnologia da multinacional Logicalis (FUTURECOM, 2018).

Diante dos resultados obtidos nesse trabalho e das projeções de crescimento do mercado para tecnologia, pode-se concluir que aplicações IOT é um imenso campo que deve ser explorado cada vez mais de maneira a auxiliar a vida do ser humano. Pois embora muitas pesquisas já tenham sido realizadas esse campo se apresenta como uma válvula para advento de um planeta cada vez mais eficiente e melhor para se viver.

REFERÊNCIAS BIBLIOGRÁFICAS

- ADAFRUIT. **Overview**.2015. Disponível em: < <https://learn.adafruit.com/adafruit-io/overview> >. Acesso em: 23 set. 2018.
- ARDUINO. **What is Arduino**. Disponível em: < <https://www.arduino.cc/en/Guide/Introduction> >. Acesso em: 23 set. 2018.
- BANKS, A; GUPTA, R.**MQTT Version 3.1.1**. 2014. OASIS Standart. Disponível em: < <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html> > Acesso em: 24 set. 2018.
- BARROS, M. MQTT – Protocolos para Iot. **Embarcados**.2015. Disponível em: < <https://www.embarcados.com.br/mqtt-protocolos-para-iot/> >. Acesso em: 24 set. 2018.
- CLOUDMQTT. **Plans e Pricing**. 2018. Disponível em: < <https://www.cloudmqtt.com/plans.html> >. Acesso em: 05 out. 2018.
- COLUMBUS, L. 2017 Round of internet of things forecasts. **Forbes**. 2016. Disponível em: < <https://www.forbes.com/sites/louisecolumbus/2017/12/10/2017-roundup-of-internet-of-things-forecasts/#94cca5e1480e> >. Acesso em: 20 set. 2018.
- ELETROGATE. **Módulo Bluetooth RS232 - HC06**. Disponível em: < https://www.eletragate.com/modulo-bluetooth-rs232-hc06?utm_source=Site&utm_medium=GoogleMerchant&utm_campaign=GoogleMerchant&gclid=Cj0KCQjwi8fdBRCVARIsAEkDvnL9O4HECnEUa8diPpyHB-jjJtx0bAYG4ZSGXjpqTQ1jL-CHHtRg4pEaAnBSEALw_wcB >. Acesso em: 01 out. 2018.
- EMBARCADOS, E. Sistema Embarcado – O que é? Qual a sua importância? **Embarcados**. 2013. Disponível em: < <https://www.embarcados.com.br/sistema-embarcado/> >. Acesso em: 14 set. 2018.
- ESPRESSIF SYSTEMS. **Espressif achieves the 100-million target for iot chip shipments**. 2018. Shanghai-China. Disponível em: < https://www.espressif.com/en/media_overview/news/espressif-achieves-100-million-target-iot-chip-shipments >. Acesso em: 23 set. 2018.
- ESPRESSIF SYSTEMS. **ESP32 Series datasheet**. 2.5. ed. 2018. 53 p.
- FUTURECOM. Setor agro é o mais preparado para receber IOT. **Futurecom Digital**. 2018. Disponível em: < https://digital.futurecom.com.br/2018/06/21/setor-agro-e-o-mais-preparado-para-receber-iot/?utm_source=linkedin&utm_medium=cpc&utm_campaign=lead-credenciamento&utm_content=09/2018_agro_futurecom >. Acesso em 15 out. 2018.
- GUBBI, J. et al. (2013). Internet of Things (Iot): A Vision, Architectural Elements, and Future Directions. **Future Generation Computer Systems**. Australia. 2013. 1645-1660. Disponível em: < <https://arxiv.org/ftp/arxiv/papers/1207/1207.0203.pdf> >. Acesso em: 26 set. 2018.
- GROWTHENABLER. **Market Pulse Report, Internet of Things (IOT)**. Reino Unido. 2017.
- KOYANAGI, F. **Introdução ao ESP32**. 2017. Disponível em: < <https://www.fernandok.com/2017/11/introducao-ao-esp32.html> >. Acesso em: 24 set. 2018.
- KOYANAGI, F. **IBM Watson com ESP32 como Endpoint**. 2018. Disponível em: < <https://www.fernandok.com/2018/05/ibm-watson-com-esp32-como-endpoint.html> >. Acesso em: 04 out. 2018.

LEVITON. **Home automation your introduction to the simplicity of control**. Melville, Nova Iorque. 2013. 16 p.

LIMA, F. S. **A automação e sua evolução**. 2003. 4 f. Artigo – Programa de Pós Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Norte, Natal.

LINS, T. **Internet das Coisas: Coletando dados na IOT**, Ouro Preto.2015. Disponível em: < <http://www.decom.ufop.br/imobilis/iot-coletando-dados/> > Acesso em: 13 set. 2018.

LUZ, H. J. S. **Sistema de automação residencial/predial utilizando arduino e sistema operacional Android**. 2018. 48 f. Monografia (Graduação em Engenharia de Controle e Automação) - Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, 2018.

MANDULA, K. et al. **Mobile based Horne Automation using Internet of Things (IoT)**. In: International Conference on Control, Instrumentation, Communication and Computational Technologies (LCCICCT), 2015, Kumaracoil-India.

MEHI, E. L. M. **Sistemas eletrônicos embarcados**. Material do Professor. Departamento de Engenharia Elétrica. Universidade Federal do Paraná. 2011. Disponível em: < <http://www.eletrica.ufpr.br/mehl/te200/aulas/embarcados.pdf> > Acesso em: 15 set. 2018.

MINERVA, R; BIRU, A; ROTONDI, D. **Towards a definition of the Internet of Things (IOT)**.IEEE Internet of Things. 2015. 86 p. Disponível em: <<https://iot.ieee.org/definition.html> >. Acesso em: 26 set. 2018

MOHAN, C. M. et al. **IOT Based Home Automation**. In: International Journal of Innovative Technology and Research (IJITR), n. 5.2017, p 5979 – 5871.

MONACO, J. **Blockchain e Internet das Coisas: uma combinação vencedora**.2018. Disponível em: < <http://www.semantix.com.br/blog/blockchain-e-internet-das-coisas-uma-combinacao-vencedora/> >. Acesso em: 15 out. 2018.

MORAIS, H. S. **Automação residencial: sistema integrado de uma residência inteligente utilizando controlador lógico programável**. 2017. 77 f. Monografia (Graduação em Engenharia Elétrica) – Instituto de Ciências Exatas e Aplicadas, Universidade Federal de Ouro Preto, João Monlevade, 2017.

MORAIS, M. **Os benefícios dos beacons para empresas**. 2017. Disponível em < <https://usemobile.com.br/beneficios-dos-beacons-para-empresas/> >. Acesso em: 15. out 2018.

MOSQUITTO. **Eclipse Mosquitto**. Disponível em < <https://mosquitto.org/> >. Acesso em: 25 set. 2018.

MQTT.ORG. **Frequently Asked Questions**. 2018. Disponível em < <http://mqtt.org/faq> >. Acesso em: 24 set.2018.

NERYS, J. W. L. **Microprocessadores e Microcontroladores**. Material do Professor. Escola de Engenharia Elétrica e de Computação. Universidade Federal de Goiás.2016. Disponível em < https://inf.ufes.br/~zegonc/material/Introducao_a_Computacao/Microprocessadores%20-%20Parte%201.pdf > Acesso em: 15 set. 2018.

PALMIERE, S. E. CLP versus Microcontrolador. **Embarcados**. Disponível em: < <https://www.embarcados.com.br/clp-versus-microcontrolador/> >. Acesso em: 16 set. 2018.

PAVITHRA, D; BALAKRISHNAM, R. IOT based monitoring and control system home automation. In: **Global Conference on Communication Technologies**, 2015, Thuckalay - India.

PEREIRA, C. E. P. A internet das Coisas (IoT): Cenário e Perspectivas no Brasil e Aplicações Práticas. In: **VII SRST – Seminário de Redes e Telecomunicações**. 2017, Santa Rita do Sapucaí – MG.

PIYARE, R. Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone. In: **International Journal of Internet of Things**, v. 2, n. 1, 2013, Córrea do Sul, p. 5-11

PONCELA, J., MORENO, J, M. AAMIR, M. Analysis of m2m capabilities in 4g, Wire- less Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE). In: **International Conference on IEEE**, 4º, 2014, Aalborg - Dinamarca, p. 1–5.

RODRIGUES, E; PEDO, R.; TEDESCO, L. P. Sistemas Embarcados e sua Aplicação na Indústria. In: **Workshop em Sistemas e Processos Industriais**, 2013, Santa Cruz do Sul – Brasil.

SCHNEIDER ELETRIC. **Modicon is now Schneider Electric**. Disponível em < <https://www.schneider-electric.co.uk/en/about-us/company-profile/brands/modicon.jsp> > Acesso em: 14 set. 2018.

SGARBI, J. A; TONIDANDEL, F. Domótica Inteligente: Automação residencial baseada em comportamento. In: **Workshop de Teses e Dissertações em Inteligência Artificial**, 2006, Ribeirão Preto – SP.

SILEVIRA, L; LIMA, W. Q. **Um breve histórico conceitual da Automação Industrial e Redes para Automação Industrial**. 2003. 3f. Artigo – Programa de Pós Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Norte, Natal.

SOUZA, N. S. **Apostila de acionamentos elétricos**. Material do Professor. Instituto Federal de Educação, Ciência e Tecnológica do Rio Grande do Norte. 2009. Disponível em < <https://docente.ifrn.edu.br/heliopinheiro/Disciplinas/maquinas-e-acionamentos-eletricos-ii/apostila-basica> >. Acesso em: 23 set. 2018.

SOUZA, S. F. **A contribuição da automação residencial na solução de problemas de acessibilidade no cotidiano do idoso**. 2015. 134 f. Dissertação (Mestrado em Arquitetura e Urbanismo) – Universidade Federal de Viçosa, Viçosa, 2015.

Instituto de Engenheiros Eletricistas e Eletrônicos. **Internet of Things (IOT) Ecosystem Study**. 2015. 3 p. Disponível em: < https://iot.ieee.org/images/files/pdf/iot_ecosystem_exec_summary.pdf >. Acesso em: 26 set. 2018.

TEZA, V. R. **Alguns aspectos sobre a automação residencial – domótica**. 2002. 106 f. Dissertação (Mestrado em Ciências da Computação) – Universidade Federal de Santa Catarina, Florianópolis, 2002.

VIEGAS, D. F. Smart cities: **embarcando plataformas generalistas**. 2017. 119 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Minas Gerais, Belo Horizonte, 2017,

YANG, H.; LEE, W. LEE, H. IoT Smart Home Adoption: The Importance of Proper Level Automation. In: **Journal of Sensors**. 2018

YUAN. M. **Conhecendo o MQTT**. 2017. Disponível em <
<https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html>>.
Acesso em: 25 set 2018.

APÊNDICES

Código Utilizado no Arduino

```
/*
 *   Universidade Federal de Ouro Preto
 *   Engenharia de Controle e Automação
 *   Monografia: Implantação de um sistema de automação predial utilizando
 *   plataforma em nuvem.
 *   Este código foi criado por Guilherme Rafael Pereira da Silva, na IDE do Arduino.
 *   O código foi feito para uma placa de desenvolvimento ESP32.
 *
 */

#include <WiFi.h>
#include <PubSubClient.h>

//ID para ESP
#define ID_ESP "ESP32Client"
#define LIGADA "On"
#define DESLIGADA "Off"

// Descrição das Finalidades
#define LED_BLUE 2
#define LED_BLUE_DIR (pinMode(LED_BLUE,OUTPUT))
#define LED_BLUE_ON (digitalWrite(LED_BLUE,HIGH))
#define LED_BLUE_OFF (digitalWrite(LED_BLUE,LOW))
#define LAMP_1 4
#define LAMP_1_DIR (pinMode(LAMP_1,OUTPUT))
#define LAMP_1_ON (digitalWrite(LAMP_1,HIGH))
#define LAMP_1_OFF (digitalWrite(LAMP_1,LOW))
```

```

#define LAMP_2 15
#define LAMP_2_DIR (pinMode(LAMP_2,OUTPUT))
#define LAMP_2_ON (digitalWrite(LAMP_2,HIGH))
#define LAMP_2_OFF (digitalWrite(LAMP_2,LOW))
#define INTERRUPTOR 5

// Tópicos
#define TOPIC_LED      "LED_STATE"
#define TOPIC_LED_D    "LED_DIMMER"
#define TOPIC_LAMP_1   "LAMP_1_STATE"
#define TOPIC_LAMP_2   "LAMP_2_STATE"
#define TOPIC_TEMPERATURA "TEMPERATURA"
#define TOPIC_HALL     "HALL"


// Configuração necessária do Sensor de Temperatura Interno
#ifdef __cplusplus
extern "C" {
#endif
uint8_t temprature_sens_read();
#ifdef __cplusplus
}
#endif
uint8_t temprature_sens_read();


// Redes disponíveis para Acesso

//const char* ssid = "PUREZA";
//const char* password = "pureza79anos";

const char* ssid = "Guilherme";
const char* password = "cruzeiro";

```

```

// Cliente Wifi
WiFiClient espClient;

// Definições para o CloudMQTT
const char* mqttServer = "m12.cloudmqtt.com";
const int mqttPort = 18280;
const char* mqttUser = "mrsxwswr";
const char* mqttPassword = "3mjtNjqsdneB";

// Variáveis auxiliares para conversão de sinal dos sensores internos
double temp_atual = 0;
double hall = 0;
String Temperatura;
String Hall;

// LED Dimmer
int freq = 5000;
int ledChannel = 0;
int resolution = 8;
int dutyCycle = 0;

int cont;

// Estado da Luz 1
int est_l = 0;

// Flag Timer
volatile int interruptCounter;
int totalInterruptCounter;

```

```

hw_timer_t * timer = NULL;

portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;


// Cria um cliente para o serviço MQTT
PubSubClient MQTT(espClient);


// Protótipos das Funções
void Conecta_Wifi (void);
void Mantem_Conexoes(void);
String leStringSerial(void);
void retorno(char* topico, byte* mensagem, unsigned int tamanho);
void IRAM_ATTR onTimer(void);


void setup()
{

    LAMP_1_OFF;
    LAMP_1_DIR;

    LAMP_2_OFF;
    LAMP_2_DIR;


    // Config LED as a Dimmer
    ledcSetup(ledChannel, freq, resolution);
    ledcAttachPin(LED_BLUE, ledChannel);
    ledcWrite(ledChannel, dutyCycle);


    // Conexão Serial
    Serial.begin(9600);

```

```

// Conexão Wifi
Conecta_Wifi();

// Conexão MQTT
MQTT.setServer(mqttServer, mqttPort);
MQTT.setCallback(retorno);

// Config Timer
timer = timerBegin(0, 80, true); // Setup Timer 0
timerAttachInterrupt(timer, &onTimer, true);
timerAlarmWrite(timer, 4000000, true); // 45s
timerAlarmEnable(timer);

// Configurando Pino do Interruptor
pinMode(INTERRUPTOR, INPUT_PULLUP);
}

void loop()
{
  Mantem_Conexoes();
  if (digitalRead(INTERRUPTOR)==LOW)
  {
    while (digitalRead(INTERRUPTOR) == LOW);
    delay(500);
    if (est_l == 0)
    {
      LAMP_1_ON;
      est_l = 1;
      MQTT.publish(TOPIC_LAMP_1,"LIGADA");
    }
  }
}

```

```

    }
    else
    {
        LAMP_1_OFF;
        MQTT.publish(TOPIC_LAMP_1,"DESLIGADA");
        est_1 = 0;
    }
}

if (interruptCounter > 0) {

    portENTER_CRITICAL(&timerMux);
    interruptCounter--;
    portEXIT_CRITICAL(&timerMux);

    // Sensor de Efeito Hall
    hall = hallRead();
    Hall = String(hall);
    char HALLL[Hall.length()+1];
    Hall.toCharArray(HALLL,Hall.length());
    MQTT.publish(TOPIC_HALL,HALLL);

    // Sensor de Temperatura
    temp_atual = (((temprature_sens_read() - 32) / 1.8)-20);
    Temperatura = String(temp_atual);
    char TEMP[Temperatura.length() + 1];
    Temperatura.toCharArray(TEMP,Temperatura.length());
    MQTT.publish(TOPIC_TEMPERATURA,TEMP);

    totalInterruptCounter++;
}

```

```

//Serial.print("An interrupt as occurred. Total number: ");
//Serial.println(totalInterruptCounter);

}
MQTT.loop();
}

void Conecta_Wifi (void)
{
// Conexão WIFI
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED)
{
//WiFi.begin(ssid, password);
delay(500);
//Serial.println("Connecting to WiFi..");

if (digitalRead(INTERRUPTOR)==LOW)
{
while (digitalRead(INTERRUPTOR) == LOW);
delay(500);
if (est_l == 0)
{
LAMP_1_ON;
est_l = 1;
//MQTT.publish(TOPIC_LAMP_1,"LIGADA");
}
else
{
LAMP_1_OFF;

```

```

    est_1 = 0;
    //MQTT.publish(TOPIC_LAMP_1,"DESLIGADA");
}
}
}
Serial.println("Connected to the WiFi network");
LED_BLUE_ON;
delay(1000);
LED_BLUE_OFF;
delay(1000);
LED_BLUE_ON;
delay(1000);
LED_BLUE_OFF;
delay(1000);
}

void Mantem_Conexoes (void)
{
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.println("Connecting to WiFi..");
        Conecta_Wifi();
        //WiFi.begin(ssid, password);
    }
    while (!MQTT.connected())
    {
        Serial.println("Connecting to MQTT...");
        if (MQTT.connect("ESP32Client", mqttUser, mqttPassword )) // Definições de
Configuração para o Cloud MQTT

```

```

{
  Serial.println("connected");
  if (est_l == 0)
  {
    MQTT.publish(TOPIC_LAMP_1,"LIGADA");
  }
  else
  {
    MQTT.publish(TOPIC_LAMP_1,"DESLIGADA");
  }
  MQTT.subscribe(TOPIC_LED);
  MQTT.subscribe(TOPIC_LAMP_1);
  MQTT.subscribe(TOPIC_LAMP_2);
  MQTT.subscribe(TOPIC_LED_D);
}
else
{
  Serial.print("failed with state ");
  Serial.print(MQTT.state());
  delay(2000);
}
}
}

```

String leStringSerial(void)

```

{
  String conteudo = "";
  char caractere;

```

```

// Enquanto receber algo pela serial

```

```

while (Serial.available() > 0) {
    // Lê byte da serial
    caractere = Serial.read();
    // Ignora caractere de quebra de linha
    if (caractere != '\n') {
        // Concatena valores
        conteudo.concat(caractere);
    }
}

//Serial.print("Recebi: ");
//Serial.println(conteudo);

return conteudo;
}

void retorno(char* topico, byte* mensagem, unsigned int tamanho)
{
    mensagem[tamanho] = '\0';
    String strMensagem = String((char*)mensagem);
    String strtopico = String((char*)topico);

    //Serial.println(strtopico);
    //Serial.println(strMensagem);

    // LED
    if (strtopico==TOPIC_LED_D)
    {
        dutyCycle = map(strMensagem.toInt(), 0, 100, 0, 255);
        ledcWrite(ledChannel, dutyCycle);
    }
}

```

```

Serial.print("Intensidade Luminosa do LED: ");
Serial.print(strMensagem.toInt());
Serial.println("%");
delay(250);
}
// Lâmpada 1
else if (strtopico == TOPIC_LAMP_1)
{
    if (strMensagem == "LIGADA")
    {
        LAMP_1_ON;
        Serial.println("Lampada - 1 - Ligada");
        est_1 = 1;
    }
    else if (strMensagem == "DESLIGADA")
    {
        LAMP_1_OFF;
        Serial.println("Lampada - 1 - Desligada");
        est_1 = 0;
    }
}
// Lâmpada 2
else if (strtopico == TOPIC_LAMP_2)
{
    if (strMensagem == "LIGADA")
    {
        LAMP_2_ON;
        Serial.println("Lampada - 2 - Ligada");
    }
    else if (strMensagem == "DESLIGADA")

```

```

{
    LAMP_2_OFF;
    Serial.println("Lampada - 2 - Desligada");
}
}
}

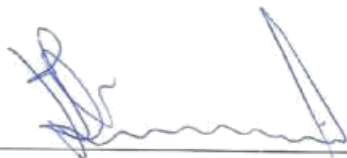
```

```

void IRAM_ATTR onTimer(void)
{
    portENTER_CRITICAL_ISR(&timerMux);
    interruptCounter++;
    portEXIT_CRITICAL_ISR(&timerMux);
}

```

Certifico que o trabalho de conclusão de curso intitulado **“Implantação de um Sistema de Automação Predial Utilizando Plataforma em Nuvem”** de autoria do aluno **Guilherme Rafael Pereira da Silva**, foi aprovado sem recomendações de alteração pela banca examinadora e que estou de acordo com a versão final do trabalho.

A handwritten signature in blue ink, consisting of a series of loops and a long horizontal stroke, positioned above a horizontal line.

Diógenes Viegas Mendes Ferreira
Orientador

Ouro Preto, 28 de novembro de 2018.