

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
GRADUAÇÃO EM ENGENHARIA MECATRÔNICA

JONATHAN DA SILVA BRAGA

**PROPOSTA DE UMA FERRAMENTA DE BAIXO CUSTO PARA O
MONITORAMENTO DE ENERGIA RESIDENCIAL**

NATAL – RN
JUNHO DE 2018

JONATHAN DA SILVA BRAGA

**PROPOSTA DE UMA FERRAMENTA DE BAIXO
CUSTO PARA O MONITORAMENTO DE
ENERGIA RESIDENCIAL**

Trabalho de Conclusão de Curso de Engenharia Mecatrônica da Universidade Federal do Rio Grande do Norte, apresentado como requisito parcial para a obtenção do grau de Bacharel em Engenharia Mecatrônica

Orientador: Carlos Manuel Dias Viegas

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
GRADUAÇÃO EM ENGENHARIA MECATRÔNICA

NATAL – RN
JUNHO DE 2018

JONATHAN DA SILVA BRAGA

PROPOSTA DE UMA FERRAMENTA DE BAIXO CUSTO PARA O MONITORAMENTO DE ENERGIA RESIDENCIAL

Trabalho de Conclusão de Curso de Engenharia Mecatrônica da Universidade Federal do Rio Grande do Norte, apresentado como requisito parcial para a obtenção do grau de Bacharel em Engenharia Mecatrônica

Orientador: Carlos Manuel Dias Viegas

Trabalho aprovado. NATAL – RN, 08 de Dezembro de 2017:

Prof. Dr. Carlos Manuel Dias Viegas - Orientador
UFRN

Prof. Dr. Cicrano da Silva - Coorientador
UFRN

MSc. Alguém externo - Convidado
Empresa ou instituição

NATAL – RN
JUNHO DE 2018

Este trabalho é dedicado a Marta e Osmildo.

AGRADECIMENTOS

Este trabalho não poderia ser concluído sem a ajuda de diversas pessoas as quais presto minha homenagem:

A Deus, pois sem ele eu não existiria e não conseguiria chegar onde cheguei.

Ao meu orientador Professor Dr. Carlos Manuel Dias Viegas, pela paciência, confiança e ideias que tornaram possível a elaboração deste trabalho.

Aos meus pais, Osmildo e Marta, que tanto me guiaram, me incentivaram e me apoiaram a crescer espiritualmente e profissionalmente, a minha irmã Catherine e ao meu cunhado Talis, que sempre foram um exemplo de perseverança e nunca deixaram de acreditar em mim.

À minha companheira Juliana por ter me incentivado e me apoiado em todos os momentos.

À toda a minha família e a todos os meus amigos, pois, sem eles, isso não seria possível.

*“Feliz o homem que encontrou a sabedoria e alcançou o entendimento,
porque a sabedoria vale mais do que a prata,
e dá mais lucro que o ouro.”
(Bíblia Sagrada, Provérbios 3, 13-14)*

RESUMO

Este trabalho consiste no desenvolvimento de um sistema domiciliar de monitoramento de energia elétrica de baixo custo, em tempo real. Tem como proposta um dispositivo e um sistema que juntos somam a nova realidade de meios para a economia de energia. O equipamento tem como base a placa de desenvolvimento NodeMCU e um sensor de corrente que afere o consumo em tempo real de um dispositivo. Através dos dados coletados e dos cálculos realizados pelo sistema é possível acompanhar o consumo de energia elétrica de uma forma mais fácil. O intuito é que o constante monitoramento possa trazer uma conscientização da economia de energia.

Palavras-chaves: Economia de energia. Monitoramento. Eficiência Energética. IoT.

ABSTRACT

This work consists of the development of a real-time home monitoring system for electricity, which is fully customizable. It proposes a device and a system that together add the new reality of means for energy saving. The equipment is based on the NodeMCU development board and a current sensor that measures the real time consumption of a device. Through the data collected and the calculations performed by the system, it is possible to monitor the electricity consumption in an easier way. The intention is that constant monitoring can bring an awareness of energy savings.

Keywords: Energy saving. Monitoring. Energy Efficiency. IoT

LISTA DE ILUSTRAÇÕES

Figura 1 – Capacidade de instalada de geração elétrica no Brasil (MW)	13
Figura 2 – Estrutura do Consumo de fontes primárias	14
Figura 3 – Diagrama de conexão via websocket	18
Figura 4 – ESP8266	19
Figura 5 – NodeMCU	20
Figura 6 – SCT-013-000	20
Figura 7 – Visão geral do ambiente	21
Figura 8 – Tela inicial do sistema	22
Figura 9 – Chamada de API	23
Figura 10 – Esquema das tabelas do banco de dados	24
Figura 11 – Cadastro de um cômodo	25
Figura 12 – Lista dos cômodos cadastrados	25
Figura 13 – Cadastro de um dispositivo	26
Figura 14 – Listagem dos dispositivos cadastrados	26
Figura 15 – Tela de configuração dos parâmetros do sistema	27
Figura 16 – Demonstrativo do gasto de cada dispositivo	28
Figura 17 – Consumo geral de todos os dispositivos por cômodo ao longo do ano . .	28
Figura 18 – Consumo geral de todos os dispositivos por cômodo ao longo do mês atual	29
Figura 19 – Consumo geral de todos os dispositivos por cômodo ao longo do mês anterior	29
Figura 20 – Visão geral da gerência de um cômodo	30
Figura 21 – Vincular dispositivo ao cômodo	30
Figura 22 – Ações que podem ser realizadas no dispositivo vinculado	31
Figura 23 – Alerta exibido quando o consumo supera os 50% do previsto	31
Figura 24 – Alerta exibido quando o consumo supera os 80% do previsto	31
Figura 25 – Simulador de gastos	32
Figura 26 – Circuito demonstrativo para comunicação com o <i>power monitor</i>	34

LISTA DE TABELAS

Tabela 1 – Custos do dispositivo de demonstração	35
--	----

LISTA DE ABREVIATURAS E SIGLAS

HTML	<i>HyperText Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
API	<i>Application Programming Interface</i>
HTTP	<i>HyperText Transfer Protocol</i>
TCP	<i>Transmission Control Protocol</i>
SQL	<i>Structured Query Language</i>
SDK	<i>Software Development Kit</i>
IoT	<i>Internet of Things</i>
JSON	<i>JavaScript Object Notation</i>
SCT	<i>Split-Core Current Transformer</i>
REST	<i>Representational State Transfer</i>
EPE	<i>Empresa de Pesquisa Energética</i>
ANEEL	<i>Agência Nacional de Energia Elétrica</i>
CEMIG	<i>Companhia Energética de Minas Gerais</i>
SIN	<i>Sistema Interligado Nacional</i>
IBGE	<i>Instituto Brasileiro de Geografia e Estatística</i>
SOC	<i>System on a Chip</i>

LISTA DE SÍMBOLOS

GWh	Gigawatt-hora
KWh	Quilowatt-hora
$R\$$	Moeda corrente oficial da República Federativa do Brasil

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Motivação	15
1.2	Objetivos	15
1.3	Metodologia	15
1.4	Levantamento bibliográfico	16
1.5	Estrutura do Trabalho	16
2	EMBASAMENTO TEÓRICO	17
2.1	<i>Ferramentas e linguagens</i>	17
2.1.1	<i>Node.js</i>	17
2.1.2	<i>JavaScript</i>	17
2.1.3	<i>WebSocket</i>	18
2.1.4	<i>SQL</i>	18
2.1.5	<i>Fritzing</i>	19
2.2	<i>Componentes Físicos</i>	19
2.2.1	<i>ESP8266</i>	19
2.2.2	<i>Sensor de Corrente SCT 013-000</i>	20
3	DESENVOLVIMENTO	21
3.1	<i>Visão Geral</i>	21
3.2	<i>Software</i>	21
3.3	<i>Hardware</i>	32
3.4	<i>Resultados</i>	34
4	CONCLUSÃO	36
	REFERÊNCIAS	38

1 INTRODUÇÃO

Com o crescente consumo de energia elétrica nos últimos tempos, a demanda por produção da mesma teve um crescimento significativo, trazendo consigo impactos ambientais e econômicos. O Brasil por mais que possua em seu território grandes possibilidades para a construção de fontes de obtenção de energia, não está isento do problema da alta demanda por energia elétrica. Problema que se agravou em 2015 quando o país começou a passar por uma crise hídrica (MATTOS, 2015).

Como a Figura 1 mostra, a maior parte da energia elétrica gerada no Brasil é por meio de hidrelétricas, essa dependência energética junto com a crise hídrica que o país sofreu culminou em uma política de racionamento e aumento dos impostos - taxa inflacionária no consumo de energia elétrica - que impactou diretamente a vida de cada cidadão brasileiro, trouxe consequências, como o aumento do custo da energia elétrica. Entre 2015 e 2016 a crise hídrica no Brasil não interferiu apenas na conta de luz mas trouxe um aumento na inflação do país.

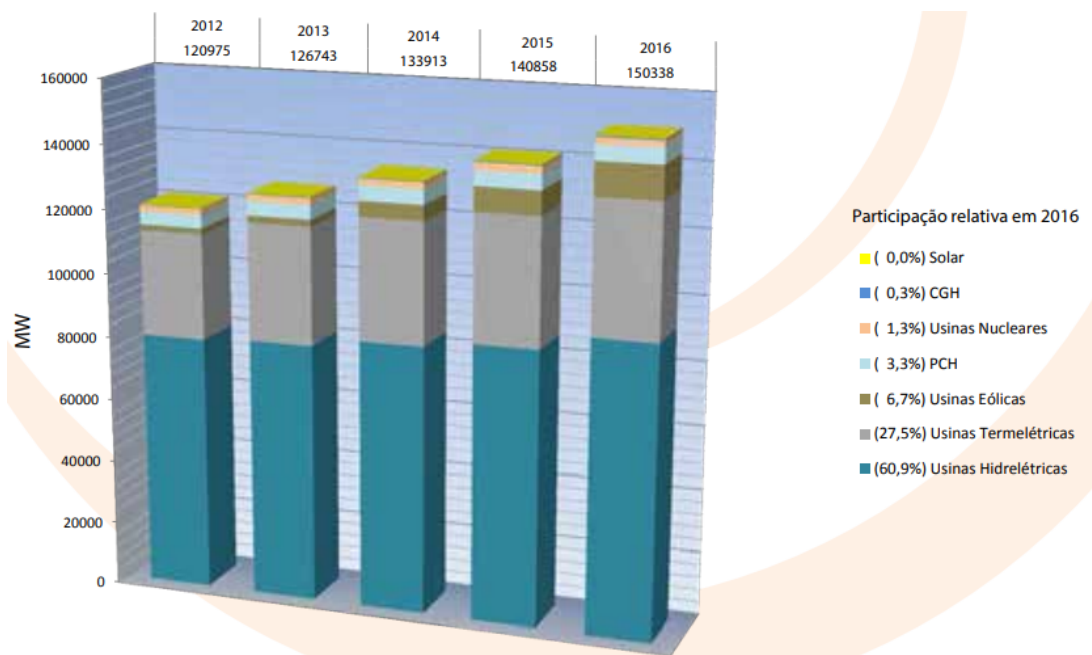


Figura 1 – Capacidade de instalada de geração elétrica no Brasil (MW)

Fonte – (EPE ANUÁRIO, 2017, p. 57).

O Consumo de energia elétrica é um dos principais indicadores de desenvolvimento e de qualidade de vida de um país. Esse índice é tão importante que reflete diretamente no ritmo de vida de uma população, pois mostra se as atividades industriais de uma nação estão ou não em um bom ritmo e pode detectar se o comércio está em alta, devido aos

bens e serviços que o povo adquiriu (GOLDEMBERG, 1998). Porém um crescimento desordenado na população e um crescimento exponencial no consumo de energia pode acarretar em problemas para um determinado país. Analisando os dados (EPE BALANÇO FINAL, 2017), o consumo de energia elétrica no Brasil vem crescendo ao longo dos anos, nos últimos 35 anos teve um crescimento médio de 6,72%. Após a crise que o Brasil sofreu entre os anos 2002 e 2005 houve um crescimento de 4,91% na demanda energética do país. A Figura 2 nos mostra o cenário de crise energética que o Brasil vinha passando ao longo dos anos, até 2008 o país consumia mais do que produzia.

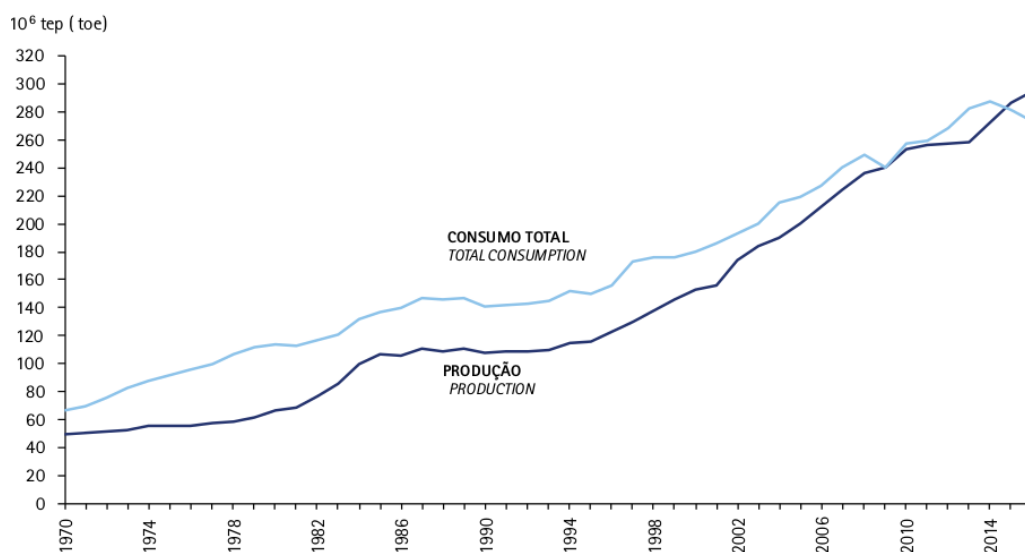


Figura 2 – Estrutura do Consumo de fontes primárias

Fonte – (EPE BALANÇO FINAL, 2017, p. 43).

Para poder acompanhar a crescente demanda por energia elétrica, o governo brasileiro autorizou a construção de mais de 24 hidrelétricas (ANDRADE, 2011). Um grande problema desse planejamento que o governo fez são os inúmeros impactos ambientais e econômicos, um exemplo prático é a usina de Belo Monte - Rio Xingu, Pará - obra que foi planejada para ser a maior hidrelétrica do Brasil, com capacidade de abastecer 40% das residências (GOVERNO, 2011). Deveria ter sido finalizada completamente e ter seu total funcionamento no segundo semestre de 2015, mas até os dias atuais não opera em 100%. Vale salientar que a construção trouxe o desmatamento de áreas indígenas, alagamentos permanentes, comprometimento da fauna e flora e aumento da dificuldade dos transportes fluviais de comunidades ribeirinhas.

Diante da problemática do alto consumo de energia elétrica, surge uma indagação: "Construir usinas mesmo sabendo dos impactos negativos que podem surgir, ou não construí-las e aumentar a tarifação pelo consumo de energia visando um uso mais consciente?". A resposta para essas e outras questões que podem aparecer não são fáceis. Entretanto o governo brasileiro optou pela construção de novas usinas e pelo aumento da tarifação no

consumo de energia elétrica. Uma medida totalmente cabível que ainda é desconhecida por alguns brasileiros é a chamada "*exposição da informação*", deixando bem claro quanto o consumidor tem gastado ou consumindo ao longo do mês em sua residência, isso é possível graças a equipamentos que estão sempre monitorando a rede elétrica.

O trabalho apresentará uma forma barata e eficiente de monitorar a energia elétrica de uma residência em tempo real, possibilitando ao usuário possuir informações valiosas a todo momento. O sistema também disponibiliza de ferramentas para que caso o usuário possua o conhecimento necessário o mesmo venha a modificar ou aprimorar o gerenciamento de energia. O sistema nomeado de *Power Monitor* traz consigo uma nova forma de enxergar o consumo dos aparelhos presentes em uma residência, trocando o quilowatt-hora pela moeda de circulação no país, o real, tal mudança é o principal fator de conscientização, pois aproxima o consumidor dos gastos energéticos.

1.1 Motivação

A motivação deste trabalho partiu da análise da atual conjuntura do cenário energético brasileiro que é baseado na falta de transparência da informação levada ao usuário. Notando-se a dificuldade de entender o consumo de energia elétrica dos dispositivos presentes em uma residência, fez-se necessário o desenvolvimento de um sistema de monitoramento para uso consciente da energia elétrica.

1.2 Objetivos

O objetivo deste trabalho é propor um sistema de monitoramento de energia elétrica, de baixo custo, para o uso consciente da mesma. O sistema faz uso de medições e cálculos para uma melhor visualização do consumo de energia. Onde o seu principal diferencial é a troca da unidade de medida referente ao consumo elétrico de um aparelho pela moeda de circulação no país.

1.3 Metodologia

O Power Monitor foi implementado utilizando a linguagem de programação JavaScript que por sua vez se comunica com uma placa de desenvolvimento NodeMCU. O Power Monitor possui um servidor web que é responsável por gravar todos os dados coletados em um banco de dados e fazer a comunicação entre software e Hardware. A interface web consome todos os dados gravados e após trata-los e realizar alguns cálculos o resultado é apresentado em forma de gráficos e alertas.

1.4 Levantamento bibliográfico

1.5 Estrutura do Trabalho

O trabalho foi estruturado para que se possa mostrar nos próximos capítulos os seguintes conteúdos:

- Capítulo 2: Embasamento Teórico - Descrição das tecnologias utilizadas neste trabalho;
- Capítulo 3: Desenvolvimento - Será descrita as etapas de implementação do projeto, desde uma visão geral do funcionamento a comunicação entre *hardware* e servidor *web*;
- Capítulo 4: Conclusão - Conclusão e trabalhos futuros;

2 EMBASAMENTO TEÓRICO

O *Power Monitor* surgiu da necessidade da conscientização do gasto energético e da melhor compreensão da conta de luz. Baseado nesse conceito, foi desenvolvido um *software* que permite uma fácil comunicação com qualquer equipamento construído que tenha a finalidade de monitorar a energia elétrica e um *hardware* para demonstração da comunicação entre ambos. O sistema traz uma forma mais fácil e próxima do consumidor final de se quantificar a energia elétrica consumida em um estabelecimento. No lugar do quilowatt-hora, medida usada atualmente, o *software* propõe mensurar o gasto energético em reais (R\$), trazendo a realidade do consumo mensal.

Este capítulo trará os conceitos essenciais para o entendimento do trabalho, descrevendo todas as tecnologias utilizadas no desenvolvimento do *Power Monitor*.

2.1 *Ferramentas e linguagens*

No decorrer do desenvolvimento do *software* fez-se uso de algumas tecnologias e linguagens de programação que serão descrita a seguir.

2.1.1 *Node.js*

Node.js é um *framework*, interpretador do código JavaScript (subseção 2.1.2), com o foco do uso da linguagem do lado do cliente para servidores. Com um objetivo simples que é ajudar desenvolvedores na criação de aplicações de alta escalabilidade, com códigos capazes de administrar e manipular várias conexões simultaneamente em um único servidor. O *Node.js* é baseado na *runtime V8 JavaScript Engine*. Foi desenvolvido por Ryan Danhl em 2009, e o seu desenvolvimento é mantido pela fundação *Node.js* e *Linux Foundation* (PATEL, 2018).

2.1.2 *JavaScript*

JavaScript é uma linguagem de programação interpretada de alto nível, juntamente com HTML e CSS é uma das linguagens mais utilizadas no mundo *web*. Após o uso da linguagem as páginas *web* começaram a ter uma maior interatividade com o usuário. A grande maioria dos *browsers* tem um mecanismo de compilação dedicado para o JavaScript (LONGEN, 2018). Por ser uma linguagem multi-paradigma o JavaScript suporta paradigmas funcionais, orientados a eventos e até mesmo paradigmas de orientação a objeto. Inicialmente era usada apenas no lado do cliente em *web browsers*, mas atualmente está

presente em vários outros tipos de *softwares* incluindo servidores - como já foi discutido na subseção 2.1.1 - *databases* e até sistemas *desktop* como os leitores de PDF, programas de música e recentemente vem ganhando espaço no desenvolvimento de aplicativos para celular (BLINI, 2016).

2.1.3 WebSocket

A ideia da tecnologia surgiu da problemática onde as comunicações entre servidor e aplicação eram baseadas na sobrecarga do HTTP, que não é indicado para aplicações de baixa latência. O WebSocket pode ser definido como uma API que estabelece a conexão entre aplicação e servidor. Resumidamente é uma conexão, baseada no protocolo TCP, persistente entre servidor e cliente onde ambas as partes podem enviar ou receber informações. A forma como a conexão acontece é simples: o cliente e o servidor antes de tudo devem negociar o *handshake*, que segundo (DAVID, 2011) é o processo pelo qual duas máquinas afirmam uma a outra que a reconheceu e está pronta para iniciar a comunicação. No momento em que a negociação é confirmada a conexão é estabelecida, criando um canal de comunicação bidirecional. A Figura 3 retrata o cenário descrito.

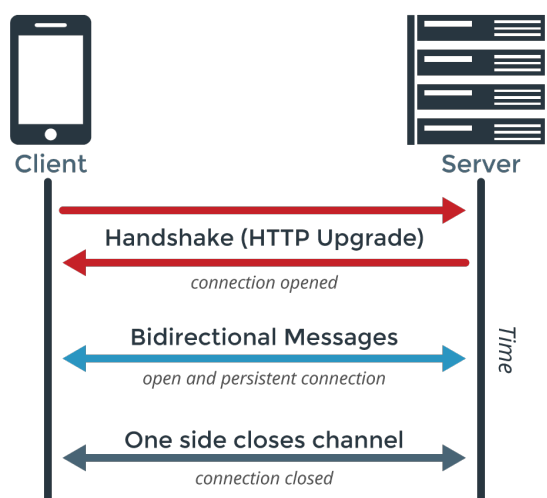


Figura 3 – Diagrama de conexão via websocket

Fonte – <https://www.pubnub.com/learn/glossary/what-is-websocket/>

2.1.4 SQL

Structured Query Language, ou comumente conhecida como SQL é uma linguagem padrão de banco de dados. Diferentemente das outras linguagens de banco de dados a consulta em SQL especifica a forma do resultado e não o caminho para chegar nele. Uma outra grande diferença é que a linguagem SQL é declarativa diferindo mais uma vez das outras linguagens que por sua vez são procedurais (REZENDE, 2013).

O MySQL é um sistema de gerenciamento de banco de dados que utiliza a linguagem SQL. Atualmente é o sistema mais popular em gerenciamento de banco de dados. Sua rápida popularização deve-se a fácil comunicação entre servidor e aplicação.

2.1.5 Fritzing

O *Fritzing* é uma iniciativa *open source* que inicialmente foi designada a desenvolvedores amadores que gostasse de tirar a sua ideia do papel. Em poucas palavras a plataforma auxilia os desenvolvedores por meio de uma interface gráfica nas primeiras montagens com Arduino ou outro microcontrolador, sua intuitiva interface proporciona ao usuário uma rápida montagem do circuito em protoboar. O *software* vai além e permite com que os desenvolvedores tenha uma visão tanto da protoboar, como do esquemático elétrico.

2.2 Componentes Físicos

No decorrer do desenvolvimento do *hardware* fez-se uso de alguns componentes eletrônicos e microcontrolador que serão descritos nessa seção.

2.2.1 ESP8266

É um SOC que é produzido por um fabricante chinês - Espressif - que tem como principal vantagem a comunicação *Wi-Fi* já integrada em seu circuito. O chip teve seu auge em 2014 quando "estourou" na cultura *maker* com o ESP-01 (Figura 4), essa placa permite que microcontroladores se conectem a uma rede sem fio fazendo conexões TCP/IP, tendo a capacidade de ser servidor ou cliente (ROBOCORE, 2014).

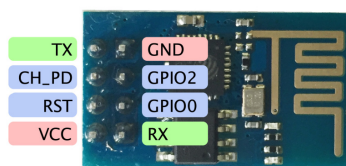


Figura 4 – ESP8266

Fonte – <http://fabacademy.org/archives/2015/doc/images/esp-01.jpg>

O NodeMcu, Figura 5, é uma plataforma de desenvolvimento *open source*. Tem como principal linguagem de script Lua, foi construído sobre o SDK ESP8266. A plataforma surgiu pouco tempo após o lançamento do ESP8266 (subseção 2.2.1). A plataforma ganhou visibilidade, pois trazia um conjunto de circuitos já previamente embutido que o ESP8266 por si só não proporcionava.

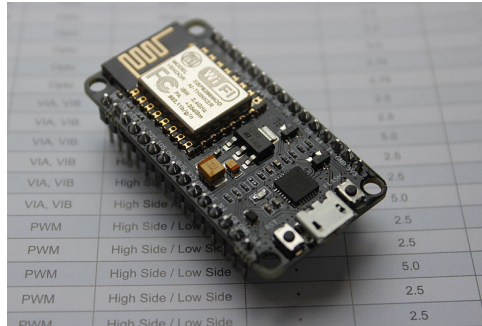


Figura 5 – NodeMCU

Fonte – https://upload.wikimedia.org/wikipedia/commons/7/7e/NodeMCU_DEVKIT_1.0.jpg

2.2.2 Sensor de Corrente SCT 013-000

O Sensor é um transformador de corrente para leituras não invasivas, possuindo um funcionamento similar a de um alicate amperímetro. Com a seguinte especificação técnica:

- 100A no primário;
- Saída de 50mA no secundário;
- Temperatura máxima 70°C;
- Temperatura mínima -25°C.

Para realizar a leitura da corrente sem a necessidade de contato elétrico, o sensor de corrente alternada utiliza as propriedades magnéticas da corrente elétrica. O SCT, Figura 6, é um sensor do tipo Transformador de Corrente, que resumidamente é um conjunto de espiras que são colocadas ao redor do condutor ao qual se quer medir a corrente.



Figura 6 – SCT-013-000

Fonte – <https://uploads.filipeflop.com/2017/07/1-34.jpg>

3 DESENVOLVIMENTO

Nesse capítulo será mostrado todo o passo a passo para o desenvolvimento do *software* e *hardware*, juntamente com a comunicação entre ambos.

3.1 Visão Geral

Em resumo pode-se ter uma visão geral de como o ambiente - *software* e *hardware* - funciona observando a Figura 7. O sistema *web* é responsável por fazer a comunicação entre o banco de dados e os dispositivos, já os eletrodomésticos são gerenciados pelo NodeMCU que possui uma comunicação direta via *websocket* com o servidor *web*.

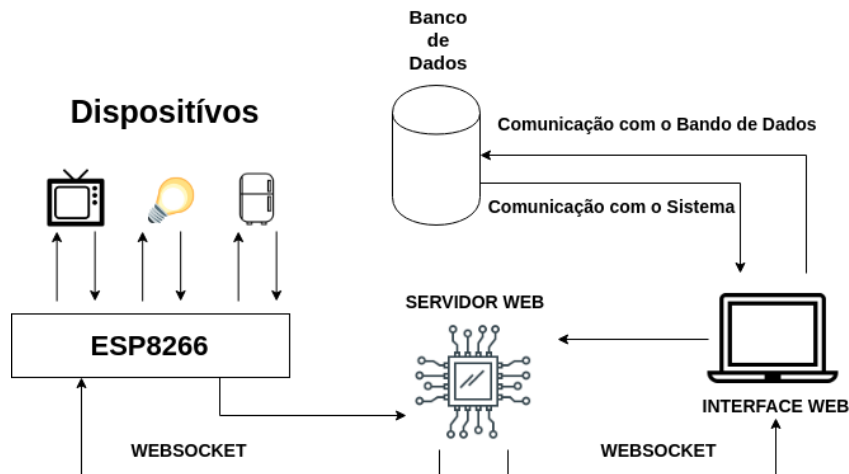


Figura 7 – Visão geral do ambiente

3.2 Software

O controle dos dispositivos de um cômodo, que estão interligados com o *NodeMCU* são controlados pelo *software*. Dessa forma todos os dispositivos que possuem comunicação com a plataforma de prototipagem e que estão cadastrados no sistema podem ser controlados (Ligar/Desligar) e também é possível ter um acompanhamento dos gastos.

O sistema possui uma interface *web* que pode ser acessada por qualquer dispositivo que tenha acesso à internet e possua um *browser*, como por exemplo: celulares, computadores, *smart tv* etc. O *software* possui uma interface de apenas um único usuário, ao acessar o sistema o usuário se depara com um visual bem agradável e fácil de se usar. Logo ao entrar no sistema o usuário se depara com a tela principal, Figura 8, nela encontra-se as principais informações que o usuário irá precisar, como também irá mostrar as opções de

cadastrar um novo dispositivo, listar os dispositivos, cadastrar um novo cômodo, listar um novo cômodo etc.

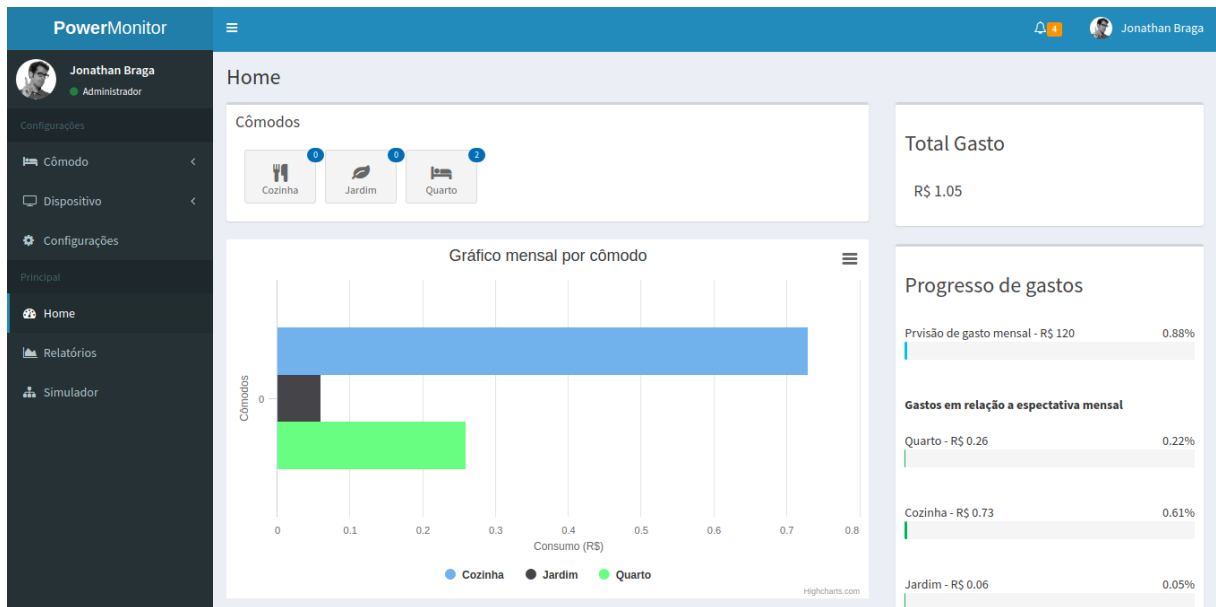


Figura 8 – Tela inicial do sistema

Todas as informações colhidas pelo servidor *web* em *node.js* (subseção 2.1.1), são recebidas e tratadas pelo interface *web*. Os dados são importantíssimos, pois mediante eles é que se torna possível a construção dos gráficos e das previsões fornecidas pelo sistema. No *power monitor* a forma de comunicação com o banco de dados é feita mediante as chamadas de API, existe uma chamada para cada ação prevista no sistema. A Figura 9 retrata bem esse cenário, pode-se perceber que o *end-point* (expressão utilizada para se referenciar a uma extremidade de um canal de comunicação.) **get-comodos** é destinado a obtenção de todos os cômodos cadastrados já o *end-point* **get-dispositivos** é destinado a obtenção de todos os dispositivos cadastrados. O motivo da comunicação entre servidor *web* e interface *web* ser feita via chamada de API é bem simples, pois qualquer sistema seja *web*, *desktop* ou qualquer outro tipo, pois basicamente precisa ter uma comunicação com uma rede *Wi-Fi* para utilizar o *power monitor*. O *software* não precisa obrigatoriamente de internet para poder funcionar, pois a comunicação entre servidor e sistema é baseada em uma rede local, justamente para que o *software* não dependa de terceiros. Para uma perfeita comunicação o ambiente só precisa está configurado na mesma rede *Wi-Fi*.

```
// Recebe dispositivos
con.query("SELECT DISTINCT * FROM dispositivo ORDER BY nome", function (err, result, fields) {
  if (err) throw err;
  io.emit("get-dispositivos", result)
});

//Recebe Dispositivos ligados
con.query("SELECT id,estado,id_dispositivo,id_comodo FROM status_dispositivo WHERE estado = 1", function (err, result, fields){
  if(err) throw err;
  io.emit("get-dispositivos-on",result);
});

//Recebe Comodos
con.query("SELECT DISTINCT* FROM comodo ORDER BY nome;", function (err, result, fields) {
  if (err) throw err;
  io.emit("get-comodos", result)
});
```

Figura 9 – Chamada de API

O *software* pode ser dividido em duas partes, servidor *web* e interface *web*. O servidor foi desenvolvido usando a linguagem *javascript* (subseção 2.1.2) e para auxílio foi utilizado o *framework node.js* (subseção 2.1.1), a comunicação entre servidor e banco de dados é feita pelo *MySQL Server* (subseção 2.1.4). A interface *web* é o agente consumidor de todos esses serviços, com uma comunicação via *webscoket* (subseção 2.1.3) com o servidor é capaz de receber e enviar dados a todo instante. A combinação desses três serviços - servidor *web*, servidor do banco de dados e interface *web* - resultou em uma aplicação intuitiva e amigável, desenvolvida para dar o total suporte às análises dos dados enviados pelo *hardware*. A Figura 10 representa o esquema das tabelas do banco de dados que foi utilizado no *power monitor*.

Como já foi apresentada a Figura 8 representa a tela inicial da interface *web*, vê-se que é possível visualizar o total gasto no mês atual, o progresso dos gastos por cômodo que por sua vez é baseado na expectativa de gasto mensal (Figura 15), a lista de todos os cômodos cadastrados e um gráfico mostrando o consumo por cômodo referente ao mês atual.

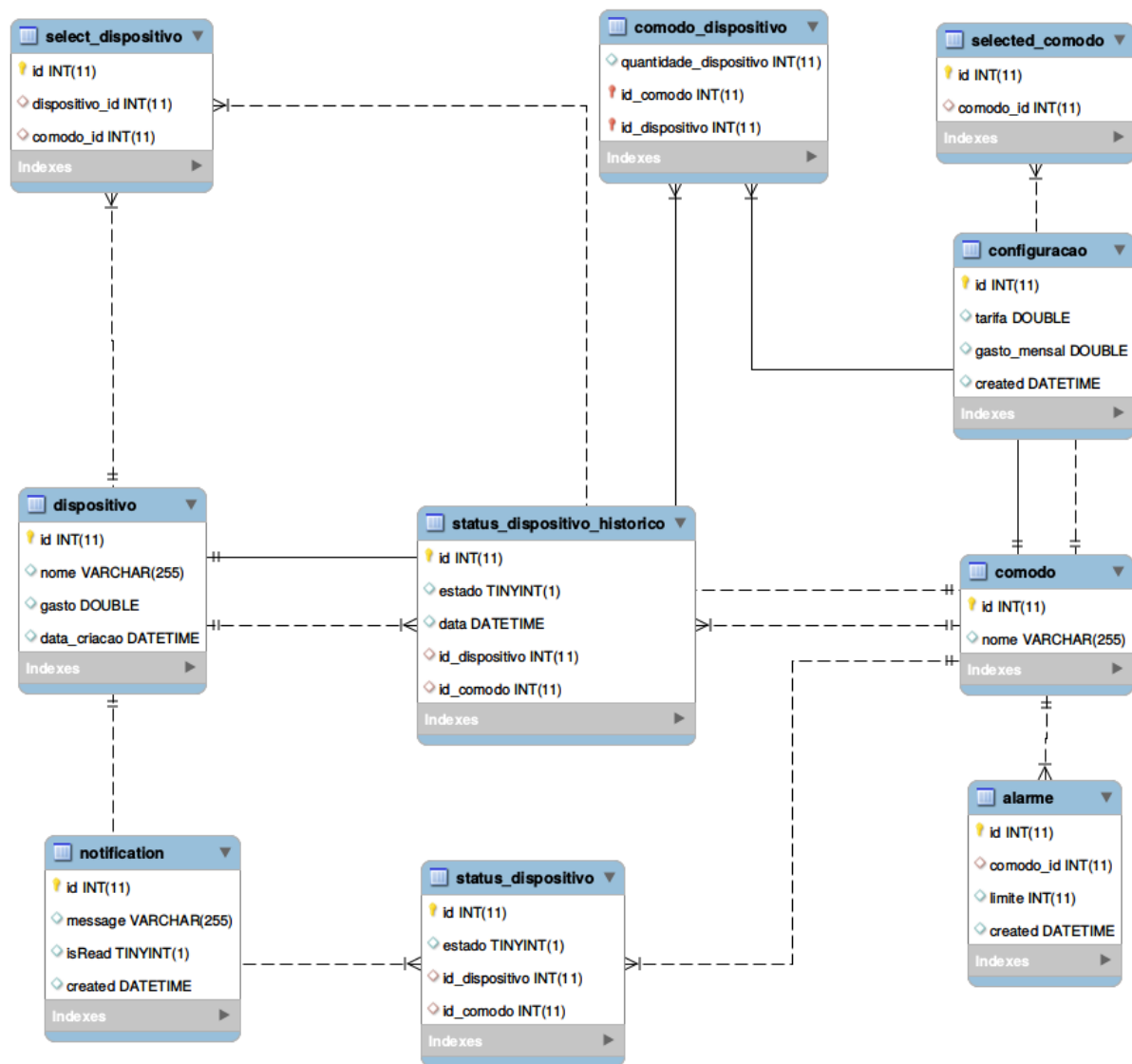


Figura 10 – Esquema das tabelas do banco de dados

As figuras: 11, 12, 13, 14 e 15, são relacionadas as telas de cadastro, listagem e de configuração da interface *web*. Nelas é possível cadastrar e listar cômodos e dispositivos assim como configurar alguns parâmetros do sistema como o preço da tarifa cobrada por quilowatt-hora pela empresa responsável e a expectativa de gasto mensal.

Uma vez que os cômodos, dispositivos e parâmetros são cadastrados no sistema é possível gerencia-los através da edição ou exclusão dos dados, que se torna possível nas telas de listagem, para os cômodos e dispositivos, já os parâmetros do sistema na própria tela de configuração se faz a edição dos dados.

The screenshot shows the 'Novo Cômodo' (New Room) form in the PowerMonitor application. The interface includes a sidebar with navigation options: Configurações, Cômodo (selected), Adicionar, Listar, Dispositivo, and Configurações. The main content area displays a form titled 'Formulário' with a single input field labeled 'Nome' and a 'Confirmar' button. The footer contains the copyright notice 'Copyright © 2017 Jonathan Braga. All rights reserved.' and the text 'Power Monitor'.

Figura 11 – Cadastro de um cômodo

The screenshot shows the 'Lista de Cômodos' (List of Rooms) in the PowerMonitor application. The interface includes a sidebar with navigation options: Configurações, Cômodo (selected), Adicionar, Listar, Dispositivo, and Configurações. The main content area displays a table titled 'Cômodos' with columns '#', 'Nome', and 'Excluir'. The table lists three rooms: Cozinha, Jardim, and Quarto. The footer contains the copyright notice 'Copyright © 2017 Jonathan Braga. All rights reserved.' and the text 'Power Monitor'.

#	Nome	Excluir
	Cozinha	Excluir
	Jardim	Excluir
	Quarto	Excluir

Figura 12 – Lista dos cômodos cadastrados

PowerMonitor

Jonathan Braga
Administrador

Configurações

Cômodo

Dispositivo

Adicionar

Listar

Configurações

Principal

Home

Relatórios

Simulador

Novo Dispositivo

Formulário

Nome

Gasto em KW/h

[Confirmar](#)

*Todo dispositivo ao ser criado tem como padrão o status de desligado.

Copyright © 2017 Jonathan Braga. All rights reserved. Power Monitor

Figura 13 – Cadastro de um dispositivo

PowerMonitor

Jonathan Braga
Administrador

Configurações

Cômodo

Dispositivo

Adicionar

Listar

Configurações

Principal

Home

Relatórios

Simulador

Lista de Dispositivos

Cômodos

#	Nome	Gasto	Data de Criação
Excluir	Ar	123 kw/h	2018-04-11T05:10:45.000Z
Excluir	Geladeira	57 kw/h	2018-05-07T20:50:42.000Z
Excluir	Televisão	23 kw/h	2018-04-11T02:50:12.000Z

Copyright © 2017 Jonathan Braga. All rights reserved. Power Monitor

Figura 14 – Listagem dos dispositivos cadastrados

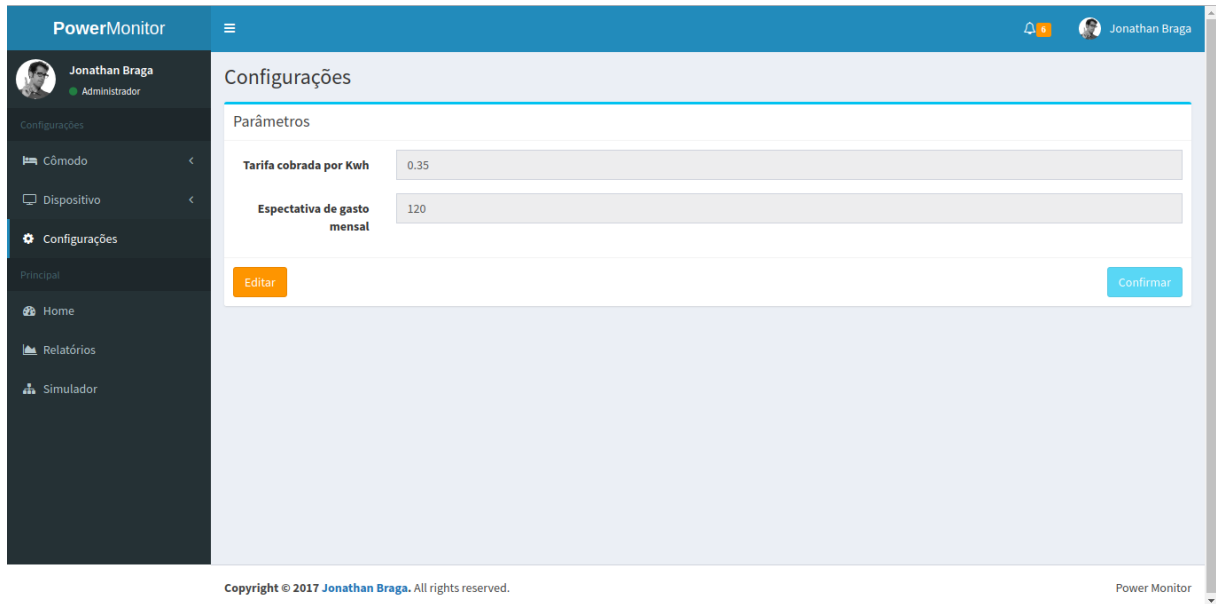


Figura 15 – Tela de configuração dos parâmetros do sistema

Uma vez que todos os cômodos, dispositivos e parâmetros já se encontram cadastrados e já exista a comunicação estabelecida com o hardware, a interface *web* disponibiliza um série de formas para visualização das informações coletadas e tratadas. Na Figura 16 é possível visualizar uma espécie de extrato do consumo de todos os dispositivos, podendo perceber quando foram ligados, quando foram desligados e assim resultando no total gasto. Já as figuras: 17, 18 e 19 referem-se aos gráficos que são construídos baseados nas informações coletadas pelo sistema.

Os gráficos por sua vez são construídos mediante aos cálculos que o sistema faz usando como base a Equação 3.1, o resultado dessa conta fornece ao sistema o consumo em reais (R\$) do dispositivo em um dado intervalo de tempo conhecido. Vale salientar que o resultado é o esperado já que o consumo do dispositivo (KWh) é pré definido pelo usuário (Figura 13), para o cálculo real do consumo usa-se a Equação 3.2 que leva em conta a corrente real que passa pelo dispositivo ao longo do tempo que ele permanece ligado. Com esses dados é que se torna viável a construção dos gráficos e extrato presentes no sistema.

$$\frac{\text{consumo do dispositivo} \times \text{tempo de uso}}{\text{número de dias no mês}} \times \text{tarifa} \quad (3.1)$$

$$\frac{(\text{tensão} \times \text{corrente}) \times \text{horas de uso por dia} \times \text{número de dias no mês}}{1000} \quad (3.2)$$

PowerMonitor

Jonathan Braga
Administrador

Relatórios

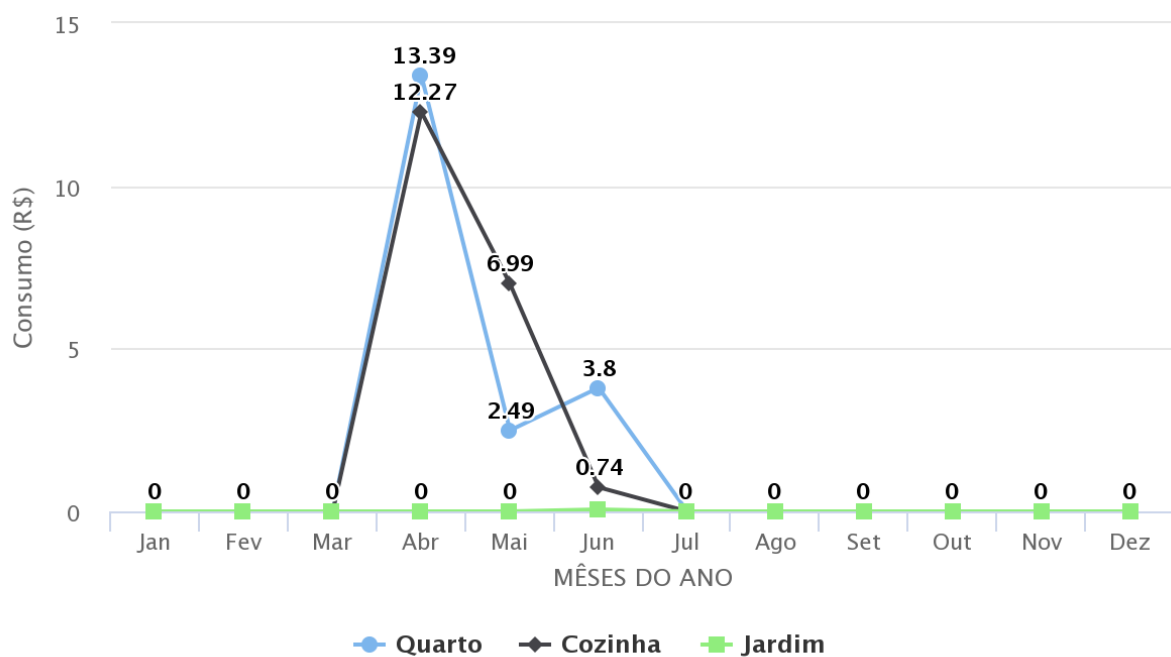
Extrato Geral

Show 10 entries

Search:

Comodo	Dispositivo	Data ao ligar	Data ao desligar	Tempo de consumo	Consumo
Cozinha	Geladeira	2018-05-13 20:37:07	2018-05-19 21:41:49	6 days	R\$ 4,020
Cozinha	Geladeira	2018-06-07 04:03:02	2018-06-07 11:00:06	6 hours	R\$ 0,193
Cozinha	Geladeira	2018-05-12 13:18:33	2018-05-12 13:24:59	6 minutes	R\$ 0,003
Cozinha	Geladeira	2018-05-12 13:11:19	2018-05-12 13:11:25	6 seconds	R\$ 0,000
Cozinha	Ar	2018-06-06 16:43:54	2018-06-06 16:44:00	6 seconds	R\$ 0,000
Cozinha	Ar	2018-06-06 17:47:21	2018-06-06 17:47:27	6 seconds	R\$ 0,000
Quarto	Televisão	2018-05-12 14:22:52	2018-05-19 21:21:47	7 days	R\$ 1,956
Quarto	Ar	2018-04-28 20:52:09	2018-04-28 20:59:55	7 minutes	R\$ 0,008
Cozinha	Ar	2018-06-07 03:32:51	2018-06-07 03:32:58	7 seconds	R\$ 0,000
Cozinha	Ar	2018-04-29 00:00:13	2018-05-07 13:02:38	8 days	R\$ 12,260

Figura 16 – Demonstrativo do gasto de cada dispositivo



Highcharts.com

Figura 17 – Consumo geral de todos os dispositivos por cômodo ao longo do ano

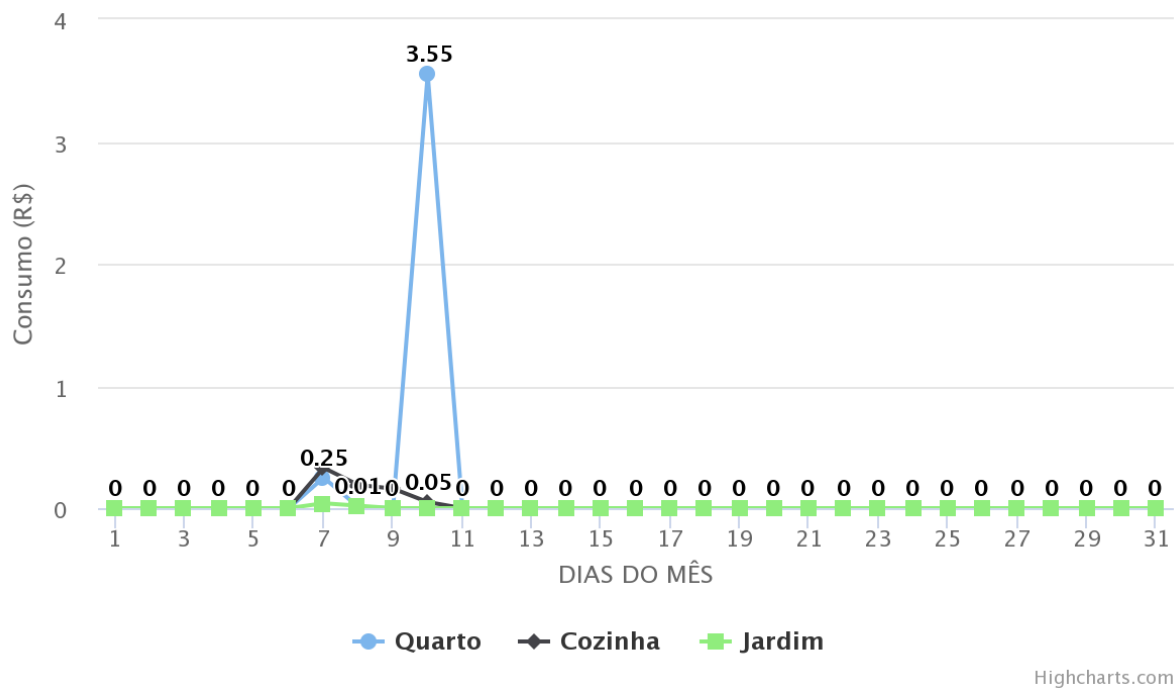


Figura 18 – Consumo geral de todos os dispositivos por cômodo ao longo do mês atual

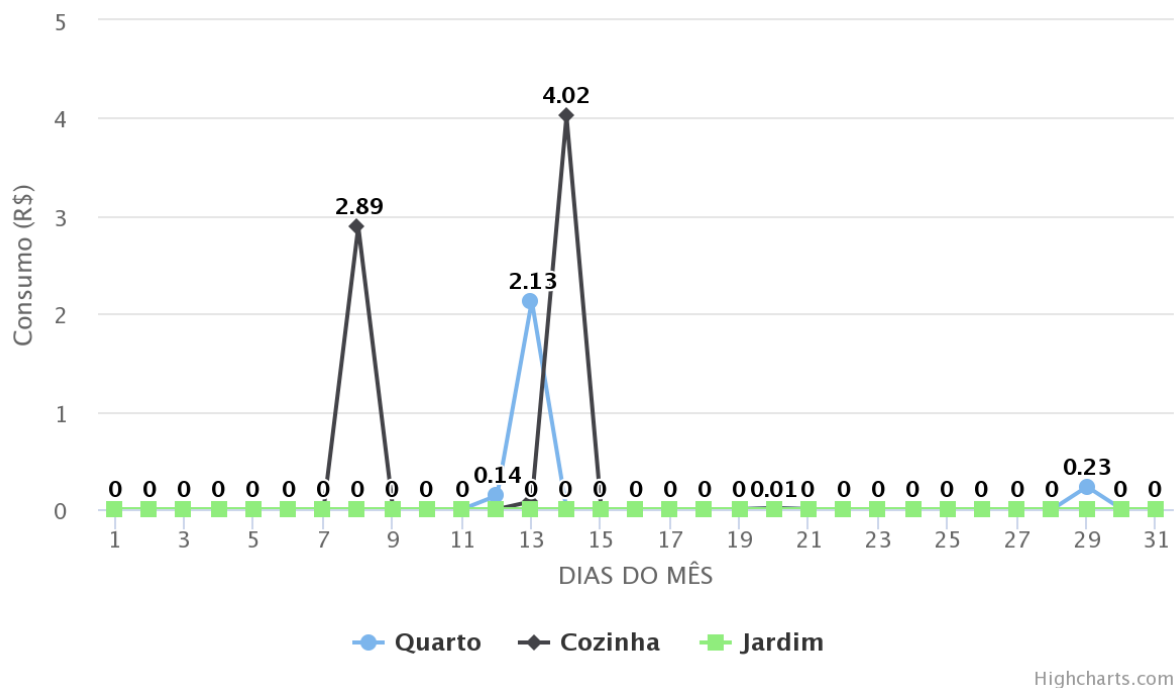


Figura 19 – Consumo geral de todos os dispositivos por cômodo ao longo do mês anterior

O *software* também possui um local específico para a gerência dos cômodos, um espaço destinado para a vinculação de dispositivos ao cômodo, visualização do total gasto em relação ao mês atual em forma de gráfico e a opção de ligar, desligar ou excluir o dispositivo do cômodo. As figuras: 20, 21 e 22 retratam o cenário descrito.

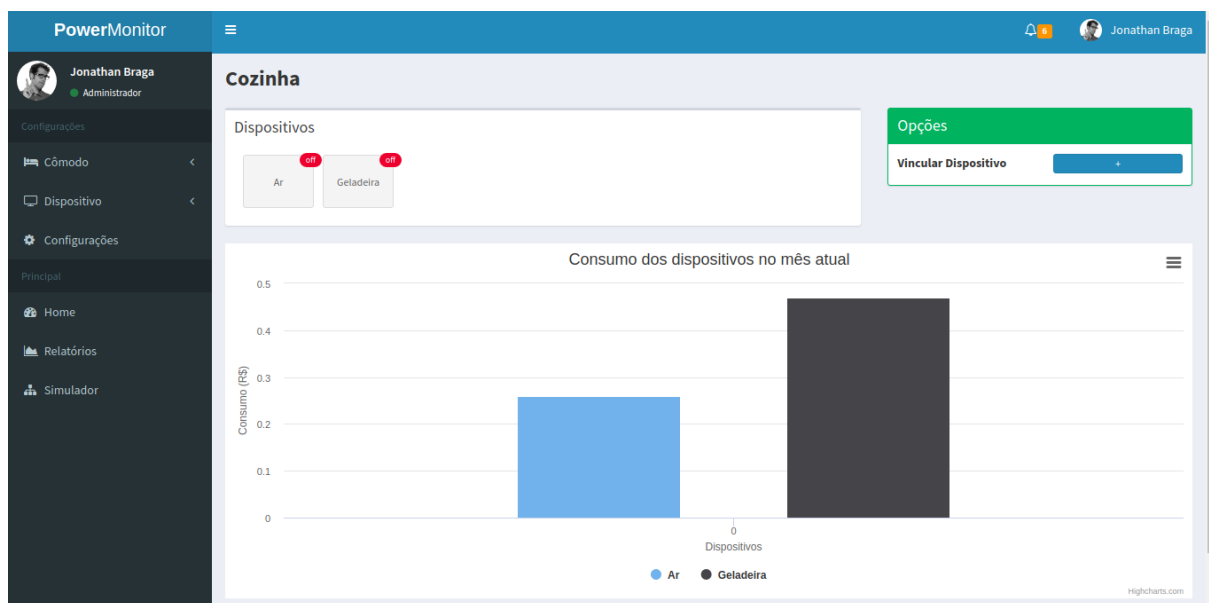


Figura 20 – Visão geral da gerência de um cômodo

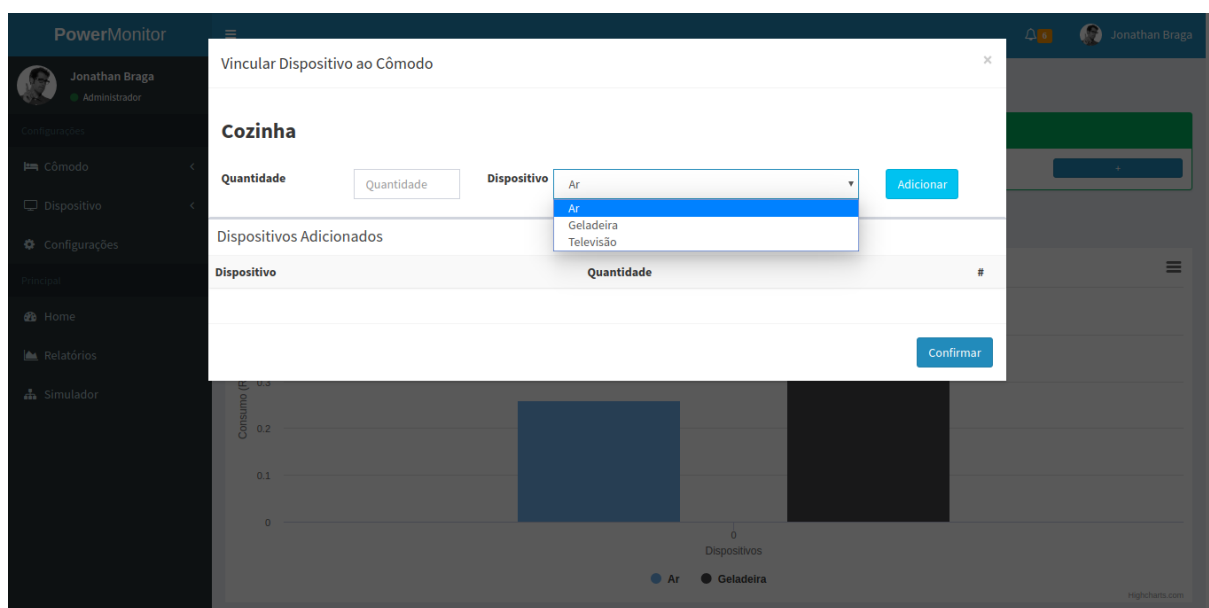


Figura 21 – Vincular dispositivo ao cômodo

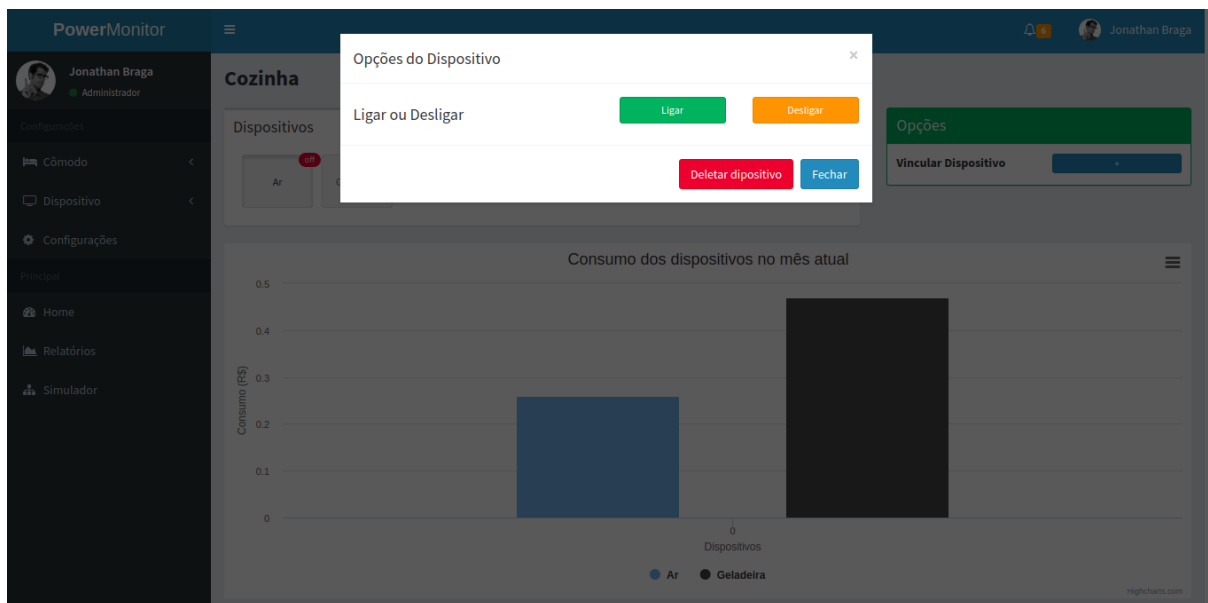


Figura 22 – Ações que podem ser realizadas no dispositivo vinculado

Outras duas funcionalidades presentes no sistema são os alarmes gerados quando o usuário passa dos 50% e dos 80% do consumo esperado, cadastrado previamente pelo mesmo. Também existe um simulador de gastos, onde é possível calcular o gasto mensal e diário que um dispositivo irá trazer para uma residência. As figuras 23, 24 e 25 retratam o descrito.



Figura 23 – Alerta exibido quando o consumo supera os 50% do previsto

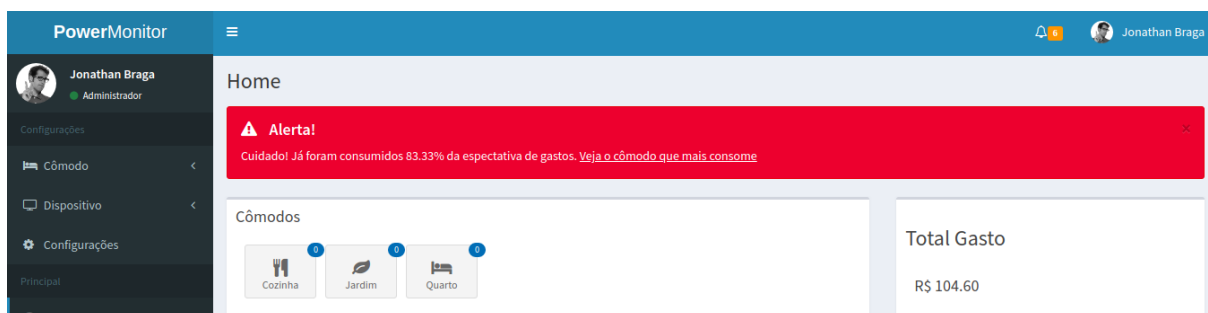


Figura 24 – Alerta exibido quando o consumo supera os 80% do previsto

PowerMonitor

Jonathan Braga
Administrador

Configurações

Cômado

Dispositivo

Configurações

Principal

Home

Relatórios

Simulador

Simulador de Gastos

Parâmetros

Consumo do aparelho (Kwh) Exemplo: 57

Espectativa de uso em horas ao longo do dia Exemplo: 12

Espectativa de uso em dias ao longo do mês Exemplo: 20

Tarifa cobrada pela concessionária Exemplo: 0.35

Calcular

*No lugar da vírgula usar o ponto.

Copyright © 2017 Jonathan Braga. All rights reserved. Power Monitor

Figura 25 – Simulador de gastos

3.3 Hardware

O desenvolvimento do *hardware* para demonstração da comunicação com o *power monitor* envolve uma série de sensores e componentes eletrônicos. A plataforma de prototipagem eletrônica utilizada para a construção desse *hardware*, foi o NodeMCU(subseção 2.2.1). O principal sensor utilizado foi o SCT 013-000 (subseção 2.2.2), que tem o papel de aferir dados da corrente que passa pelos dispositivos ao longo do tempo que o mesmo se encontra ligado, também vale destacar o uso do relé que é responsável por toda a lógica de liga e desliga do dispositivo. O NodeMCU faz o intermédio da comunicação entre *hardware* e servidor *web*, fazendo toda a comunicação eletrônica entre o sensor e o circuito montado (Figura 26), e enviando os dados recebidos pelo sensor de corrente para o servidor *web* que por sua vez, salva as medições aferidas no banco de dados. Com relação aos dados enviados para o banco, foi feito um algoritmo que ao identificar que o dispositivo está ligado verificando a corrente elétrica no presente momento e quando o mesmo é desligado é aferido novamente a corrente elétrica, esse resultado é multiplicado pelo valor da tensão e assim é obtido o consumo do dispositivo (KWh). Após esse processo a informação é enviada via *websocket* para o servidor *web* que salva tudo no banco de dados, e por fim a informação é consumida pela interface *web*.

É válido lembrar que, para a comunicação *websocket* é necessário o *hardware* e o *software* estarem conectados na mesma rede *Wi-Fi*. O grande motivo para a escolha do NodeMCU como plataforma de prototipagem foi a sua fácil comunicação com uma rede *Wi-Fi*, o código a seguir é um exemplo de como estabelecer a comunicação com uma rede sem fio.

```
#include <NodeMCUWiFi.h>
const char* ssid = NOME DA REDE;
const char* password = SENHA;
```

```
WiFi.begin(ssid , password);
```

Após estabelecer a conexão o próximo passo será interligar o servidor *web* com o *hardware* através da comunicação por *websocket*, que será facilitada por meio da biblioteca *SocketIOClient*, ela fornece alguns métodos como: ***emit***¹, ***on***²e ***connect***³que ajudam no momento de concretizar a comunicação total do *hardware*. A seguir terá um exemplo de como usar os métodos citados em conjunto com o código anterior.

```
#include <SocketIOClient.h>
SocketIOClient socket;
const char* ssid = NOME DA REDE;
const char* password = SENHA;
String host = IP DO SERVIDOR WEB;
int port = PORTA QUE FOI FORNECIDA AO SERVIDOR WEB;

void led(String state) {
  Serial.println("[led] " + state);
  if (state == "\"state\":true") {
    socket.emit("post-informacao", "{ \"data\": \"1\" }");
  }
  else {
    socket.emit("post-informacao", "{ \"data\": \"0\" }");
  }
}

void setup() {
  WiFi.begin(ssid , password);
  socket.on("ligar", ligar);
  socket.connect(host , port);
}

void loop() {
  socket.monitor();
}
```

¹Função responsável por emitir os dados para o servidor *web*

²Função responsável por receber os dados do servidor *web*

³Função responsável por estabelecer conexão com servidor *web*

A Figura 26 mostra bem o circuito demonstrativo utilizado neste trabalho para exemplificar a comunicação com o *power monitor*. O esquema representa uma forma prática e simples de se usar o sensor de corrente, o SCT 013-000 é ativado quando o usuário liga a lâmpada no interruptor (representado por um botão) ou pelo sistema *web* (Figura 22), fazendo com que o relé ative e permita com que o dispositivo mude o estado para ligado. Após esse processo o sistema irá medir corrente que passa pelo dispositivo, e ao ser desligado o resultado do consumo do dispositivo irá ser transmitido via *websocket* para o servidor *web* e se juntará com o resultado do tempo em que o dispositivo permaneceu ligado para que por fim se possa se ter o consumo e o gasto deferido pelo equipamento.

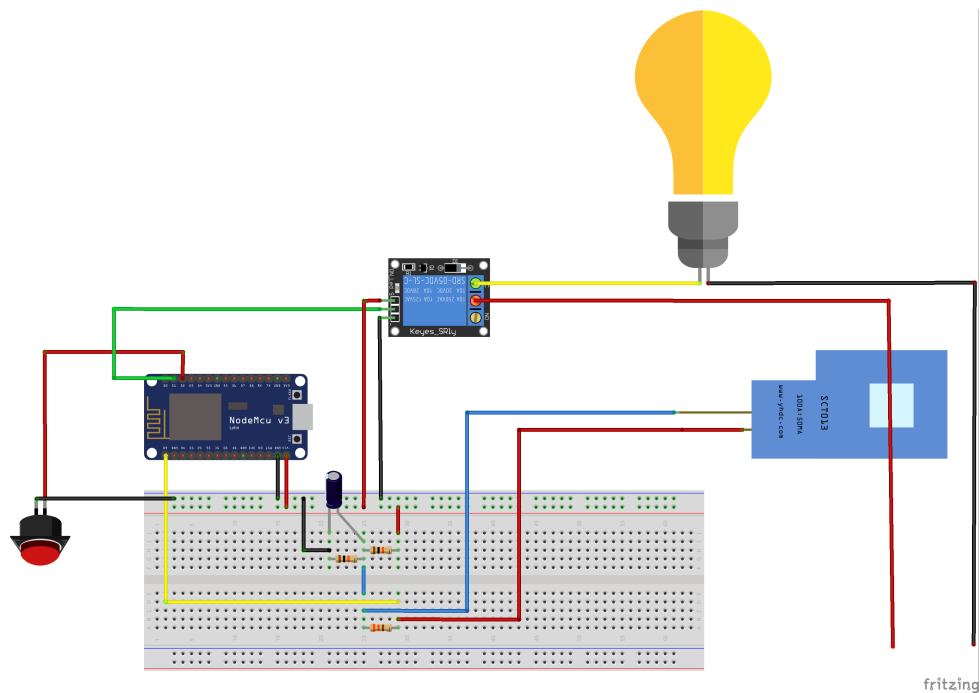


Figura 26 – Circuito demonstrativo para comunicação com o *power monitor*

3.4 Resultados

Depois de todo o estudo realizado, possibilidades discutidas e desenvolvimento detalhado, chegou-se ao funcionamento da primeira versão do *software power monitor*. O sistema disponibiliza meios para uma fácil comunicação com qualquer *hardware* de monitoramento de energia e uma interface que leva o usuário a ter um novo entendimento e uma nova conscientização sobre o gasto energético. Trazendo uma forma menos complicada de mensurar a quantidade de energia elétrica utilizada em uma residência, possibilitando também uma boa e simples leitura de quanto tem se gastado.

O *power monitor* foi pensado para ser uma solução simples e barata, onde qualquer pessoa que tenha os devidos conhecimentos possa instalar, comunicar com *hardwares* personalizados ou até mesmo modificar o código fonte. Como o *software* foi desenvolvido

para que todos possam usar e modificar, todo o código fonte está disponível na plataforma de hospedagem de código fonte chamada *GitHub* no seguinte endereço: <https://github.com/jonathanbraga/power-monitor>. A Tabela 1 mostra bem os custos do dispositivo que foram utilizados para a demonstração da comunicação com o *power monitor*.

Quantidade	Dispositivo	Preço (R\$)
2	Resistor 10k Ω	0,99
1	Resistor 330 Ω	0,99
1	Módulo Relé 5v	9,00
1	NodeMCU	59,76
1	SCT 013-000	58,80
1	Capacitor eletrolítico 100uF	2,47
1	Conjunto de 40 Jumpers	9,00
1	Protoboard	20,00
1	Chave gangorra 2 terminais	3,00
TOTAL		165,00

Tabela 1 – Custos do dispositivo de demonstração

4 CONCLUSÃO

Do início ao fim deste trabalho de conclusão de curso, alguns conceitos foram apresentados e demonstrados a respeito do consumo de energia elétrica, visando criar um cenário comparativo a respeito da situação energética do país. Ao longo dos anos o povo brasileiro ouviu sobre propostas na melhoria do meio ambiente, propostas sustentáveis e até melhoria na infraestrutura urbana. Porém na maioria das vezes nunca passou de apenas propostas.

Por meio do estudo realizado neste trabalho constatou-se que a leitura do consumo energético feita em tempo real e disponibilizada para o usuário de uma maneira entendível e direta, desperta a curiosidade e consequentemente a conscientização do mesmo. Apenas mostrar os resultados coletados não seria suficiente para a conscientização do cidadão brasileiro, a forma com que os resultados são relatados para o consumidor tem um grande impacto, pois no momento em que a unidade de medida do consumo de energia elétrica é "trocada" do quilowatt-hora para a moeda em circulação no país, o real, faz com que todos os brasileiros independente da classe social ou do grau de escolaridade entenda o quanto tem se consumido e desperdiçado em sua residência.

Este trabalho traz uma solução eficiente no monitoramento de energia elétrica, tanto no âmbito do *software* como na comunicação com *hardware*. O Sistema foi pensado de tal maneira que mesmo em que não exista a presença de um *hardware* para controlar os dispositivos físicos o utente possa, ainda sim, monitorar os seus gatos por meio das simulações que levam em conta o consumo esperado do dispositivo. A facilidade de estabelecer a comunicação entre servidor *web* e outros dispositivos físicos, traz a possibilidade do usuário poder sempre expandir e melhorar as formas de monitoramento de energia em sua residência. Além disso o trabalho implementa um novo padrão para se mensurar o consumo da energia elétrica nas residências, no lugar do complexo quilowatt-hora, medida que atualmente é utilizada para quantificação do consumo energético, o *power monitor* explora uma abordagem mais simples e mais perto do dia a dia de todo brasileiro, mensura todo o consumo de energia elétrica da residência em reais, moeda em circulação no país.

A troca do quilowatt-hora pela moeda real, faz com que o consumidor entenda a verdadeira situação da sua conta de energia, possibilitando um melhor planejamento financeiros e o incentivo a consumir apenas o que é necessário. Apesar do sistema atingir os resultados esperados, existem algumas melhorias que podem ser consideradas para trabalhos futuros. A primeira seria criar um *hardware* mas robusto, onde adicionaria um Arduino Uno que contribuiria com mais entradas analógicas e digitais, isso influenciaria em uma maior acoplação de relés que por sua vez influencia na quantidade de eletrodomésticos

adicionados ao *power monitor*, o arduino por sua vez se comunicaria via conexão serial com o NodeMCU. A Segunda melhoria seria em relação ao monitoramento da tensão em um dispositivo ligado, a inclusão de um sensor de tensão seria a melhor opção para poder se obter a potencial real que o dispositivo consome. A última mudança a ser destacada é na parte do *software*, melhorias em relação ao servidor *web* como por exemplo, maior segurança nos dados armazenados, possibilidade para criação de mais usuários, segurança nas transações de informação com o *hardware* e com a interface *web* e um aumento nos *end-points* que acarretaria em uma maior disponibilidade de dados. Na parte da interface *web* as melhorias seriam direcionadas a correção de *bugs* existentes na primeira versão e uma melhor estruturação para que se possibilite uma customização do usuário em relação aos dados em que ele deseja visualizar.

Em conclusão o sistema proposto por este trabalho, supre as necessidades de gerenciamento dos equipamentos eletrônicos presente em uma residência e de um melhor entendimento em relação aos gastos e desperdícios de energia elétrica.

REFERÊNCIAS

- ANDRADE, R. *Governo prevê até 2020 mais 24 hidrelétricas*. 2011. Disponível em: <<https://economia.estadao.com.br/noticias/geral,governo-preve-ate-2020-mais-24-hidreletricas-imp-,727860>>. 14
- BLINI, F. *Desenvolvimento de Aplicativos Móveis com Javascript: Ionic, React Native e NativeScript. Qual escolher?* 2016. Disponível em: <<https://goo.gl/h5ei9F>>. 18
- DAVID, W. *Handshake ou aperto de mão*. 2011. Disponível em: <<http://wdredes.blogspot.com/2011/05/handshake-ou-aperto-de-mao.html>>. 18
- EMPRESA DE PESQUISA ENERGÉTICA. *Anuário Estatístico de Energia Elétrica 2017 ano base 2016*. Brasília, 2017. 232 p. 13
- EMPRESA DE PESQUISA ENERGÉTICA. *Balanço Energético no Brasil Relatório Final*. Rio de Janeiro, 2017. 294 p. 14
- GOLDEMBERG, J. *Energia e desenvolvimento*. 1998. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-40141998000200002>. 14
- GOVERNO. *Energia para abastecer 40% do consumo residencial do País*. 2011. Disponível em: <<http://www.brasil.gov.br/editoria/infraestrutura/2011/12/energia-para-abastecer-40-do-consumo-residencial-do-pais>>. 14
- LONGEN, A. *O que é JavaScript? Conheça essa linguagem de programação*. 2018. Disponível em: <<https://www.weblink.com.br/blog/programacao/o-que-e-javascript/>>. 17
- MATTOS, L. *Crise Hídrica de 2015 pode ser a mais grave dos últimos anos*. 2015. Disponível em: <<https://www.otempo.com.br/capa/brasil/crise-h%C3%ADdrlica-de-2015-pode-ser-a-mais-grave-dos-%C3%BAltimos-anos-1.1042347>>. 13
- PATEL, P. *What exactly is Node.js?* 2018. Disponível em: <<http://www.brasil.gov.br/editoria/infraestrutura/2011/12/energia-para-abastecer-40-do-consumo-residencial-do-pais>>. 17
- REZENDE, R. *Desvendando a SQL*. 2013. Disponível em: <<https://www.devmedia.com.br/desvendando-a-sql-parte-1/28328>>. 18
- ROBOCORE. *Módulo WiFi - ESP8266*. 2014. Disponível em: <<https://www.robocore.net/loja/produtos/modulo-wifi-esp8266.html>>. 19