

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE ESPECIALIZAÇÃO EM INTERNET DAS COISAS

KONRADO BIENERT

PLATAFORMA DE CONTROLE DE INTENSIDADE ILUMINOSA

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA

2018

KONRADO BIENERT

PLATAFORMA DE CONTROLE DE INTENSIDADE ILUMINOSA

Monografia de Especialização, apresentada ao Curso de Especialização em Internet das Coisas, do Departamento Acadêmico de Eletrônica – DAELN, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. Dr. Glauber Gomes de Oliveira Brante

CURITIBA

2018



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Curitiba

Diretoria de Pesquisa e Pós-graduação
Departamento Acadêmico de Eletrônica
Curso de Especialização em Internet das Coisas



TERMO DE APROVAÇÃO

PLATAFORMA DE CONTROLE DE INTENSIDADE ILUMINOSA

por

KONRADO BIENERT

Esta Monografia foi apresentada em 23 de Novembro de 2018 como requisito parcial para a obtenção do título de Especialista em Internet das Coisas. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Glauber Gomes de Oliveira Brante
Orientador

Prof. M.Sc. Danillo Leal Belmonte
Membro titular

Prof. M.Sc. Omero Francisco Bertol
Membro titular

- A Folha de Aprovação assinada encontra-se arquivada na Secretaria Acadêmica -

AGRADECIMENTOS

Agradeço a todos os professores pelo conhecimento adquirido, amigos de turma, coordenação e direção da instituição pela qual me guiaram durante a realização deste curso.

Agradecimento também ao meu colega de turma Luiz Denis de Lara, pela parceria no projeto e possível realização deste projeto.

RESUMO

BIENERT, Konrado. **Plataforma de controle de intensidade iluminosa**. 2018. 38 p. Monografia de Especialização em Internet das Coisas, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

Quase todas as nossas atividades diárias envolvem a energia elétrica em alguma forma, direta ou indiretamente. A mais difundida forma de usar-se a energia elétrica é a iluminação residencial e industrial. Em muitos locais, a qualidade e quantidade luminosa não é correta e/ou desnecessária acarretando um desperdício de energia elétrica, exemplo uma sala de reuniões com lâmpadas acessas, mas sem nenhuma pessoa. Nesse contexto, o objetivo desse trabalho é desenvolver um sistema que contenha um painel de controle luminoso e gráficos apresentando o consumo gerado no local com a possibilidade de gerar economia elétrica. Para o desenvolvimento da aplicação foi utilizado o protocolo MQTT para comunicação e a linguagem de programação web Angular.

Palavras-chave: Lâmpada. Iluminação. Controle. Instrumentação.

ABSTRACT

BIENERT, Konrado. **Lighting intensity control platform**. 2018. 38 p. Monografia de Especialização em Internet das Coisas, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

Almost all our daily activities involve electricity in some form, directly or indirectly. The most widespread way to use electric power is residential and industrial lighting. In many places, the quality and quantity of light is not correct and / or unnecessary, resulting in a waste of electricity, for example a meeting room with access lamps, but without any person. In this context, the objective of this work is to develop a system that contains a luminous control panel and graphs showing the consumption generated in the place with the possibility of generating electric economy. For the development of the application, the MQTT protocol for communication and the Angular web programming language were used.

Keywords: Lamp. Lighting. Control. Instrumentation.

LISTA DE ILUSTRAÇÕES

Figura 1 - Diagrama em blocos da solução parcial	16
Figura 2 - Fluxo aplicação angular	19
Figura 3 - Módulo ESP8266 ESP-01	21
Figura 4 - Esquema elétrico de um relê	22
Figura 5 - NodeMCU definição de pinos	23
Figura 6 - Nebular Framework Light Theme.....	24
Figura 7 - Tabela SQL de Logs	28
Figura 8 - Painel de lâmpadas na aplicação web	31
Figura 9 - Gráfico de economia na aplicação web	31
Figura 10 - Gráfico economia do nó número 1	32
Figura 11 - Client MQTT inscrito no tópico genérico	33
Diagrama 1 - Construção do nó	26
Diagrama 2 - Conexão nós com MQTT broker.....	27
Diagrama 3 - Conexão banco de dados com MQTT broker	28
Diagrama 4 - Conexão aplicação web com MQTT broker.....	29
Diagrama 5 - Conexão aplicação web com banco de dados	30
Quadro 1 - Descrições e funções dos pinos do ESP-01	21

LISTA DE SIGLAS

BD	Banco de Dados
CNTP	Condições Normais de Temperatura e Pressão
CSS	<i>Cascading Style Sheets</i>
SQL	<i>Structured Query Language</i>
DLL	<i>Dynamic-link library</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
PWM	<i>Pulse Width Modulation</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
WirelessHART	<i>Wireless Highway Addressable Remote Transducer</i>

LISTA DE ACRÔNIMOS

Cloud	Nuvem computacional
SQL	Linguagem para banco de dados
Ethernet	Arquitetura de interconexão para redes locais
IoT	<i>Internet of Things</i>
JavaScript	Linguagem de script para páginas web
MQTT	<i>Message Queuing Telemetry Transport</i>
Web	<i>World Wide Web</i>
Wi-Fi	<i>Wireless Fidelity</i>

SUMÁRIO

1 INTRODUÇÃO	11
1.1 PROBLEMA	12
1.2 OBJETIVOS.....	13
1.2.1 Objetivo Geral	13
1.2.2 Objetivos Específicos	13
1.3 JUSTIFICATIVA.....	13
1.4 ESTRUTURA DO TRABALHO	14
2 METODOLOGIA	15
2.1 FUNDAMENTAÇÃO TEÓRICA	16
2.1.1 Internet das Coisas	16
2.1.2 Aplicação Web	16
2.1.3 Banco de Dados.....	17
2.1.4 Microcontrolador	18
2.2 TECNOLOGIAS UTILIZADAS	18
2.2.1 Angular.....	18
2.2.2 Mqtt	19
2.2.3 Módulo Esp8266	20
2.2.4 Sensor de Corrente Elétrica	22
2.2.5 Relê.....	22
2.2.6 Nodemcu.....	23
2.2.7 Framework Nebular Ngx	23
3 PROJETO	25
3.1 COMPONENTES.....	25
3.2 DIAGRAMAS	25
3.2.1 Diagrama 1: Criação dos Nós	26
3.2.2 Diagrama 2: Nó Conectado ao Mqtt Broker	26
3.2.3 Diagrama 3: Broker Mqtt Conectado ao Banco de Dados.....	28
3.2.4 Diagrama 4: Broker Mqtt Ligado a Aplicação Web.....	29
3.2.5 Diagrama 5: Aplicação Web Ligado ao Banco de Dados Web	29
3.3 APLICAÇÃO WEB	30
4 RESULTADOS.....	32
4.1 APLICAÇÃO WEB	32
4.2 BROKER MQTT.....	33
4.3 NÓ	33
5 TRABALHOS FUTUROS.....	34
5.1 MICRONTROLADOR	34
5.2 BROKER MQTT.....	34
5.3 BANDO DE DADOS	35

6 CONSIDERAÇÕES FINAIS.....	36
REFERÊNCIAS.....	37

1 INTRODUÇÃO

A iluminação consome 16% da energia elétrica gasta no país. Maior parte deste consumo, ocorre durante o período de pico do sistema, isto é, das 18 às 22 horas, durante a maior solicitação de energia elétrica. Em média, cada estabelecimentos comerciais gastam cerca de 44% da sua eletricidade com iluminação, mas essa participação pode chegar a 70% no caso dos shoppings e das grandes lojas de departamentos (CEPA, 2018).

Diante deste cenário de uso intensivo, uma solução se oferece: a economia ou conservação de energia destinada a iluminação, através do desenvolvimento e difusão de novas tecnologia mais eficientes.

Nos últimos anos, a automação de iluminação residencial e comercial tem oferecido qualidades como o gerenciamento inteligente de cargas elétricas, área a qual vêm ganhando destaque internacional como tecnologia de sustentabilidade e de conforto no espacial. Como resultado, este serviço foi o mais demandado por integradores brasileiros em 2016 (NEOCONTROL, 2018).

Sem dúvidas, a maior vantagem da automação de iluminação residencial é sua lucratividade no longo prazo, ou retorno de investimento. Aqui, está sendo indicado a sua capacidade de incorporação progressiva de novos subsistemas que a longo prazo será através da eficiência do consumo energético. Estima-se que os sistemas integrados podem recuperar de 10 a 30% do valor inicial, em uma perspectiva não conservadora dos dados. Sobretudo, por que estes fatores dependerão do tipo de automação utilizado, limitações do projeto elétrico e quantidade de integrações.

As soluções têm se comprovado cada vez mais viáveis e lucrativos ao longo do tempo, pois o custo de automação está caindo. No cenário brasileiro, a automação sem fio representa menos de 1% do valor inicial de obra. A medida que populariza seu uso para o público geral no Brasil tende-se a baratear este custo, pois nem todas as marcas e protocolos de automação estão presentes no mercado (NEOCONTROL, 2018).

Nos próximos anos, com a preservação do meio-ambiente ainda mais em pauta, os sistemas de iluminação inteligente se tornarão essenciais na construção civil. Pela sua fácil inserção na arquitetura sustentável, as soluções terão uma

demanda crescente de projetos no longo prazo. Por um lado, otimizando o consumo de recursos e por outro, mantendo a capacidade de personalização do projeto.

Saindo do campo da criatividade, o uso da automação de iluminação residencial também auxilia na melhoria da qualidade de vida do usuário. Quando pensamos projetos de iluminação, não podemos deixar de lado sua influência sobre os moradores da residência. A luz dos espaços tem importância direta na qualidade de vida dos habitantes, afetando a performance e tolerância às suas atividades cotidianas da casa. São diversas ações desempenhadas ao longo do dia, a maior parte delas expostas à luz artificial. Com a simples utilização de periféricos para dimerização e iluminação econômica, é possível oferecer uma solução inovadora.

O presente trabalho trata essa questão, ou seja, do controle luminoso de uma residência por dimerização, intensidade de brilho. O objetivo deste projeto é habilitar um usuário a atuar de forma imediata e remota no acionamento e desligamento de dispositivos luminosos através de um painel de controle virtual, onde será possível saber qual objeto o mesmo está atuando. Utilizando um banco de dados que armazene estas informações de forma temporal é possível gerar um relatório de econômico, o qual também irá ficar à disposição de consulta ao usuário. No comércio já existem produtos que fazem a medição e controle de dispositivos de iluminação, porém se apresentam a elevados custos e nem sempre adaptados a economia energética, o foco deste trabalho.

1.1 PROBLEMA

Em estabelecimentos comerciais ou residentes é muito comum o desperdício de energia devido a lâmpadas acessas sem nenhum propósito. Dependendo do tempo de duração desta lâmpada ligada há um impacto direto na conta de luz, fora o desperdício de energia. Tanto no âmbito residencial quanto industrial, este controle é feito de forma manual, o qual expõe qualquer tipo de estabelecimento a falhas humanas.

Assim, constatou-se que na prática é necessário o controle de iluminação do ambiente através de algum sistema automatizado e digital que possa, em tempo real, fazer as alterações no nível luminoso ou eliminá-lo onde não há a necessidade do mesmo.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

A proposta é a construção de um protótipo capaz de inferir no estado luminoso de uma ou várias lâmpadas em um ambiente, seja ele comercial ou residencial. E que concentrem as informações coletadas ou inferidas em um banco de dados para que se possa processar o tratamento matemático específico para o controle de custo energético.

1.2.2 Objetivos Específicos

Para atender ao objetivo geral neste trabalho de conclusão de curso os seguintes objetivos específicos serão abordados:

- Construir um protótipo que tenha a capacidade de dimerização de uma lâmpada bem como o acionamento ou desligamento da mesma;
- Implementar a comunicação dos nós com *broker MQTT* por meio de internet sem fio (Wi-Fi, do inglês *Wireless Fidelity*);
- Utilizar servidor para a implementação do banco de dados e *host* de uma aplicação web;
- Possibilitar a análise dos dados obtidos em tempo real através de uma aplicação web.

1.3 JUSTIFICATIVA

Visto a importância de economia de energia elétrica, é possível visualizar várias formas de se obter economia energética neste meio. Uma delas é evitar o desperdício e o gasto excessivo de energia elétrica em forma de luz.

Exposto isso, conclui-se que esse estudo é enfatizado para uso comercial e residencial, mas pode ser utilizado em outras áreas como industrial. A metodologia de desenvolvimento deste projeto pode ser aplicada a qualquer área que necessite um controle de iluminação.

1.4 ESTRUTURA DO TRABALHO

O trabalho tem a seguinte estrutura:

- Capítulo 1 - Introdução: são apresentados o problema, os objetivos do projeto, a justificativa e a estrutura geral do trabalho.
- Capítulo 2 - Metodologia: serão apresentados a divisão dos blocos do sistema e os aspectos de interconexão entre esses blocos. As tecnologias e dispositivos utilizados no projeto serão esmiuçadas em suas características fundamentais.
- Capítulo 3 - Projeto: serão apresentados os requisitos e diagramas dos circuitos que serão utilizados durante todo o projeto. Também serão apresentadas detalhadamente como foi a construção de cada bloco do sistema, as interfaces de entrada e saída, aspectos de interconexão e testes de cada bloco.
- Capítulo 4 - Resultados: serão apresentados os testes com a plataforma implementada e uma análise crítica dos resultados.
- Capítulo 5 - Trabalhos futuros: serão expostos aspectos industriais da transformação do protótipo em produto comercial.
- Capítulo 6 - Considerações finais: será feito um balanço geral do trabalho.

2 METODOLOGIA

Para o desenvolvimento do protótipo da plataforma SMARTLIGHT foram adotadas algumas premissas fundamentais, a saber: construir um produto que atenda às necessidades do cliente e utilizando o mínimo de recursos possíveis.

Esta solução é formada pelos seguintes componentes:

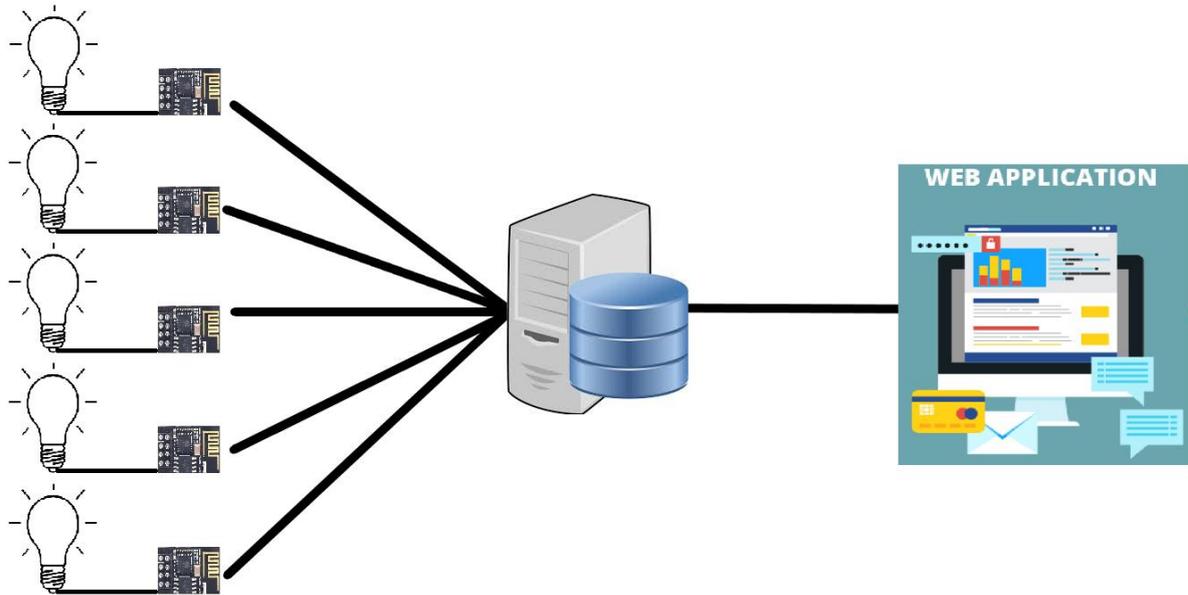
- **Nó:** é o bloco que fica em contato direto com o ambiente no qual se quer controlar os níveis de iluminação. Trata-se de um sistema que contém um microcontrolador atuando em conjunto com o sensor de gás. O nó fica latente a maior parte do tempo, a não ser que haja um comando da aplicação *web*. Após um determinado tempo o microcontrolador sai do estado de latência, faz a aferição do estado da lâmpada bem como seu nível luminoso e volta à latência. Esse ciclo é repetido por tempo indeterminado;
- **Servidor:** para cada ambiente é necessário um concentrador de informações, algum elemento que colete as informações de cada nó e envie para outro elemento que irá armazenar essas informações e deixará os dados no formato necessário para o devido tratamento matemático;
- **Banco de dados:** elemento que armazena as informações de todos nós temporalmente.
- **Aplicação *web*:** Informações ficam à do usuário que monitora o sistema e o mesmo toma decisões de acionamento ou desligamento de equipamentos luminosos de acordo com sua vontade.

A comunicação entre os nós e o servidor será sem fio, através de módulos Wi-Fi, de forma altamente escalável, sem sistemas redundantes. O servidor será responsável por registrar dados no banco de dados.

A interface com o usuário final será feita através de uma aplicação *web* que fornece painéis onde o usuário pode visualizar os gráficos de consumo de energia e um painel para atuar em cada dispositivo luminoso.

A solução proposta é ilustrada na Figura 1, onde se pode observar as conexões entre as lâmpadas e os blocos de sensor, bem como os blocos sensor/microcontrolador com o servidor e Banco de Dados (BD) e pôr fim a conexão entre servidor e aplicação *web*.

Figura 1 - Diagrama em blocos da solução parcial



Fonte: Autoria própria, utilizando o software Paint3D.

2.1 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentadas a introdução teórica e o detalhamento dos aspectos fundamentais de cada tecnologia que será utilizada.

2.1.1 Internet das Coisas

De forma simplificada, esse conceito pode ser entendido como um conjunto de tecnologias que possibilitem a interligação de equipamentos de forma que a troca de informações e ações ocorra através da internet. O mercado de IoT (*Internet of Things*, ou Internet das Coisas) tende a crescer muito nos próximos anos com o objetivo de simplificar e facilitar os processos diários das pessoas, sejam eles em casa, na rua ou na empresa. Assim criando de forma direta e indiretamente novas possibilidades de negócios (OLIVEIRA, 2017).

2.1.2 Aplicação Web

O nome *web* vem do termo mais amplo *World Wide Web* (WWW) que representa um sistema de informações mundial distribuído, onde as informações são "ligadas" umas às outras por links de hipertexto. Desenvolvimentos mais recentes

desta tecnologia têm permitido distribuir, junto com as informações, objetos e programas que realizam atividades junto ao cliente consumidor da informação.

Em computação, uma aplicação web refere-se a um ou mais sistemas digitais projetados para utilização através de um navegador, através da internet. Aplicativos desenvolvidos utilizando tecnologias web HTML, JavaScript e CSS. HTML é uma das linguagens utilizadas para desenvolver websites. O acrônimo HTML vem do inglês e significa *Hypertext Markup Language* (ou em português, Linguagem de Marcação de Hipertexto). CSS é uma linguagem de folha de estilos, que tem o papel de tornar uma página apresentável na web, relacionada diretamente com o design e aparência. E por fim, JavaScript é uma linguagem de programação interpretada originalmente implementada como parte dos navegadores web para que scripts pudessem ser executados do lado do cliente e interagissem com o usuário sem a necessidade deste script passar pelo servidor (UNILAB, 2018).

Aplicações web podem ser executados a partir de um servidor HTTP (Web Host) ou localmente, no dispositivo do usuário. O HTTP usa um modelo de solicitações e respostas. Uma solicitação ocorre quando o usuário faz uma solicitação HTTP e o servidor web devolve uma resposta HTTP, sendo que o browser verifica como tratar esse conteúdo. Se a resposta que vem do servidor for uma página HTML, então é inserido na resposta HTTP (PLURALSIGHT, 2018).

2.1.3 Banco de Dados

Um banco de dados é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico, ou seja, sempre que for possível agrupar informações que se relacionam e tratam de um mesmo assunto, pode se dizer que existe um banco de dados (KORTH; SILBERSCHATZ, 1994).

Para armazenar um dado, é necessário criar tabelas, dentro das tabelas são criadas colunas, onde as informações são armazenadas. Para os dados da base de dados ficarem organizados, devem ser criadas tabelas que não misturem informações.

2.1.4 Microcontrolador

Os microcontroladores são microprocessadores que podem ser programados para funções específicas. A estrutura interna de um microcontrolador apresenta um processador, bem como circuitos de memória e periféricos de entrada e saída. O seu consumo em geral é relativamente pequeno, normalmente na casa dos miliwatts e possuem geralmente habilidade para entrar em modo de espera aguardando por uma interrupção ou evento externo. Estas habilidades tornam microcontroladores altamente viáveis para a implementação de um projeto de Internet das Coisas.

Em geral, microcontroladores são utilizados para controlar circuitos e, por isso, são comumente encontrados dentro de outros dispositivos, sendo conhecidos como "controladores embutidos".

2.2 TECNOLOGIAS UTILIZADAS

2.2.1 Angular

Angular é uma plataforma e *framework* para construção da interface de aplicações usando HTML, CSS e, principalmente, JavaScript, criada pelos desenvolvedores da Google. Para criar aplicações com Angular, pode-se usar as linguagens TypeScript, JavaScript e Dart.

O TypeScript é uma linguagem de programação criada pela Microsoft. Ela basicamente é um *superset* do JavaScript, ou seja, faz tudo o que o JavaScript faz e ainda mais alguns recursos. Os navegadores (*browsers*) não conseguem executar código TypeScript, por isso, o que acontece é que os desenvolvedores escrevem códigos em TypeScript e esses códigos são "transpilados" para JavaScript.

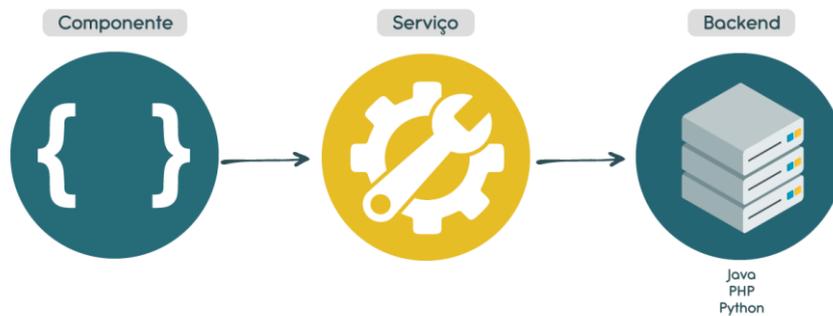
Transpilação é parecido com a compilação, mas ao invés de gerar um código de mais baixo nível, é gerado o código em outra linguagem, ou na mesma linguagem, só que com o objetivo de deixar compatível com o ambiente que vai ser executado.

No caso de aplicações Angular desenvolvidas com TypeScript, a transpilação é o mesmo que gerar código *JavaScript* a partir de código TypeScript. Esse por sua vez, compreendido por navegadores.

Uma aplicação Angular é baseada em componentes e serviços, como apresentado na Figura 2. Com estes, pode se encapsular comportamentos e regras

da interface, o que torna a criação de aplicações algo mais simples uma vez que estes componentes podem ser reutilizados em outras partes da mesma aplicação. Um componente é basicamente composto por três itens: a) Template HTML, b) CSS, e c) uma classe que o gerencia.

Figura 2 - Fluxo aplicação angular



Fonte: Algaworks (2018).

Serviços são criados para concentrar todas as regras de negócio da aplicação. Com isso, pode-se deixar os componentes focados em gerenciar partes da tela que o usuário visualizará e os serviços focados em código com um outro propósito específico, como por exemplo, interagir com um banco de dados.

2.2.2 Mqtt

A conexão com a Internet é um requisito para dispositivos considerados de Internet das Coisas. Esta permite que os dispositivos trabalhem entre si. O protocolo de rede subjacente da Internet é o TCP/IP.

Desenvolvido com base na pilha TCP/IP, o *Message Queue Telemetry Transport* (MQTT) tornou-se o padrão para comunicações de IoT.

O protocolo MQTT define dois tipos de entidades na rede: um *broker* e inúmeros clientes. O *broker* é um servidor que recebe todas as mensagens dos clientes e, em seguida, roteia essas mensagens para os clientes de destino relevantes. Um cliente é um dispositivo capaz de interagir com o *broker* e receber ou enviar mensagens.

Como as mensagens do MQTT são organizadas por tópicos, aplicações tem a flexibilidade de especificar que determinados clientes somente podem interagir com determinadas mensagens, isto se dá por tópicos descendentes.

A conexão entre o *broker* e o cliente acontece da seguinte forma: a) Primeiro, o cliente conecta-se ao *broker* enviando uma mensagem CONNECT. A mensagem CONNECT pede para estabelecer uma conexão do cliente com o *broker*. b) Depois que uma conexão for estabelecida, o cliente poderá enviar uma ou mais mensagens SUBSCRIBE ao *broker* para indicar que ele receberá mensagens do *broker* de determinados tópicos.

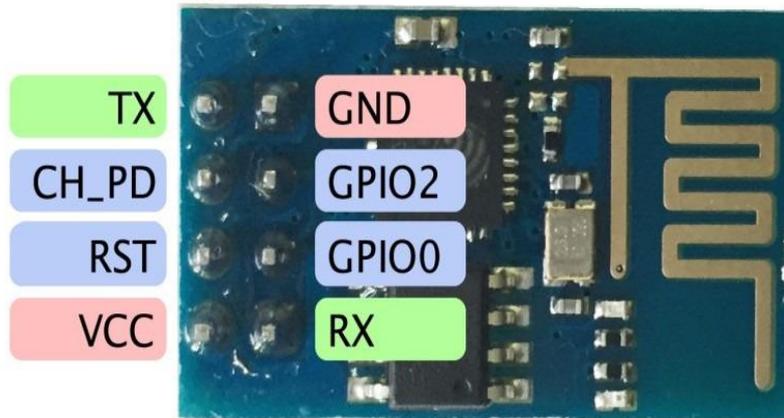
Uma vez que a inscrição em um tópico é realizada, o cliente pode enviar mensagens PUBLISH ao *broker*. A mensagem contém um tópico e uma carga de dados. Em seguida, o *broker* encaminha a mensagem a todos os clientes que assinam esse tópico (YUAN, 2018).

2.2.3 Módulo Esp8266

O microcontrolador ESP8266, produzido pela fabricante chinesa Espressif, é um microcontrolador de 32 bits. Se apresenta como solução para quem busca um microcontrolador com módulo WiFi embarcado. O processamento da pilha de protocolos Wi-Fi usa cerca de um quinto da capacidade de processamento desse processador. Com isso, quatro quintos dessa capacidade podem ser utilizadas em aplicações customizáveis. Levando isto em consideração para o propósito deste trabalho, não há necessidade de um outro microcontrolador para ser o atuador em dispositivos luminosos, o próprio ESP8266 é suficiente para esta função.

O módulo mais popular é o ESP-01 (Figura 3). Este possui apenas o módulo de memória externa entre 512kB e 1MB, sendo mais que suficiente para a maioria dos casos reais. Possui apenas 8 pinos incluindo os pinos de alimentação e de comunicação serial. Em termos de embutir códigos customizáveis, é necessário um adaptador USB utilizando os pinos TX e RX, junto a uma interface de compilação de código em formato LUA, este formato necessário pelo fabricante do microcontrolador.

Figura 3 - Módulo ESP8266 ESP-01



Fonte: Alves (2016).

As descrições e as funções dos pinos do módulo ESP8266 ESP-01 (Figura 3) são apresentadas no Quadro 1.

Quadro 1 - Descrições e funções dos pinos do ESP-01

Pino	Descrição	Função
8	VCC	Tensão de alimentação 3,3V. Módulo consome até 300mA.
1	GND	Sinal de GND.
2	GPIO2	I/O que pode ser controlada pelo firmware.
3	GPIO0	Pode ser controlado pelo firmware e deve ser colocado em nível baixo (GND) para modo update ou em nível alto para operação normal.
6	CH_PD	Sinal de habilitação do chip (chip enable), usado para gravação de firmware ou atualização. Deve ser mantido em nível ALTO para operação normal.
7	RESET	Sinal de Reset/Restart acionado em nível baixo (GND).
4	Rx (Transmissão Serial - Recebe)	Sinal de Rx do módulo, a ser conectado no Tx do microcontrolador.
5	Tx (Transmissão Serial - Transmite)	Sinal de Tx do módulo, a ser conectado no Rx do microcontrolador.

Fonte: Curvello (2015).

O ESP8266 possui ainda um recurso denominado atualização *Over The Air* (OTA), que possibilita que seu firmware seja atualizado pela rede Wi-Fi, em vez de ser atualizado exclusivamente pela porta USB/Serial. Essa característica é muito importante na atualização de software para correção de erros, já depois de implantado (OLIVEIRA, 2017).

É possível que os dispositivos sejam atualizados periodicamente, assim como ocorre com os sistemas operacionais modernos. Apesar de que esta funcionalidade não foi implementada ou testada neste trabalho.

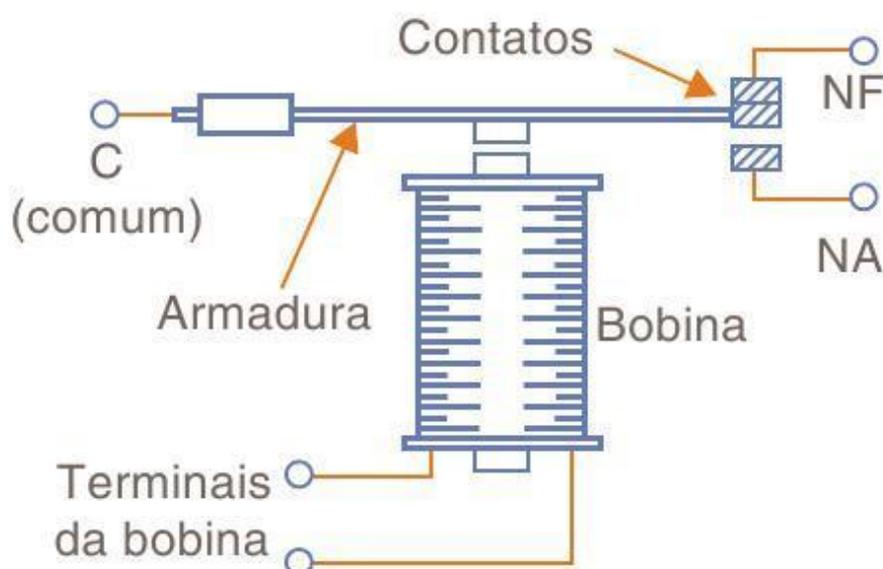
2.2.4 Sensor de Corrente Elétrica

O sensor de corrente SCT-013-000 é sensor capa de medir corrente AC até 100 Amperes e que não é invasivo, ou seja, não há a necessidade de contato físico com o meio em que a corrente elétrica está sendo transportado para haver aferição. Muito usado em projetos com Arduino de automação residencial como proteção de motores AC, iluminação e dentre outros.

2.2.5 Relê

O relê é um dispositivo eletromecânico que permite o acionamento de cargas elétricas a partir de um circuito de comando de baixa potência, como, por exemplo, um sinal digital de um microcontrolador. Um relê é constituído basicamente por uma bobina e contatos. Possuem contatos NA (Normalmente Aberto) ou NF (Normalmente Fechado), dependendo do modelo do relê escolhido. Esta nomenclatura se refere principalmente ao estado de repouso da bobina do relê. Quando o relê está desligado, o contato NA está aberto e o NF fechado, porém quando é acionada a bobina do relê os contatos mudam de estado, ou seja, o contato NA é fechado e o NF é aberto. Esse estado permanece enquanto a bobina estiver acionada, e, quando ela é desligada, os contatos retornam para o estado de repouso. A Figura 4, representa a ideia de um relê.

Figura 4 - Esquema elétrico de um relê

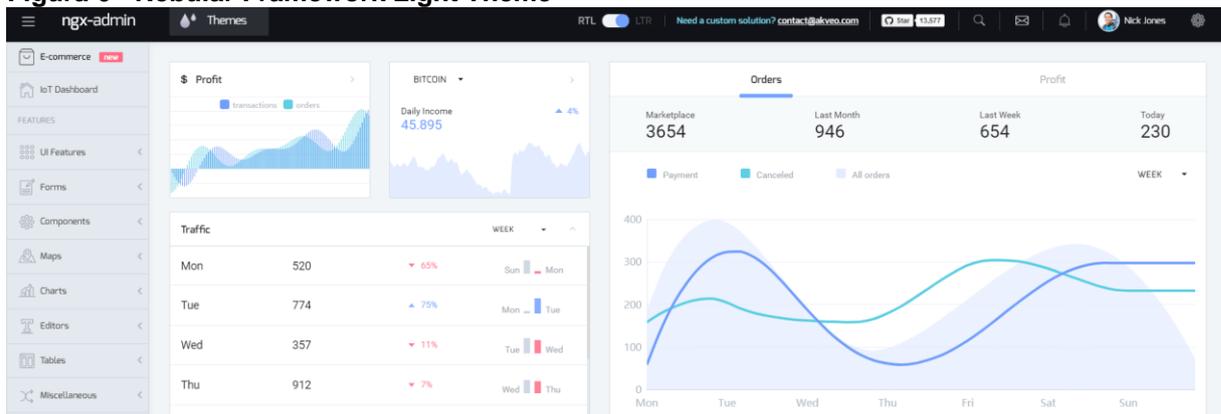


Fonte: Mara (2012).

rápida para que o desenvolvedor consiga se preocupar com a lógica de negócio ou produto.

Com o objetivo de evitar procurar bibliotecas específicas, passar por inúmeros processos de instalação. É com este motivo, a escolha do framework Nebular, apresentado na Figura 6, que reuni os módulos e bibliotecas mais úteis e junta-os em um padrão de interface gráfica, criando um excelente kit de ferramentas para facilitar a configuração.

Figura 6 - Nebular Framework Light Theme



Fonte: Github (2018).

3 PROJETO

Este capítulo descreve e detalha o processo de construção de todos os blocos que constituem o protótipo do projeto. Em cada etapa são apresentadas as entradas, saídas de cada bloco separadamente.

3.1 COMPONENTES

Para a execução do projeto foram necessários os seguintes componentes:

- NodeMCU: 5 unidades;
- Fonte 3.3V/1A: 5 unidades;
- Led verde: 5 unidades;
- Protoboard: 5 unidades;
- Modulo Relé: 5 unidades;
- Lâmpadas de 110Volts: 5 unidades;
- Link ADSL com modem/roteador: 1 unidade;
- Computador pessoal com Windows 10: 1 unidade.

Além disso, um computador pessoal, os protoboards e as fontes foram utilizados ao longo de todo o projeto. O Arduino Uno R3 e Arduino MEGA 2560 foram utilizados na fase inicial de prototipação e os Arduino Pro Mini foram utilizados para a implementação dos “nós” completos. A placa FDTI 232 foi utilizada para gravar o código do programa, via IDE Arduino, nas placas Arduino Pro Mini.

3.2 DIAGRAMAS

No decorrer do projeto foram desenvolvidos os componentes ilustrados em 5 (cinco) diagramas: 1) Criação de nós, 2) Nó conectado ao MQTT Broker, 3) Broker MQTT conectado ao banco de dados, 4) Broker MQTT ligado a aplicação web, e 5) Aplicação web ligado ao banco de dados web.

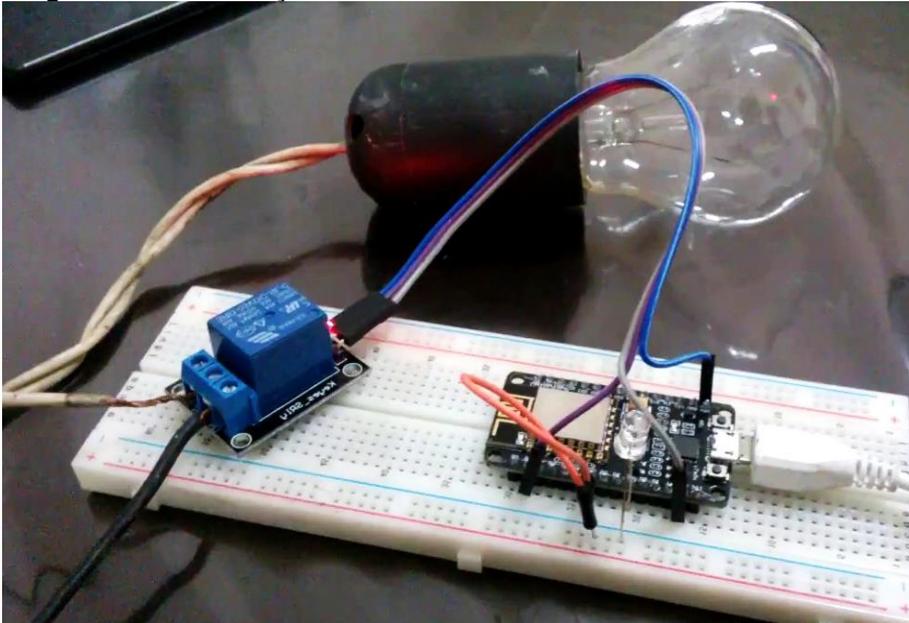
3.2.1 Diagrama 1: Criação dos Nós

O Diagrama 1 ilustra a construção do nó na prática. O NodeMCU está conectado a uma fonte por meio de um cabo serial o qual o alimenta com 5 Volts. O NodeMCU contém o ESP8266 embarcado conectado junto ao módulo relé atrelado a uma lâmpada incandescente de 110 Volts.

Para o desenvolvimento do projeto foi utilizando a linguagem Python, suportada pelo módulo ESP8266. Durante o processo de desenvolvimento requer-se a nomeação do módulo durante a primeira inicialização, após a configuração do SSID e senha da rede.

Com a rede configurada o nó inicia sua rotina de enviar dados ao *broker MQTT* a cada minuto para futuramente ser tratado no painel de gerenciamento. Um mecanismo de interrupção é acionado quando uma mensagem MQTT chega a este nó, assim assegurando que a lâmpada terá seu estado alterado.

Diagrama 1 - Construção do nó



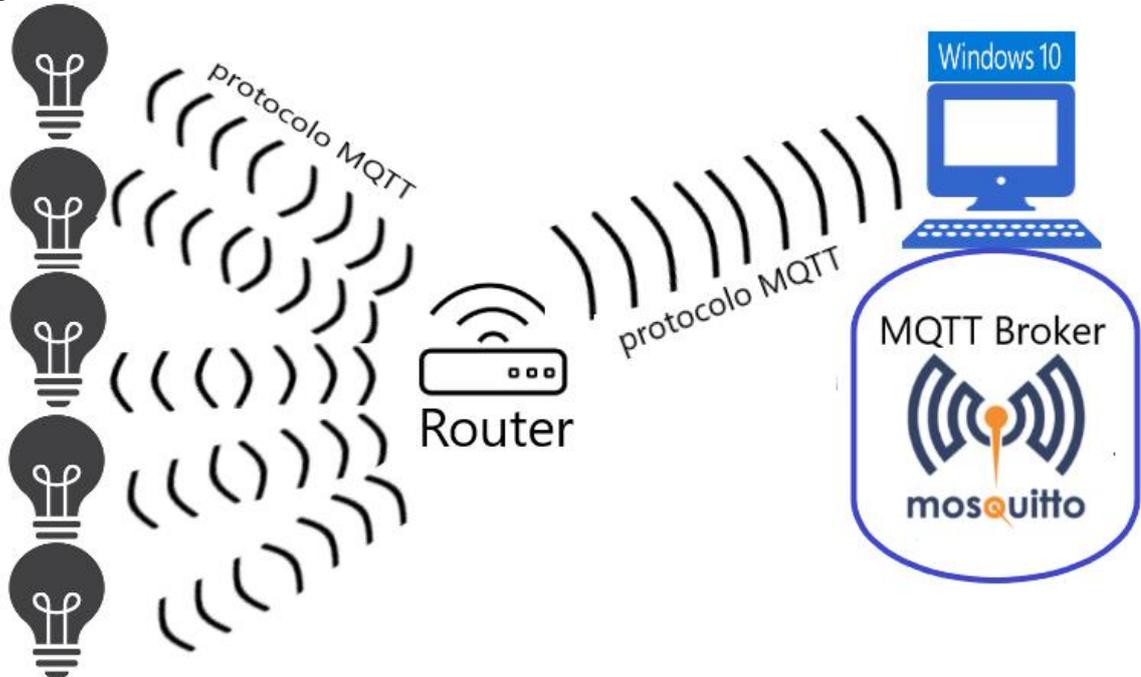
Fonte: Autoria própria.

3.2.2 Diagrama 2: Nó Conectado ao Mqtt Broker

O Diagrama 2, ilustra como o nó consegue se conectar com o *broker MQTT*. As lâmpadas, por meio do módulo Wi-Fi, conectam-se ao roteador Wi-Fi, este por sua vez faz a interface com o servidor que se encontra o *broker MQTT*. Importante

mencionar que o *broker MQTT* utilizado neste projeto foi o *mosquitto*. O mesmo foi instalado em um computador utilizando Windows 10 (64 bits).

Diagrama 2 - Conexão nós com MQTT broker



Fonte: Autoria própria, utilizando o software Paint3D.

Para que este *broker* funcione de forma correta alguns passos foram tomados. Primeiramente, o *mosquitto* opera em sistemas 32 bits logo teve que ser instalado o OpenSSL de 32 bits. O qual contém uma implementação de código aberto dos protocolos de criptografia SSL e TLS, necessários para o funcionamento correto do protocolo MQTT. Logo após isto foi necessário o download da *Dynamic-link library* (DLL) pthreadVC2, esta que tem por função instalar registros em sistemas 32 bits, necessário para registros do *broker* no sistema operacional.

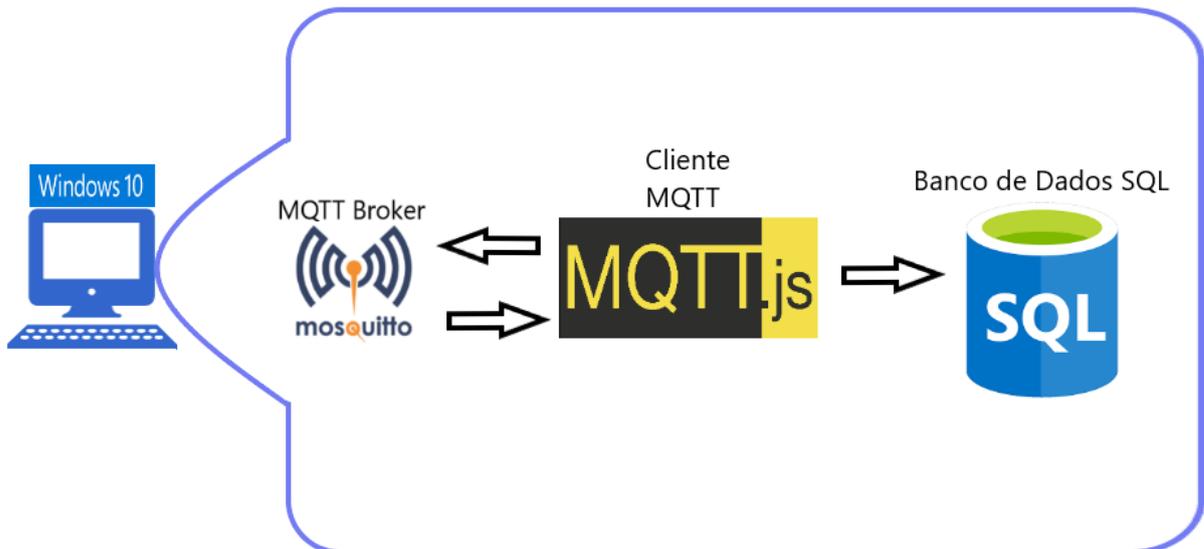
Com estas duas correções o *broker MQTT* está preparado para receber informações do nó. A troca de informações acontece da seguinte forma:

Basicamente, o nó envia uma mensagem CONNECT ao *broker MQTT*. O cliente recebe um CONNACK do *broker*, indicando que a conexão foi estabelecida com sucesso. A partir desse momento cada lâmpada pode enviar o seu estado ou dimerização para o tópico correspondente. Exemplo: Lâmpada 1 envia “1” no tópico indicando “mqtt/ID1” que sua lâmpada esta acesa.

3.2.3 Diagrama 3: Broker Mqtt Conectado ao Banco de Dados

O Diagrama 3, ilustra como o banco de dados é alimentado a partir de um cliente escrito em *javascript* e utilizando a biblioteca PAHO MQTT. Restritamente não há conexão fora do servidor para registros no banco de dados. Sendo assim, o cliente MQTT é ligado ao *broker* para captura de todas as informações. Este cliente faz um CONNECT e logo em seguida um SUBSCRIBE no tópic “mqtt/#”, o qual permite por meio deste *wildcard* receber todas mensagens geradas pela aplicação. Assim que uma informação é capturada, há construído uma lógica dentro deste *javascript* que habilita a gravação de dados por meio do comando SQL INSERT.

Diagrama 3 - Conexão banco de dados com MQTT broker



Fonte: Autoria própria, utilizando o software Paint3D.

A estrutura de logs do banco de dados é apresentada na Figura 7. Essa estrutura é necessária pois a aplicação necessita de dados históricos para apresentar a quantidade de energia economizada.

Figura 7 - Tabela SQL de Logs

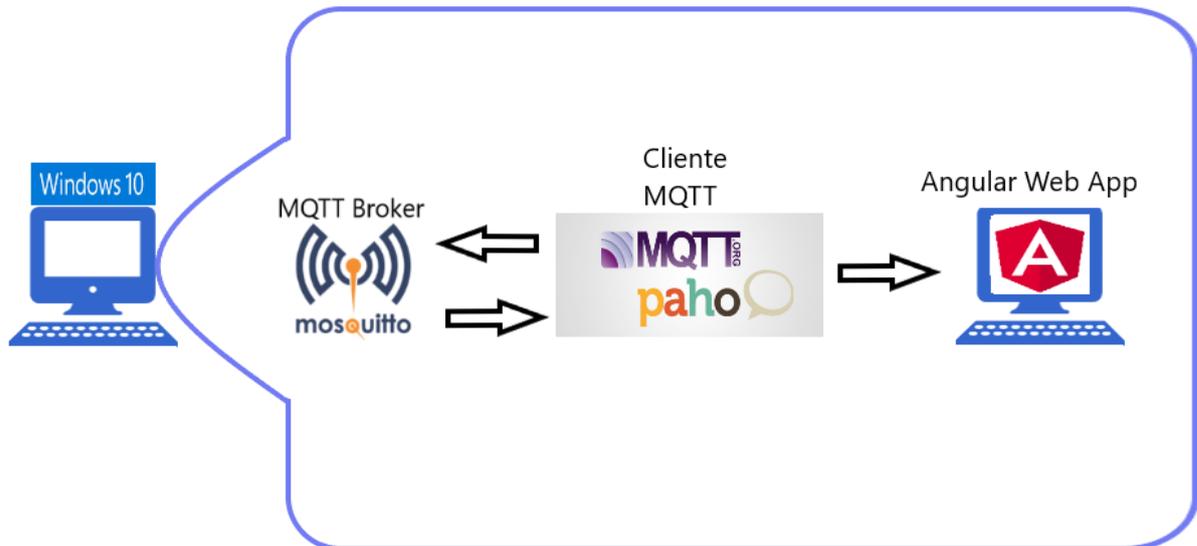
id_tb_log	status	dimmer	consumo	grupo	heartbeat
1	1	100	1	1	2018-10-15 00:00:00

Fonte: Autoria própria, utilizando o software Paint3D.

3.2.4 Diagrama 4: Broker Mqtt Ligado a Aplicação Web

O Diagrama 4, encapsula o funcionamento da aplicação web ligada ao *broker* MQTT. Esta conexão se faz necessária pois a aplicação deve enviar mensagens no protocolo MQTT para os nós com a finalidade de mudar o estado dos mesmos. Assim, a aplicação web é responsável por ser o controlador da plataforma, a parte que é considerada como atuador.

Diagrama 4 - Conexão aplicação web com MQTT broker



Fonte: Autoria própria, utilizando o software Paint3D.

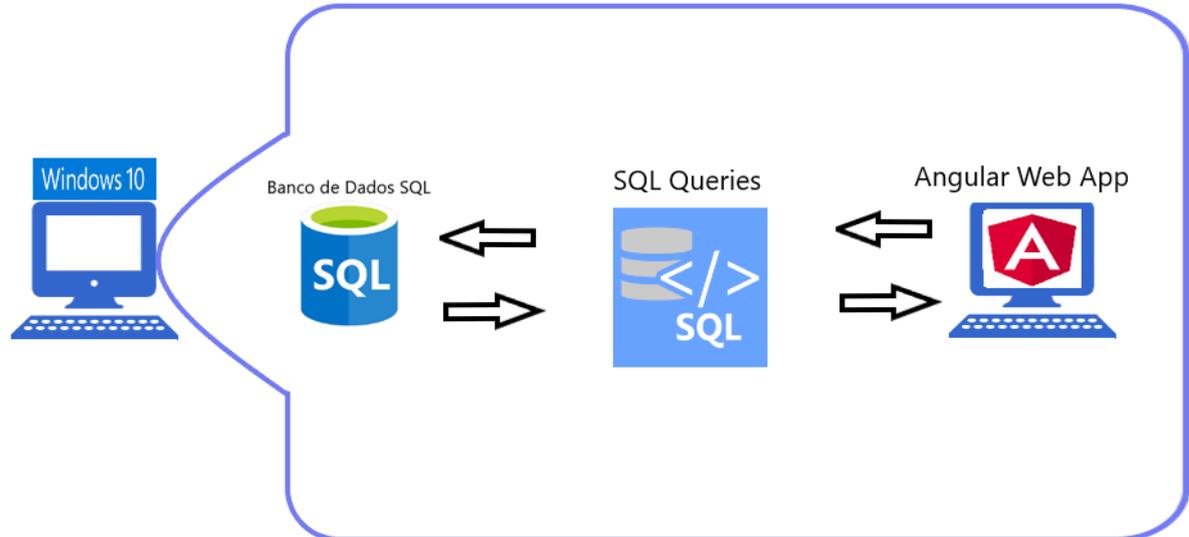
Com a finalidade de enviar mensagens MQTT se fez necessário a utilização de uma biblioteca compatível com o protocolo. Para este projeto utiliza-se a biblioteca chamada Paho MQTT. A Paho MQTT trata-se de uma biblioteca *open-source* para desenvolvimento de clientes MQTT. Esta biblioteca é um projeto do iot.eclipse.org e possui como principais características suportar comunicação MQTT segura (TLS/SSL) e por possuir versões para várias linguagens de programação dominantes.

3.2.5 Diagrama 5: Aplicação Web Ligado ao Banco de Dados Web

O Diagrama 5, ilustra a comunicação da aplicação web com o banco de dados SQL. Esta é feita por motivos de registrar o grupo e o número de identificação de cada nó, valor de `id_tb_log`, ainda na Figura 7, e para coleta de dados na hora de mostrar dados históricos em forma gráfica. Com dados atualizados no banco de dados, resolve-se problemas de concorrência entre múltiplas sessões da aplicação web, bem

como, alivia a carga do *broker MQTT*, que teoricamente, poderia ser uma outra opção para comunicação destes dados.

Diagrama 5 - Conexão aplicação web com banco de dados



Fonte: Autoria própria, utilizando o software Paint3D.

3.3 APLICAÇÃO WEB

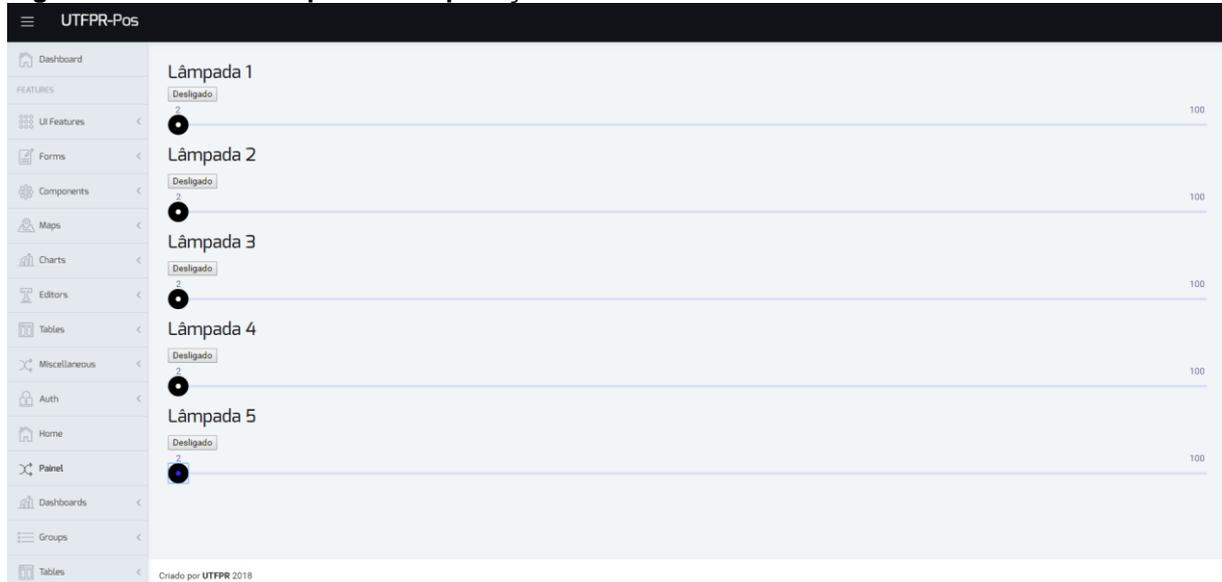
A aplicação web foi desenvolvida a partir do framework Nebular NGX, o qual permite que o desenvolvedor foque em regras de negócio ao invés de interfaces. A partir da instalação do Angular foi possível baixar todo o framework utilizando comandos “npm install”. O Npm é um gerenciador de pacotes para a linguagem de programação JavaScript. É o gerenciador de pacotes padrão para o ambiente de tempo de execução do JavaScript Node.js, exatamente o ambiente do Angular.

Após a adição do framework foi necessário o uso do “npm” novamente para a instalação da biblioteca Paho MQTT, já mencionada na seção 3.2.4 deste trabalho. Com a combinação destas duas bibliotecas foi possível aplicar as duas lógicas necessárias para atingir o objetivo desse trabalho. A primeira um mecanismo que seja capaz de interferir no estado de uma lâmpada e o segundo um gráfico capaz de entregar a economia que a plataforma gera.

A primeira parte pode ser vista na Figura 8, cada nó contém uma barra que pode ser arrastada e que representa o nível de porcentagem de dimerização que será aplicada ao nó correspondente, bem como, um botão que desliga ou liga o

equipamento luminoso atrelado ao nó. Isto é feito por envio de mensagens MQTT explicados no capítulo 3.2.4.

Figura 8 - Painel de lâmpadas na aplicação web



Fonte: Autoria própria, utilizando o software Paint3D.

A segunda parte é responsável por gerar um relatório de consumo sobre cada nó. Esta parte utiliza informações gravadas de forma histórica no banco de dados e aliado a informação de custo do kWh relacionado a cidade de Curitiba sobre um ambiente domiciliar, no valor de 0,37035 reais, é possível estimar a economia gerada pelo nó. Este esquema pode ser visto na Figura 9 (COPEL, 2018).

Figura 9 - Gráfico de economia na aplicação web



Fonte: Autoria própria, utilizando o software Paint3D.

4 RESULTADOS

Após a utilização de conceitos técnicos explicados no capítulo 2 e sua aplicação prática exemplificada no capítulo 3º o projeto teve os resultados esperados. Ou seja, a plataforma com sua aplicação web conseguiu inferir no estado de um equipamento luminoso e mostrar dados históricos sobre o mesmo.

Posteriormente, quando os nós estávamos prontos e conectados no *broker MQTT* foi possível verificar o acionamento e desligamento das lâmpadas, bem como o efeito de dimerização, o qual diminui ou aumenta a intensidade luminosa.

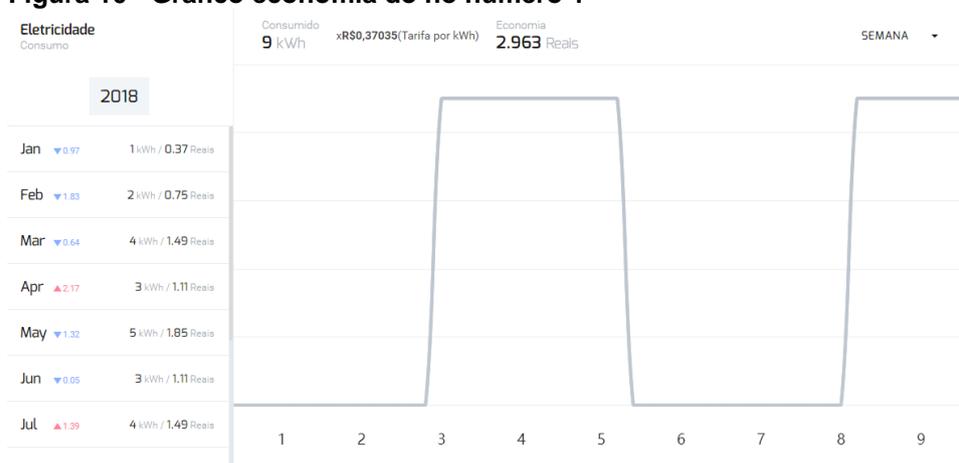
Houve uma dificuldade na prática na parte de gravar os registros do *broker* no banco de dados com a rotina escrita em *javascript* pois essa poderia apenas processar uma mensagem MQTT de cada vez. Para contornar esse problema, foi necessário reescrever este módulo para utilização de threads.

Com os blocos em funcionamento e após diversos testes, chegou-se aos resultados descritos nos tópicos 4.1 a 4.3.

4.1 APLICAÇÃO WEB

A Figura 10, ilustra o *dashboard* da interface web com os testes executados no nó 1 da plataforma. É possível observar 2 picos de medição no valor máximo de consumo da lâmpada. Esses picos foram conseguidos no momento do acionamento das lâmpadas e registrados no bando de dados para posteriormente parecer no gráfico da aplicação web.

Figura 10 - Gráfico economia do nó número 1



Fonte: Autoria própria, utilizando o software Paint3D.

4.2 BROKER MQTT

A Figura 11, ilustra a mensagem do nó 1 da plataforma comunicando-se com o *broker MQTT*. O número “1” indica o acionamento da lâmpada neste caso, e consequentemente o número “0” para desligamento da mesma. Os números entre “99” e “89” significam o nível de porcentagem de intensidade luminosa no nó 1. Ou seja, logo o nó foi acionado seu nível de luminosidade foi diminuída para 89%. Este foi um teste concebido para verificar se o nó estava funcionando juntamente ao *broker MQTT*.

Figura 11 - Client MQTT inscrito no tópico genérico

```

C:\WINDOWS\system32\cmd.exe - mosquitto_sub.exe -t "mqtt/#"
C:\Users\Konrado Bienert\Downloads\mosquitto-win32-binary-master\mosquitto-win32-binary-master>mosquitto_sub.exe -t "mqtt/#"
1
99
99
98
98
97
97
96
96
95
95
94
94
93
93
92
92
91
91
90
90
89
89
0
1
0
  
```

Fonte: Autoria própria, utilizando o software CMD.

4.3 NÓ

Para um trabalho futuro, em que se diz respeito ao nó, seria possível trocar o hardware associado a este. Uma vez que o NodeMCU é destinado a prototipação e em estados diferentes de temperatura, pressão e interferência eletromagnética.

Além disso, todo o circuito para acionamento da lâmpada não contempla um mecanismo para acionamento manual, como por exemplo, um interruptor. O qual pode ser explorado futuramente.

Outro recurso não utilizado é a gravação por meio WiFi, o qual ajudaria se o projeto houvesse a necessidade de atualizar os nós em tempo real e de forma remota.

5 TRABALHOS FUTUROS

Para a construção de um produto que possa ser testado no mercado comercial são necessárias algumas alterações para tornar o projeto mais confiável principalmente no que diz respeito ao processo ao envio de informações pelos nós e gravação de dados no banco de dados.

5.1 MICRONTROLADOR

Embora a utilização do ESP8266 seja muito conveniente para a elaboração de protótipos, na prática recomenda-se que a implementação de produtos finais seja feita com a utilização de módulos mais recentes, como o ESP32 sucessor do ESP8266, especialmente por ter melhor suporte a novos protocolos ou implementações novas de protocolos já difundidos em Internet das Coisas.

Este trabalho induz que o microcontrolador do ESP8266 esteja como atuador também, logo desta forma pode-se utilizar outro microcontrolador, como por exemplo, o microcontrolador CC2650 da Texas Instruments para esta função. Assim o projeto contaria com dois microcontroladores, o primeiro responsável por enviar mensagens ao *broker MQTT* e o segundo para o controle de estado das lâmpadas.

5.2 BROKER MQTT

No protótipo foi utilizado um link ADSL ligado a um modem/roteador doméstico, este por sua vez conectado a um computador de uso doméstico. Em condições normais de temperatura, pressão, umidade essa abordagem atende aos requisitos de teste. Quando o ambiente, principalmente o industrial, se distancia do ambiente doméstico, é necessária uma abordagem diferente para esse tipo de solução.

Outras arquiteturas de rede podem ser utilizadas, como a utilização de um *broker MQTT* que esteja hospedado em uma solução em nuvem. Assim garantindo o uso contínuo, sem quedas e com uma latência constante de serviço.

5.3 BANDO DE DADOS

Para a implementação do protótipo da plataforma foi utilizado um banco de dados MYSQL. A escolha dessa tecnologia foi devido à simplicidade de desenvolvimento, por ser grátis para fins acadêmicos e de pequenos negócios e porque implementa a funcionalidade básica necessária para o projeto, gravação de dados estruturalmente. Para aplicações comerciais pode-se utilizar versões disponíveis em nuvem ao invés de um computador doméstico o que deixaria a ferramenta mais adequada para utilização profissional.

Dependendo da criticidade de sigilo das informações de cada organização ou empresa, pode-se também pensar em uma abordagem proprietária da funcionalidade, como servidores internos dedicados a esse tipo de implementação.

6 CONSIDERAÇÕES FINAIS

Após a execução de inúmeros testes com o protótipo desenvolvido, houve à conclusão de que todos os requisitos do protótipo foram atendidos. Trata-se de uma plataforma de controle luminoso o qual apresenta os resultados em um ou mais gráficos para o cliente através de uma aplicação web. Os conceitos utilizados para a construção desse protótipo podem ser replicados para quaisquer outras tecnologias. A implementação do protótipo utilizando-se a abordagem em blocos com entradas e saídas bem definidas também foi importante para uma visão de futuro do produto. Bem como a escolha de desenvolvimento de uma aplicação web tornou este produto acessível também a partir de plataformas móveis, as quais estão bem populares atualmente.

REFERÊNCIAS

ALGAWORKS. **Fluxo aplicação Angular**. Copyright© 2018, Amazon Web Services, consultado em: 28 out. 2018. Disponível em: <<http://s3.amazonaws.com/algaworks-blog/wp-content/uploads/Angular-services.png>>. Acesso em: 10 nov. 2018.

ALVES, Josemar. **Configurando GPIO do ESP8266-01**. Copyright© Embarcados, nov. 2016. Disponível em: <<https://www.embarcados.com.br/configurando-gpio-do-esp8266-01/>>. Acesso em: 17 nov. 2018.

CEPA. **Lâmpadas**. Centro de Ensino e Pesquisa Aplicada (CEPA), Programa Pró Ciência - FAPESP, out. 2018. Disponível em: <<http://www.cepa.if.usp.br/energia/energia1999/Grupo6A/tipolamp.htm>>. Acesso em: 13 out. 2018.

COPEL. **Preço de tarifa energética**. Consultado em: 28 out. 2018. Disponível em <<http://www.copel.com/hpcopel/root/nivel2.jsp?endereco=%2Fhpcopel%2Froot%2Fpagcopel2.nsf%2Fdocs%2FCDA0A6C0E19F787803257EBB00451CC5>>. Acesso em: 28 out. 2018.

CURVELLO, André. **Apresentando o módulo ESP8266**. Copyright© Embarcados, publicado em: 29 abr. 2015. Disponível em: <<https://www.embarcados.com.br/modulo-esp8266/>>. Acesso em: 17 out. 2018.

GITHUB. **Ngx-admin demo**. Copyright© GitHub, Inc, consultado em: 28 out. 2018. Disponível em: <<https://github.com/akveo/ngx-admin/>>. Acesso em: 15 nov. 2018.

KORTH, H. F.; SILBERSCHATZ, A. **Sistemas de bancos de dados**. 2. ed. São Paulo: Makron Books, fev. 1994.

MARA, Jaqueline. **Eletroímãs: Efeitos magnéticos de corrente**. Copyright© Jaqueline Mara, publicado em: 20 nov. 2012. Disponível em: <<http://trabalhofisica2012.blogspot.com>>. Acesso em: 15 out. 2018.

NEOCONTROL. **5 Benefícios de automação de iluminação residencial**. Copyright© neocontrol, out. 2018. Disponível em: <<https://www.neocontrol.com.br/news/5-beneficios-de-automacao-de-iluminacao-residencial>>. Acesso em: 17 out. 2018.

OLIVEIRA, Sérgio de. **Internet das Coisas com ESP8266, Arduino e Raspberry PI**. 1. ed. São Paulo: Novatec, jun. 2017.

PLURALSIGHT. **Curso web applications**. Copyright© 2004-2018 Pluralsight LLC, 2018. Disponível em: <<https://www.pluralsight.com/>>. Acesso em: 20 out. 2018.

PELAYO, Roland. **Nodemcu pinout reference**. Copyright© Teach Me Microcontrollers, publicado em: 26 abr. 2018. Disponível em: <<https://www.teachmemicro.com/nodemcu-pinout>>. Acesso em: 27 nov. 2018.

UNILAB. **Glossário de termos**. Universidade da Integração Internacional da Lusofonia Afro Brasileira (UNILAB), Diretoria de Tecnologia da Informação (DTI), 2018. Disponível em: <<http://dti.unilab.edu.br/dicas/glossario-de-termos/>>. Acesso em: 27 nov. 2018.

YUAN, Michael. **Conhecendo o MQTT**: Por que o MQTT é um dos melhores protocolos de rede para a Internet das Coisas? Copyright© IBM, publicado em: 04 out. 2017. Disponível em: <<https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html>>. Acesso em: 27 nov. 2018.