

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

LEONARDO BIELSKI DA SILVA
RAPHAEL MARTINS ROCHA

CONTROLE DE CONSUMO DE ENERGIA

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2018

LEONARDO BIELSKI DA SILVA
RAPHAEL MARTINS ROCHA

CONTROLE DE CONSUMO DE ENERGIA

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Augusto Foronda

PONTA GROSSA

2018



TERMO DE APROVAÇÃO

CONTROLE DE CONSUMO DE ENERGIA

Por

LEONARDO BIELSKI DA SILVA
RAPHAEL MARTINS ROCHA

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 22 de maio de 2018 como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação. Os candidatos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr Augusto Foronda
Orientador

Prof. Dr Richard Duarte Ribeiro
Membro titular

Prof. Dr. Diego Robert Antunes
Membro titular

Prof^a. Dr.^a Helyane Bronoski Borges
Responsável pelo Trabalho de Conclusão
de Curso

Prof. Dr. André Pinz Borges
Coordenador do curso

AGRADECIMENTOS

Agradecemos ao professor Dr. Augusto Foronda pelas orientações e ensinamentos que nos proporcionou nesse trabalho.

Agradecemos ao Leandro Augusto Kisielewicz analista de sistemas do protesto nacional, que nos ajudou na estrutura do banco de dados do projeto.

Agradecemos ao Ângelo Henrique Mino e Luciano Kisielewicz donos da empresa Linux Ponta de Ponta Grossa, por compartilhar o local de trabalho para testes e auxílios no código.

Eu, Leonardo Bielski da Silva, agradeço aos meus familiares, por compreenderem a minha ausência e pela paciência enquanto desenvolvíamos o projeto.

Eu, Raphael Martins Rocha, agradeço a minha noiva Ligia Bitencourt pela ajuda e paciência nos dias de estresse que nada dava certo, estando ao meu lado em cada momento de dificuldade.

Agradecemos a Universidade Tecnológica Federal do Paraná pelo ambiente de estudo e apoio durante todo curso. Pelos recursos tecnológicos e Professores extremamente qualificados que a instituição possui.

RESUMO

SILVA, Leonardo Bielski da; **Rocha**, Raphael Martins. **Controle de consumo de Energia**. 2018. 32 p. Trabalho de Conclusão de Curso de Tecnologia em Análise e Desenvolvimento de Sistemas - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2018.

Este trabalho tem como objetivo apresentar o desenvolvimento de uma ferramenta para auxiliar no controle de consumo de energia e água em um ambiente residencial, possibilitando ao usuário obter informações de consumo que não são possíveis através do medidor utilizado pelas empresas fornecedoras de energia e água, utilizando de ferramentas acessíveis de prototipação como o Raspberry Pi. Através do sensor esp12-e, coletando dados de consumo do ponto de energia, enviando para o Raspberry Pi armazenar e mostrar os dados no sistema desenvolvido para visualização e geração de relatórios. Demonstrando a configuração realizada no Raspberry e no esp12-e.

Palavras-chave: Rapsberry Pi. Controle. Energia. Consumo.

ABSTRACT

SILVA, Leonardo Bielski da; **Rocha**, Raphael Martins. **Energy Consumption Control**. 2018. 32 p. Final Paper(Graduation of Tecnology in Systems Analysis and Development)- Federal University of Technology - Paraná. Ponta Grossa, 2018.

This work aims to demonstrate the development of a tool to assist in the control of energy and water consumption in a residential environment, enabling the user to obtain consumption information that is not possible through the meter tool used by companies which provide energy and water, using accessible prototyping tools like the Raspberry Pi. Using an esp-12e sensor to collect a consumption data from an energy point, sends to the Raspberry Pi stores and displays the data in the system developed for viewing and reporting. Demonstrating the configuration made in Raspberry and esp12-e.

Keywords: Raspberry Pi. Energy. Control. Consumption.

LISTA DE ILUSTRAÇÕES

Figura 1–Diagrama do Projeto	12
Figura 2 – ESP12E NodeMCU	13
Figura 3 – Raspberry Pi 3.....	14
Figura 4 – Sensor ACS712	15
Figura 5 - Fluxo de Dados	16
Figura 6 – Função de Loop	17
Figura 7 – Código envia Dados	18
Figura 8 – Diagrama de Configuração	19
Figura 9 – Conexão Wi-Fi entre Raspberry Pi 3 e ESP12E	20
Figura 10 - Configuração DHCP no Raspberry	21
Figura 11 – Arquivo hostapd.conf.....	21
Figura 12 - Modelo Relacional do Banco	22
Figura 13 – Caso de Uso do Sistema.....	24
Figura 14 – Tela Principal do Sistema.....	25
Figura 15 – Diferença do Menu	25
Figura16 – Tela Relatórios	26
Figura 17 – Tela de Configuração de Taxas de Energia	27
Figura 18 - Tela Cadastro de Usuário	27
Figura 19 – Tela Principal Com Gráfico	28
Fotografia 1 – Protoboard com ESP12e e Sensor	23

LISTA DE SIGLAS

IoT	<i>Internet of Things</i>
GB	<i>Giga Byte</i>
LAN	<i>Local Area Network</i>

LISTA DE ACRÔNIMOS

WIFI	<i>Wireless Fidelity</i>
------	--------------------------

SUMÁRIO

1 INTRODUÇÃO	10
1.1 OBJETIVOS.....	11
1.1.1 Objetivo Geral.....	11
1.1.2 Objetivos Específicos.....	11
2 COMPONENTES DE HARDWARE DO PROJETO	12
2.1 MICRO CONTROLADOR ESP8266(ESP12E) NODEMCU.....	12
2.2 RASPBERRY PI	13
2.3 SENSOR.....	14
2.3.1 Sensor de Corrente Elétrica.....	15
3 PROGRAMAÇÃO DO MICROCONTROLADOR	16
3.1 LEITURA DOS DADOS ATRAVÉS DO ESP12E.....	16
3.2 ENVIANDO DADOS PARA O SERVIDOR	16
4 CONFIGURANDO O RASPBERRY PI 3.....	19
4.1 TOPOLOGIA DO SISTEMA.....	19
4.2 SISTEMA OPERACIONAL	19
4.3 DISPOSITIVOS DE REDE	20
4.4 SERVIDOR DHCP	20
4.5 CONFIGURANDO REDE WIFI COMO ROTEADOR.....	21
4.6 SERVIDOR WEB APACHE E PHP.....	22
4.7 BANCO DE DADOS MARIADB	22
5 PROTÓTIPO DO LEITOR DE DADOS DE ENERGIA	23
6 SISTEMA DE CONTROLE DE CONSUMO	24
6.1 MODELANDO O SISTEMA	24
6.2 INTERFACE.....	24
6.3 GERANDO GRÁFICOS	28
7 CONCLUSÃO.....	29
7.1 TRABALHOS FUTUROS	29
REFERÊNCIAS.....	31

1 INTRODUÇÃO

A cada dia, os recursos energéticos tornam-se mais caro e escasso devido a questões ambientais como o desmatamento ou o aquecimento global, aumentando a necessidade de controlar e monitorar seu consumo tanto em residências ou até mesmo na indústria.

A energia elétrica é um recurso que possibilita que as pessoas tenham água quente nos chuveiros e iluminação em suas residências, e seu desperdício tem um custo financeiro e ambiental muito alto. O uso racional deste recurso é fundamental para a redução destes custos (DUPONT, 2015).

A água doce, sendo um recurso indispensável para qualquer ser vivo, além de garantir a sobrevivência de qualquer forma de vida conhecida, é usada pelo homem, ajudando no desenvolvimento econômico e social, por exemplo, na agricultura, na geração de energia elétrica, pecuária, indústrias e recreação. Através da História, as civilizações sempre tiveram seu desenvolvimento cultural e econômico, atrelados a disponibilidade desses recursos (FARIAS, 2007).

O controle detalhado do consumo destes recursos não é fornecido para os usuários, em residências ou até mesmo empresas, surgindo muitas vezes a dúvida de onde está o maior gasto destes recursos em uma residência. O chuveiro ou o ar-condicionado, por exemplo, podem ser alguns dos dispositivos que consomem mais energia numa residência.

Descobrir através de um monitoramento qual ponto de uma residência, seja elétrico, gera o maior consumo de recursos, pode ajudar não somente na utilização consciente destes recursos, mas também no orçamento das famílias diminuindo os custos básicos de uma casa.

Existem tecnologias que podem ajudar nesse controle, tais como plataformas de prototipação Arduino e RaspberryPi. Essas plataformas de prototipação são ferramentas de baixo custo, que podem ser utilizadas em conjunto com sensores e programas de computador no desenvolvimento de um sistema que visa o monitoramento de consumo de energia em vários pontos de uma residência, analisando o consumo desse recurso.

Este trabalho propõe o desenvolvimento de um sistema que coleta dados através de sensores para o monitoramento do consumo de um ponto de energia, simulando uma residência, enviando esses dados para um sistema que irá tratá-los.

Um *software* foi desenvolvido para processar e analisar os dados coletados, resultando em informações de consumo.

Nesse sistema o usuário pode obter o consumo de energia em tempo real, como o consumo diário e mensal, podendo assim com base nos dados coletados, gerar gráficos para melhor representá-los, auxiliando o usuário a mensurar os gastos e monitorá-los separadamente.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver um sistema para monitorar e analisar o consumo de energia de uma residência.

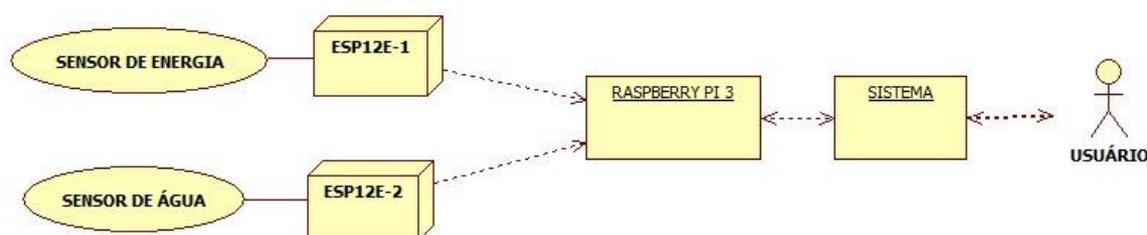
1.1.2 Objetivos Específicos

- Verificar qual plataforma de prototipação seria a mais adequada para o desenvolvimento do projeto;
- Montar e configurar o *hardware* necessário;
- Aplicar fórmulas matemáticas que convertessem os dados coletados de um sistema elétrico e um sistema hidráulico em valores numéricos e programá-las na plataforma de prototipação;
- Configurar e calibrar os sensores de fluxo de corrente elétrica;
- Programar e testar os algoritmos de obtenção e envio dos dados nos microcontroladores;
- Realizar a modelagem de um sistema de banco de dados para o armazenamento dos dados gerados pelos sensores;
- Realizar a modelagem do sistema;
- Desenvolver um sistema *Web* usando linguagens de programação para o tratamento dos dados gerados pela plataforma de prototipação, exibindo relatórios, resultados e acesso a esses dados via Internet;

2 COMPONENTES DE HARDWARE DO PROJETO

O projeto consistiu em ligar dois sensores, sendo um de energia e outro de água em micro controladores ESP12E, os quais serão responsáveis por enviar os dados coletados ao sistema localizado no Raspberry Pi 3, que coleta os dados, realiza as operações necessárias e exibe-os ao usuário, conforme mostra a figura 1.

Figura 1–Diagrama do Projeto



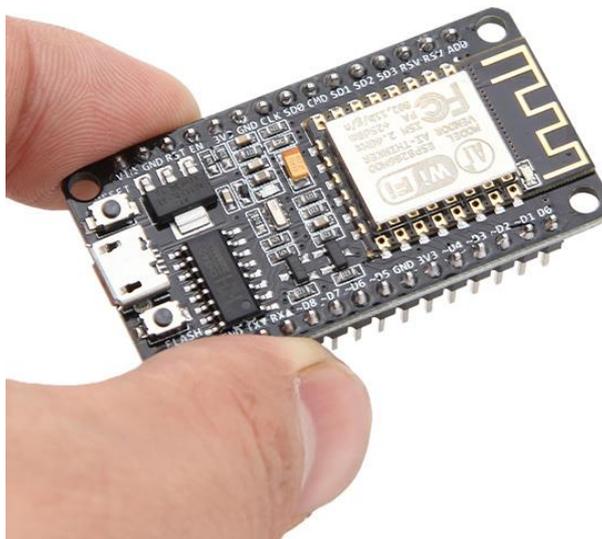
Fonte: Autoria Própria

2.1 MICRO CONTROLADOR ESP8266(ESP12E) NODEMCU

O microcontrolador ESP8266-12E NODEMCU, figura 2, é uma plataforma de prototipação desenvolvida pela empresa Shenzhen Doctors of Intelligence & Technology (SZDOIT), baseada em um *Hardware* de baixo consumo de energia concebido para auxiliar no desenvolvimento de sistemas IoT. Amplamente utilizado no desenvolvimento de aplicações para casas inteligentes, comunicação de rede local, controle industrial, entre outras aplicações (SZDOIT,2015).

Por possuir características como tamanho reduzido e comunicação *Wireless* nativamente no dispositivo, ser similar ao Arduino em comunicação com sensores e outros dispositivos e programação, além de possuir um baixo custo no mercado, o dispositivo foi o escolhido para o desenvolvimento do projeto.

Figura 2 – ESP12E NodeMCU



Fonte: Filipeflop (2018)

O Quadro 1 mostra as especificações gerais do ESP8266.

Quadro1 – Especificações Gerais do ESP8266-12E NodeMCU

Voltagem	3.3 V
Consumo de Corrente	10 μ A
Memória Flash	16MB max (512k normal)
Processador	Tensilica L106 32 bits
Velocidade do Processador	80-160MHz
RAM	32k+80k
GPIOs	17(Multiplexada com outras funções)
Analógico para Digital	1 entrada com 1024 de resolução
Suporte 802.11	b/g/n/d/e/i/k/r
Máxima Corrente de conexão TCP	5

Fonte: USO DO MICROCONTROLADOR ESP8266 PARA AUTOMAÇÃO RESIDENCIAL, (2017)

2.2 RASPBERRY PI

De acordo com o UPTON (2017), o Raspberry Pi é um microcomputador de baixo custo e tamanho reduzido, que pode ser plugado em um monitor ou uma TV e utilizado com um teclado e mouse. Possui plataformas para o auxílio no aprendizado de algumas linguagens de programação como o Python exemplo. Além de ser capaz

de realizar operações que são feitas em um computador pessoal comum, assistir a filmes, navegar na Internet, redigir documentos e ouvir músicas. Figura 3.

Figura 3 – Raspberry Pi 3



Fonte: Raspberry Pi, (2018)

O Quadro 2 demonstra as especificações gerais do Raspberry Pi 3.

Quadro2 – Especificações Gerais do Raspberry Pi 3 Model B

PROCESSADOR	Broadcom BCM2387 chipset.1.2GHz Quad-Core ARM Cortex-A53
MEMÓRIA	1GB LPDDR2
GPU	Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG and 1080p30 H.264 high-profile decode
ARMAZENAMENTO	Cartão MicroSD
CONEXAO DE REDE	10/100 BaseT Ethernet socket
CONEXAO WI-FI	802.11 b/g/n Wireless LAN
SISTEMA OPERACIONAL	Boot do MicroSD, usando uma versão Linux ou Windows 10 IoT
DIMENSÕES	85X56X17mm
FONTE DE ENERGIA	Entrada Micro USB 5v, 2.5 ^a

Fonte: RS Components, (2018)

2.3 SENSOR

Para que possa ser realizada a leitura dos dados de consumo de energia é necessário um sensor que obtém esses dados e os transmite ao micro controlador através de uma porta de leitura de dados que pode ser digital ou analógica, dependendo da saída de dados do sensor.

2.3.1 Sensor de Corrente Elétrica

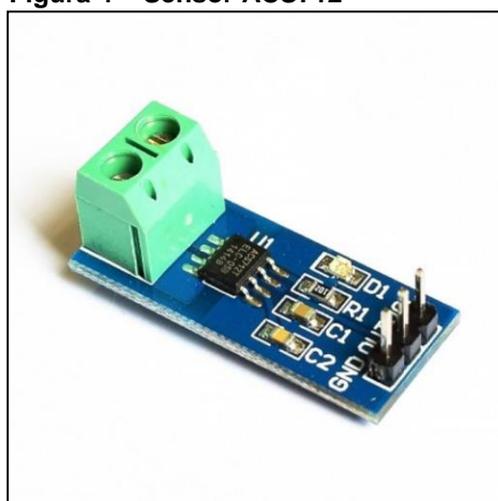
Para a realização da leitura da corrente que trafega em uma tomada ou em um ponto de energia como um chuveiro ou uma lâmpada, torna-se necessário medir qual a amperagem utilizada pelo aparelho no momento em que ele está conectado e consumindo energia.

Existem no mercado vários tipos de sensores que realizam a leitura desse consumo em amperes, sendo eles classificados como invasivos, ou seja, necessitam de ligação direta na rede elétrica, havendo a necessidade de modificação do sistema elétrico, realizando um corte no fio conectado ao ponto ao qual a medição será realizada.

Existem, também, os do tipo não invasivo, pelos quais basta colocar o sensor em volta do fio ao qual necessita realizar a medição.

O sensor de corrente ACS712, figura 4 utilizado no projeto é o modelo invasivo de capacidade de leitura de até 5A, que possui uma saída de dados de 5V aos quais iniciam a leitura em 2.5V, sendo que valores acima de 2.5V são considerados valores positivos e abaixo de 2.5V são valores negativos. Os dados são enviados através de sinal analógico, variando conforme a corrente que circula através do sensor.

Figura 4 – Sensor ACS712



Fonte: Eledrodex, (2018)

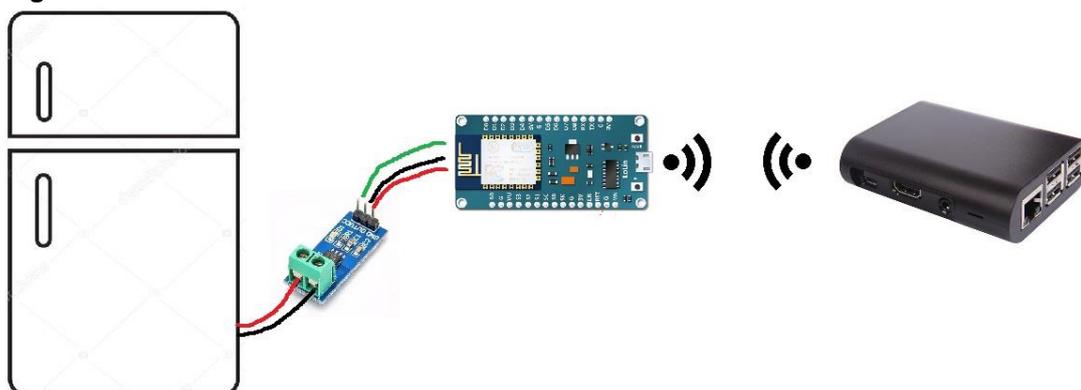
3 PROGRAMAÇÃO DO MICROCONTROLADOR

O Micro Controlador ESP12E pode ser programado em uma linguagem conhecida como eLua, derivada da linguagem lua, adaptada para sistemas embarcado (NodeMCU) e possibilita também utilizar linguagem C. Lua é executada via interpretação de *bytecodes* para uma máquina virtual baseada em registradores, e tem gerenciamento automático de memória. Essas características fazem de Lua uma linguagem ideal para configuração, automação (*scripting*) e prototipagem rápida (IERUSALIMSKY, 2015).

3.1 LEITURA DOS DADOS ATRAVÉS DO ESP12E

Para a realização da leitura dos dados, o ESP12E possui uma entrada analógica pela qual os dados obtidos pelos sensores serão recebidos. Após a obtenção dessa informação, o micro controlador, realizará os cálculos para a conversão dos dados de forma a ser útil e enviará os mesmos ao Raspberry com o banco de dados, como mostrado na figura 5.

Figura 5 - Fluxo de Dados



Fonte: Autoria Própria

3.2 ENVIANDO DADOS PARA O SERVIDOR

Como o ESP12E funciona em sistema de *loop* em seu código, faz-se necessário somente a programação de um sistema que submeta os dados ao servidor após a leitura do sensor, determinando o tempo de espera para que cada

dado seja enviado. Na Figura 6 podemos verificar que o *loop* está programado para aguardar um segundo para que seja reexecutado. (*delay(1000)*). Essa tela de informações só aparece com o *software* do *Arduino* utilizado.

Mostra também como foi implementada a função para capturar os dados através do sensor ACS 712, armazenando os mesmos em uma variável chamada *leitura*, convertendo essa leitura que vem do pino analógico, variando entre 0 e 1024, convertendo em volts e armazenando na variável *voltagem*. Após a conversão para volts, a mesma é convertida novamente para amperes e armazenada na variável *amps*. Essa leitura é realizada por dez segundos, após esses dez segundos é calculada a média da leitura e enviada ao servidor.

Figura 6 – Função de Loop

```
void loop() {
  ArduinoOTA.handle();
  leitura=analogRead(analogInput);
  voltagem=(leitura/1024)*3200;
  amps = (((voltagem - entradaMinima) / mVporAmpere));
  Serial.print("Sensor: ");
  Serial.println(leitura);
  Serial.print("Voltagem: ");
  Serial.println(voltagem);
  Serial.print("Saida Amp: ");
  Serial.println(amps);
  //enviaDados();
  delay(1000);
  media = media + amps;
  controle=controle+1;
  if(controle == 10){
    media = media/10;
    enviaDados();
    controle = 0;
    Serial.print("media: ");
    Serial.println(media);
  }
}
```

Fonte: Autoria Própria

O procedimento *enviaDados()*, faz uma requisição para o servidor enviando os dados obtidos pelo cálculo da média das 10 ultimas leituras enviando para o servidor a média calculada e o número de identificação do sensor, figura 7.

Figura 7 – Código envia Dados

```
void enviaDados(){
    if(WiFi.status()== WL_CONNECTED){//Checa a conexão Wifi Se Conectado
        String mensagem="dados="+((String)media+"&sen_id="+sen_id);//Monta Mensagem
        HTTPClient http;//Declare object of class HTTPClient
        http.begin("http://192.168.1.13/Projeto/regraNegocios/request.php");//Inicia conexão
        http.addHeader("Content-Type", "application/x-www-form-urlencoded");//Adiciona cabeçalho
        int httpCode = http.POST(mensagem);//Envia os dados
        String payload = http.getString();//Lê o payload request
        Serial.println(httpCode); //Imprime o código retornado
        Serial.println(payload); //Imprime o payload request
        http.end(); //Fecha a conexão
    }else{
        Serial.println("Erro na conexão WiFi");
    }
}
```

Fonte: Autoria Própria

Executando o procedimento *enviaDados()*, realiza-se uma verificação para assegurar que o ESP12E está conectado ao Raspberry, se a condição for verdadeira, o procedimento monta a mensagem a ser enviada ao servidor em uma variável chamada “*mensagem*”.

Declara uma variável do tipo *HTTPClient* com o nome *http*, chama a função *http.begin()* inserindo na variável “*http*” o caminho do servidor que irá receber a mensagem, com a função *http.addHeader*, adiciona um cabeçalho na mensagem informando qual o tipo de dados está sendo enviado ao servidor, envia a mensagem com a função *http.POST(mensagem)*, colocando o resultado desse envio em uma variável declarada como *httpCode*. Imprime no console de monitoramento do ESP12E as mensagens retornadas pelo servidor ou as mensagens de erro de entrega e finaliza o envio com *http.end()*.

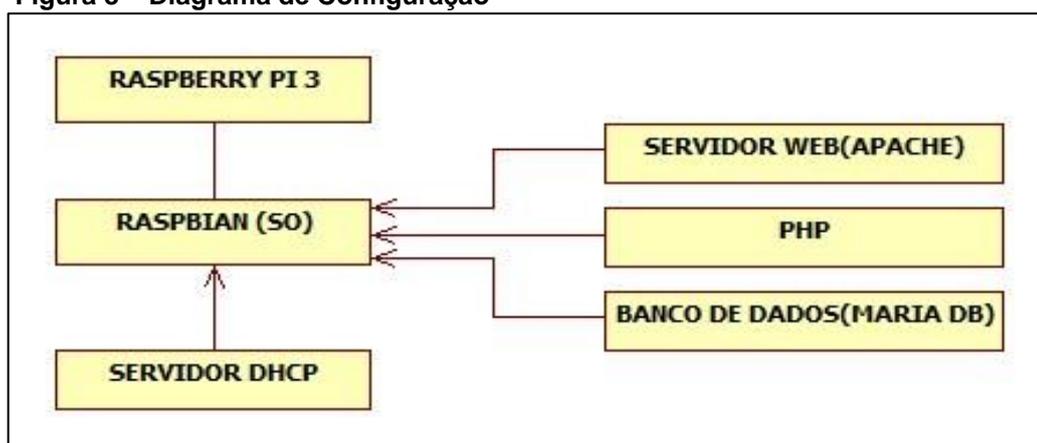
4 CONFIGURANDO O RASPBERRY PI 3

Este capítulo demonstra superficialmente a configuração utilizada no Raspberry Pi 3.

4.1 TOPOLOGIA DO SISTEMA

A figura 9 ilustra a configuração realizada no Raspberry Pi 3 para que o Sistema de controle de consumo funcione corretamente. Os dados coletados pelo sensor são enviados para o controlador ESP12E, que realiza os cálculos necessários para converter os valores coletados pelo sensor e envia para o Raspberry. O Raspberry possui um servidor *web* (apache) interpretando os códigos PHP, que por sua vez envia dados para o banco Maria DB.

Figura 8 – Diagrama de Configuração



Fonte: Autoria Própria

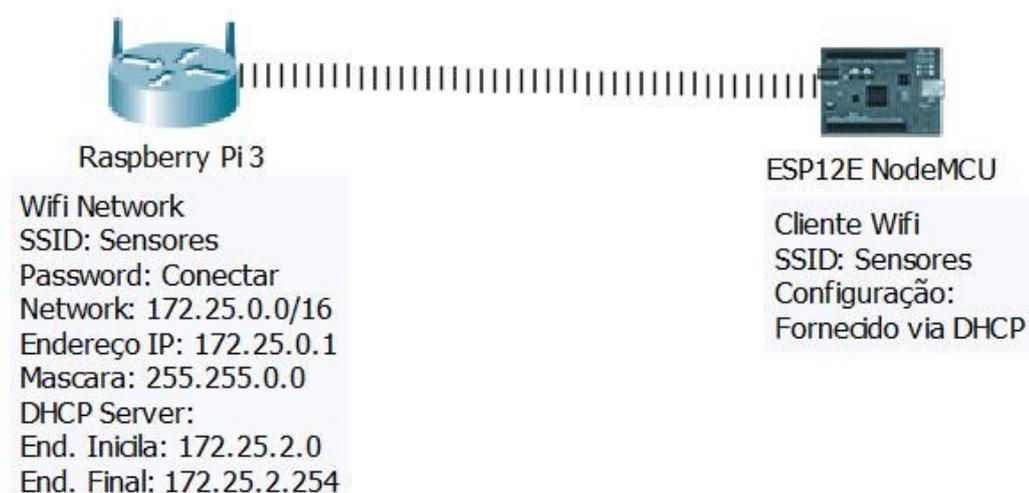
4.2 SISTEMA OPERACIONAL

O Sistema Operacional Raspbian, disponível para *download* no site da Raspberry Pi Foundation é um sistema de código fonte aberto, gratuito, que possui vários recursos, possibilitando a utilização do dispositivo em várias finalidades, como *desktop* pessoal, sistemas para ensino e navegação na Internet. Utilizando os recursos disponíveis no Raspbian, pode-se configurar servidores de aplicação, servidores de bancos de dados entre outros. Sendo assim, o Raspberry foi utilizado na elaboração do projeto (UPTON, 2017).

4.3 DISPOSITIVOS DE REDE

O Raspberry Pi 3 possui duas interfaces de rede, sendo uma interface cabeada e uma sem fio. Para a comunicação dos sensores com o Raspberry uma conexão sem fio própria do sistema foi configurado como servidor de rede *Wireless*, fornecendo uma comunicação direta entre os equipamentos de leitura de dados ESP12E e o servidor da aplicação, que roteia um *WiFi* diferente de outras redes sem fio evitando conflito de *IP (Internet Protocol)*.

Figura 9 – Conexão Wi-Fi entre Raspberry Pi 3 e ESP12E



Fonte: Aatoria Própria

4.4 SERVIDOR DHCP

Como o Raspberry possui uma interface de rede sem fio integrada, pode-se configurar o mesmo como ponto de acesso *WiFi* para conectar os sensores diretamente a ele, bastando instalar o pacote *isc-dhcp-server* para que o mesmo realize a distribuição de endereços IP (*Internet Protocol*) aos sensores, facilitando a integração entre os dispositivos.

O Arquivo de configuração do *isc-dhcp-server* encontra-se no caminho */etc/dhcp/dhcpd.conf* do Raspbian. Esse arquivo foi editado para as configurações conforme demonstra a figura 10.

Figura 10 - Configuração DHCP no Raspberry

```
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

authoritative;

log-facility local7;

subnet 172.25.0.0 netmask 255.255.0.0 {
    range 172.25.2.0 172.25.0.254;
    option domain-name-servers ns1.dobitaobyte.lan;
    option domain-name "dobitaobyte.lan"
    option routers 172.25.0.1;
    option broadcast-address 172.25.255.255;
    default-lease-time 600;
}
```

Fonte: Autoria Própria

4.5 CONFIGURANDO REDE WIFI COMO ROTEADOR

Para a comunicação direta entre o ESP12e e o Raspberry Pi 3, fez-se necessário configurar a interface de rede sem fio do Raspberry como roteador *WiFi*.

A figura 11 demonstra a configuração do arquivo *hostapd.conf* para a transformação do Raspberry em Roteador.

Figura 11 – Arquivo hostapd.conf

```
interface=wlan0
driver=nl80211
ssid=Sensores
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=Conectar
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Fonte: Autoria Própria

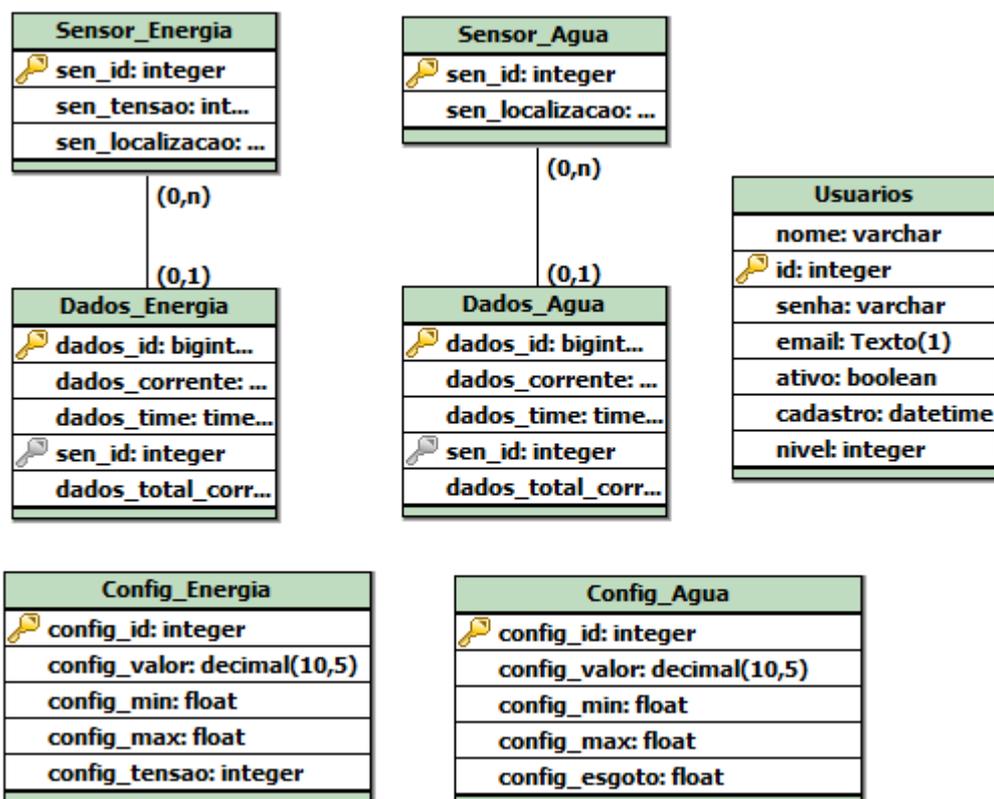
4.6 SERVIDOR WEB APACHE E PHP

Para disponibilizar a aplicação e torná-la acessível via navegador de Internet, é necessário um servidor de páginas *web*. O Apache é um interpretador de páginas *web* leve e eficiente, de fácil instalação, interpreta páginas *html* (*Hyper Text Markup Language*) estáticas, tornando necessária somente a instalação de módulo para a utilização de *scripts* PHP, para acessar bancos de dados como MySQL(MORIMOTO, 2008), tornando-o selecionado para hospedar o sistema.

4.7 BANCO DE DADOS MARIADB

Para o armazenamento dos dados, foi utilizado o MariaDB, como o esquema demonstrado na figura 12.

Figura 12 - Modelo Relacional do Banco



Fonte: Autoria Própria

5 PROTÓTIPO DO LEITOR DE DADOS DE ENERGIA

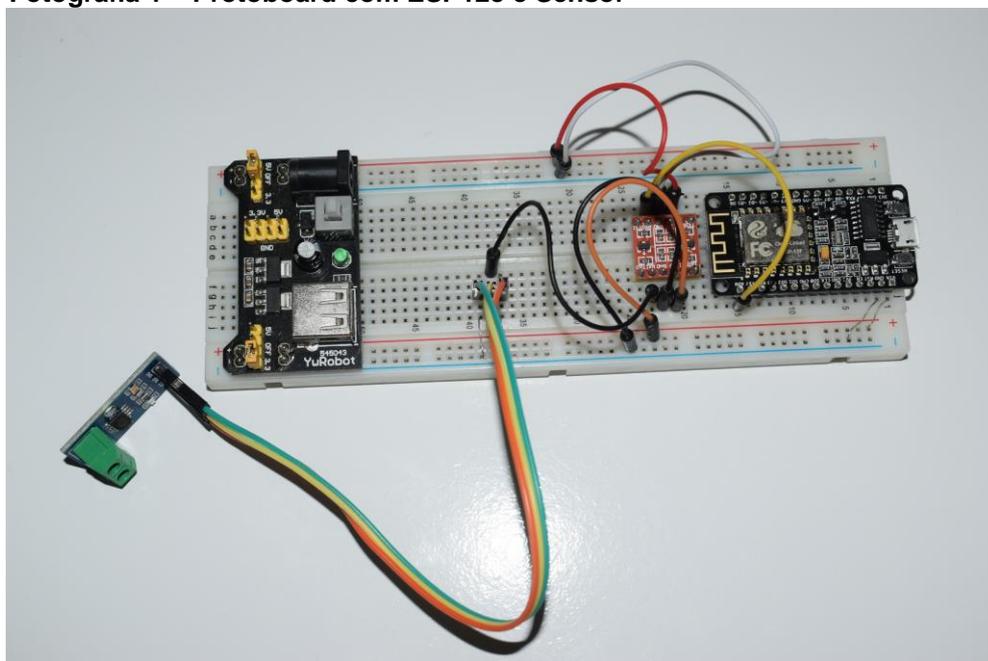
O protótipo do leitor de dados que obtém a corrente que passa em um ponto de energia foi montado em uma protoboard de 830 pontos, utilizando-se de 10 jumpers, sendo 6 deles do tipo bananinha e 4 feitos com fios condutores.

Ligado ao sensor foi utilizada uma extensão de energia com uma tomada fêmea em uma ponta e uma tomada macho na outra, e em um dos cabos foi ligado o sensor de corrente ACS712 para a realização da leitura da corrente que trafega na extensão.

Como a tensão de trabalho do acs712 é de 5V na saída de dados e possui corrente do tipo analógica, o ESP12e tem somente um pino que recebe sinais analógicos e o mesmo suporta o máximo de tensão em 3.3V, faz-se necessário a montagem de um regulador de voltagem que reduza os 5V recebidos do ACS712 para 3.3V. Esse conversor pode ser desenvolvido utilizando-se de resistores ou podem ser adquiridos na Internet algumas plataformas prontas para o devido fim.

A fotografia 1 mostra a protoboard com suas respectivas peças montadas.

Fotografia 1 – Protoboard com ESP12e e Sensor



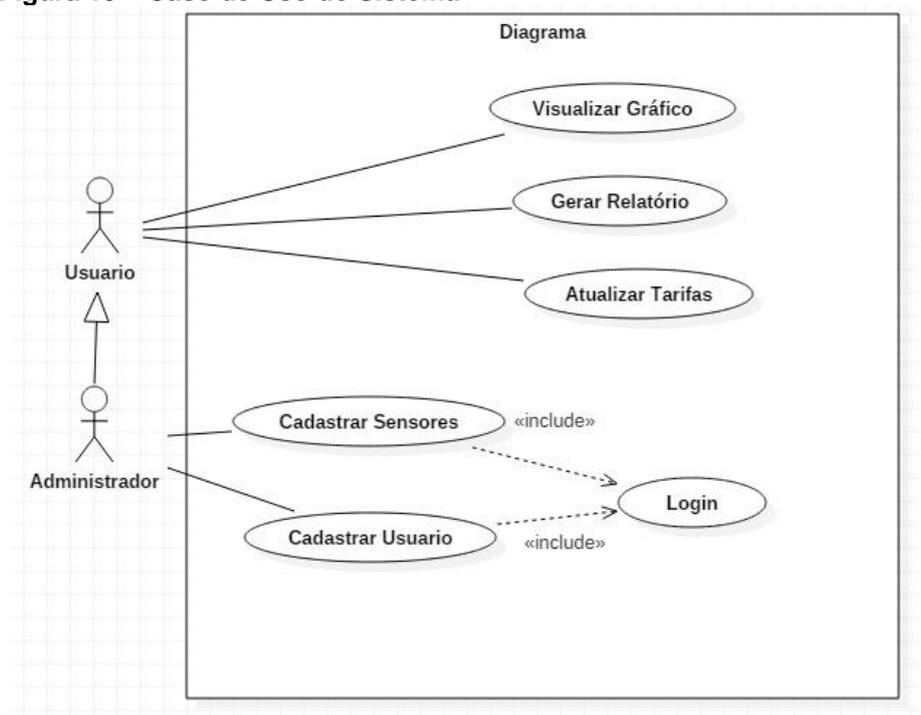
Fonte: Autoria Própria

6 SISTEMA DE CONTROLE DE CONSUMO

6.1 MODELANDO O SISTEMA

A figura 13, demonstra a modelagem inicial do sistema para trazer um entendimento do funcionamento e quais as capacidades de interação com o mesmo.

Figura 13 – Caso de Uso do Sistema

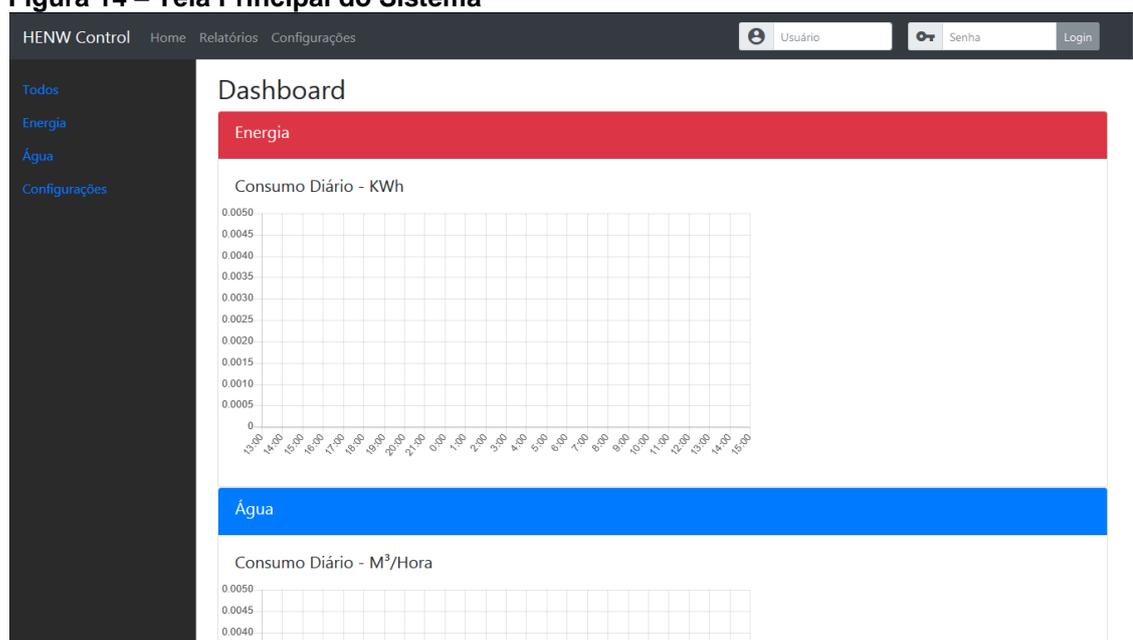


Fonte: Autoria Própria

6.2 INTERFACE

O sistema de controle de consumo exibe em sua tela principal um gráfico que indica o consumo das últimas 24 horas em um gráfico, calculando o consumo médio de hora em hora e mostrando os resultados na tela, como mostra a figura 15. O sistema exibe, também, um menu de opções que direcionam para a tela de relatórios de consumo e para as configurações do sistema, além de campos de *login*, que permite ao usuário autenticado acessar outras funcionalidades do sistema.

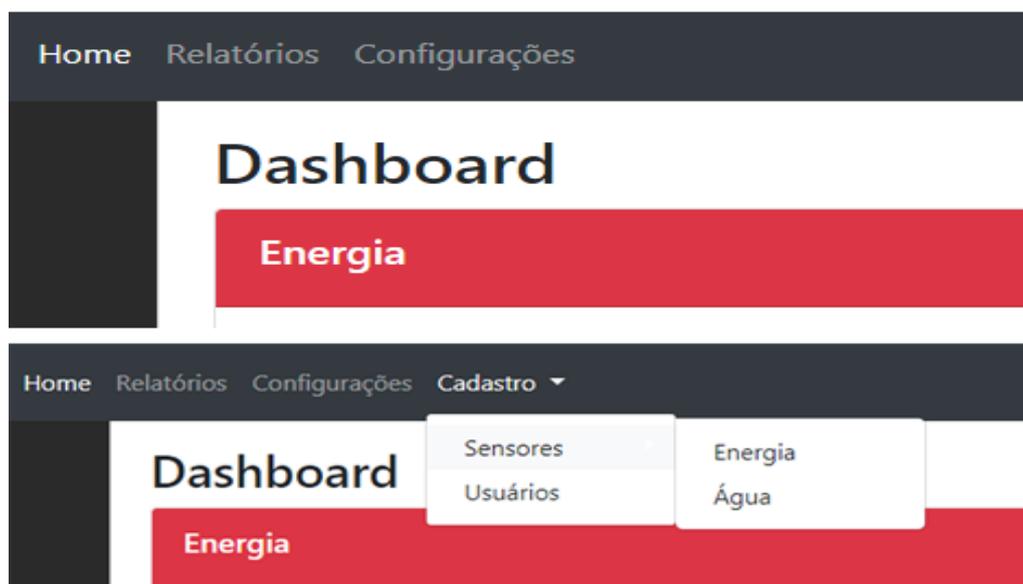
Figura 14 – Tela Principal do Sistema



Fonte: Autoria Própria

Após o usuário efetuar *login* no sistema, o mesmo recebe as permissões para adicionar outros usuários, além de cadastrar sensores ao sistema para que o mesmo possa armazenar os dados no banco de dados e exibir suas informações no gráfico, além de incluir os mesmos nos relatórios, como mostra a figura 15.

Figura 15 – Diferença do Menu



Fonte: Autoria Própria

Por questões de segurança, as configurações de adições de sensores só poderão ser realizadas por usuários com credenciais no sistema, evitando assim que usuários comuns do sistema alterem essas configurações fazendo com que o sistema não funcione corretamente.

Ao acessar a tela de Relatórios, o usuário tem a possibilidade de emissão de relatórios de consumo de energia ou de água. Esses relatórios podem ser emitidos diariamente, mensalmente, anualmente ou por períodos selecionados pelo usuário, como mostra a figura 16.

Figura16 – Tela Relatórios

Na tela de configurações, o usuário tem a possibilidade de configurar os parâmetros das taxas aplicadas pelas empresas de fornecimento de energia e água, tendo em vista que ocorrem alterações regularmente, principalmente nas taxas de energia. As taxas também são usadas para realizar os cálculos de consumo em valores, que são exibidos nos relatórios gerados pelo sistema.

Essa taxa de energia deve ser obtida com a fornecedora. A fornecedora utilizada como modelo de taxas de energia utilizada no desenvolvimento do trabalho foi a Copel em sua modalidade Tarifa Convencional – Subgrupo B1, que possui as tarifas descritas no seu site conforme o quadro 3.

Quadro1 – Tarifa Convencional – Subgrupo B1

CONVENCIONAL	Resolução ANEEL Nº 2.255, De 20 de junho de 2017	
	Resolução ANEEL	Com Impostos: ICMS e PIS/COFINS
Tarifa em R\$/kWh		
B1-Residencial Baixa Renda		
Consumo mensal inferior ou igual a 30 kWh	0,14510	0,15645
Consumo mensal entre 31 kWh e 100 kWh	0,24874	0,39024
Consumo mensal entre 101 kWh e 220 kWh	0,37310	0,58535
Consumo mensal superior a 220 kWh	0,41456	0,65039
Obs: Consumo até 30 kWh isento de ICMS		

Fonte: Copel, (2017)

A figura 17 demonstra a tela de configuração das taxas, modelado conforme a tabela de tarifas do subgrupo B1.

Figura 17 – Tela de Configuração de Taxas de Energia

Configurações

Tarifas de Energia:

Consumo até 30KWh - Valor:

Consumo de 30KWh a 100KWh - Valor:

Consumo de 101KWh a 220KWh - Valor:

Consumo acima de 220KWh - Valor:

Fonte: Autoria Própria

Conforme descrito anteriormente, o sistema possui mecanismos para cadastro de usuários afim de limitar o acesso, a adição, a remoção ou a alteração dos sensores do sistema, assim como usuários que possuem essas permissões, como mostra a figura 18.

Figura 18 - Tela Cadastro de Usuário

HENW Control Home Relatórios Configurações Cadastro Bem Vindo, administrador! LOGOUT

Cadastro de Usuários

Novo Usuário

5 resultados por página Pesquisar

ID	Nome	E-Mail	Usuário	Status	Nível de Acesso	Data do Cadastro	Ações
1	administrador	admin@localhost	admin	1	Administrador	2018-03-26 22:53:05	Editar Excluir

Mostrando de 1 até 1 de 1 registros

Anterior 1 Próximo

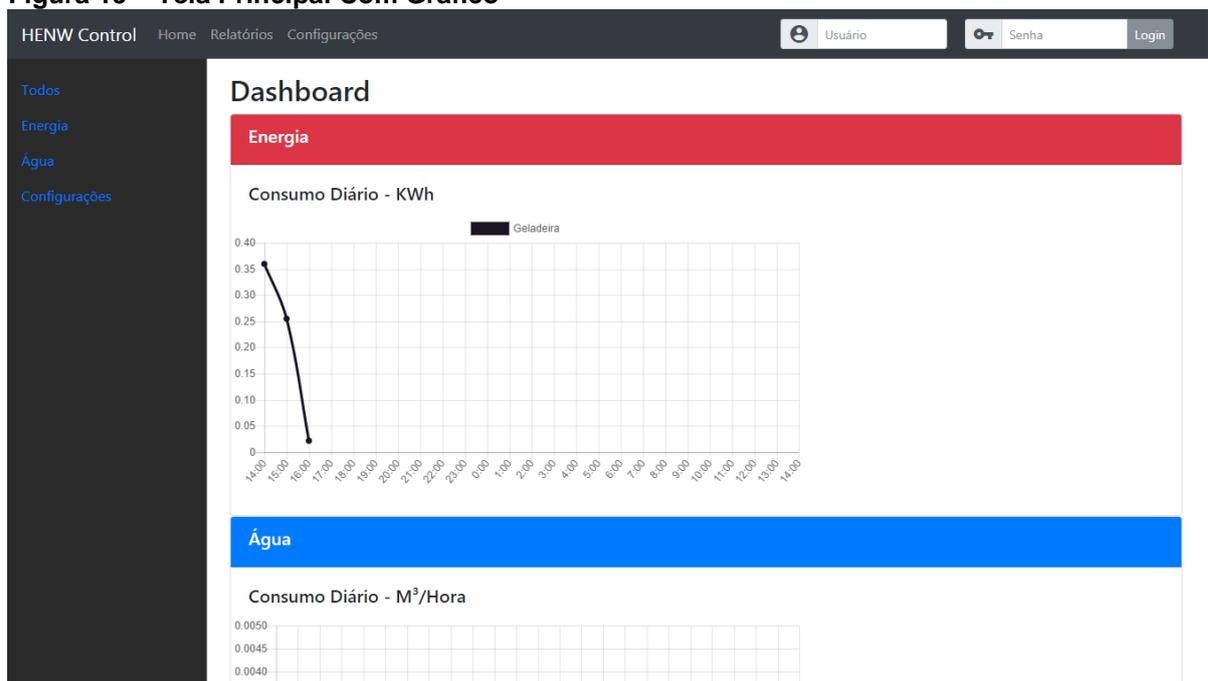
Fonte: Autoria Própria

6.3 GERANDO GRÁFICOS

O sistema tem como principal objetivo exibir dados que não são fornecidos pelas fornecedoras de energia ou de água. Para que os dados se tornem de fácil leitura, a opção escolhida para a exibição dos mesmos na tela do sistema foi através de gráficos que exibem o consumo dos recursos mensurados por hora, mostrando o quanto cada ponto de energia que possui um sensor atrelado a ele, está consumindo.

Utilizando-se da ferramenta *ChartJS*, disponível para download no site <http://www.chartjs.org> este mantido por vários colaboradores, possui uma licença de código aberto, gráficos são gerados demonstrando para cada sensor cadastrado no sistema, o consumo de seu respectivo recurso, sendo exibido na tela principal do sistema os dados das últimas 24 horas, como mostra a figura 19.

Figura 19 – Tela Principal Com Gráfico



Fonte: Aatoria Própria

7 CONCLUSÃO

O desenvolvimento do projeto apresentado, através de análises de gastos de energia, mostrou que pode retornar dados mais apurados do que os das empresas que nos fornecem esses recursos. Tais dados são apresentados por pontos mais específicos, o que permite uma melhor análise de consumo.

Este mostrou também, ferramentas básicas para o desenvolvimento de um sistema funcional de controle de consumo de energia e água, possibilitando realizar melhorias no sistema implantado.

O estudo aplicado pode ter melhorias futuras para conseguir um resultado ainda mais preciso. A oscilação de energia vinda da corrente alternada poderia ser medida através de um sensor de tensão, não precisando se fixar a tensão de entrada como foi feito neste projeto, pois apresentava resultados negativos de consumo elétrico.

Outra melhoria seria localizar uma fonte de energia que alimentasse o protótipo de forma correta, pois foram realizamos ajustes nos cálculos de consumo para que a obtenção dos dados ocorresse corretamente.

Os cálculos matemáticos feitos através de *select* do banco de dados com consumos anteriores e modelagem do banco teve um grande nível de complexidade, porém em trabalhos futuros pode ser aplicado mais uma tabela para fazer um resumo diário de consumo, agilizando assim as buscas. Já que foi verificado que somente com um sensor foram obtidos aproximadamente oito mil dados em apenas 24 horas.

7.1 TRABALHOS FUTUROS

O ESP12E com sua capacidade de controle e *WiFi* disponível para conectar ao Raspberry, fez sua função para esse projeto, assim como o Raspberry com suas funções de utilizar um banco de dados e um servidor *Web*. Ambos foram utilizados no limite, para trabalhos futuros com mais de três sensores é necessário rever suas limitações.

Desenvolver o sistema completo para adicionar as funcionalidades de consumo de água.

Aprimorar os sistemas de alimentação do esp12e para melhorar a precisão da leitura obtida.

Remodelar o banco de dados para otimizar o sistema na captura e leitura dos dados assim como sua exibição.

REFERÊNCIAS

IERUSALIMSCHY, ROBERTO. **Programando em Lua**. 3. Ed. Rio de Janeiro: LTC 2015.

MCROBERTS, MICHAEL. **Arduino Básico**. 2. Ed. Santa Terezinha: Novatec 2015.

COPEL. Disponível em: <<http://www.copel.com/hpcopel/root/nivel2.jsp?endereco=%2Fhpcopel%2Froot%2Fpagcopel2.nsf%2F5d546c6fdeabc9a1032571000064b22e%2Fc28b22b01ad9182403257488005939bb>>. Acesso em abril de 2018.

COSTA, ALYSSON P. DA; SERMANN, FERNANDA C; SILVA, GUSTAVO G. DA. **Desenvolvimento de um Protótipo para Medição de Energia Elétrica**. Disponível em: <http://nupet.daelt.ct.utfpr.edu.br/tcc/engenharia/doc-equipe/2015_1_06/2015_1_06_final.pdf>. Acesso em: 02 mai. 2018.

ELETRODEX. Disponível em: <<http://www.eletrdex.com.br/sensor-de-corrente-acs712-30a-a-30a.html>>. Acesso em: 06 abr. 2018.

FARIAS, TALDEN Q. **Direito à água e sustentabilidade hídrica**. In: Âmbito Jurídico, Rio Grande, X, n. 48, dez 2007. Disponível em: <http://www.ambito-juridico.com.br/site/index.php?n_link=revista_artigos_leitura&artigo_id=4538>. Acesso em: 05 ago. 2017.

FILIFELOP. Disponível em: <<https://www.filipeflop.com/produto/modulo-wifi-esp8266-nodemcu-esp-12/>>. Acesso em: 05 abr. 2018.

GITHUB. Disponível em: <<https://github.com/esp8266/Arduino/issues/3181>>. Acesso em: 02 mai. 2018.

GOMES, CHRISTINE H; GOMES, CYNTHIA C. M; TAGLIARI, SIMONE DE L. **Sistema Supervisório para Acompanhamento de Consumo de Energia Elétrica Residencial**. Disponível em: http://nupet.daelt.ct.utfpr.edu.br/tcc/engenharia/doc-equipe/2015_1_04/2015_1_04_final.pdf>. Acesso em: 02 mai. 2018.

MORIMOTO, CARLOS E. **Servidores Linux: Guia Prático**. 1. Ed. Porto Alegre: Sul Editores 2008.

NODEMCU. Disponível em: <<https://nodemcu.readthedocs.io/en/master/>>. Acesso em: 11 nov. 2017.

OLIVEIRA, SÈRGIO. **Internet das Coisas com Esp8266, Arduino e Raspberry Pi.** 1. Ed. Santa Terezinha: Novatec 2017.

RASPBERRY Pi. Disponível em: <<https://www.raspberrypi.org/>>. Acesso em: 12 nov. 2017.

DUPONT, FABRÍCIO H; GRASSI, FERNANDO; ROMITTI, LEONARDO. Energias Renováveis: buscando por uma matriz energética sustentável. **Revista Eletrônica em Gestão, Educação e Tecnologia Ambiental**, Santa Maria, 18 ago. 2015. Disponível em : <<https://periodicos.ufsm.br/reget/article/viewFile/19195/pdf>>. Acesso em: 20 set. 2017.

ROCHA, CAUÃ B; FERREIRA, HENRIQUE S; HEROSO, LEANDRO F; ZALESKI, RAFAEL H. **Sistema De Monitoramento De Consumo De Água Doméstico Com A Utilização De Um Hidrômetro Digital.** 2014. Disponível em: <http://www.dainf.ct.utfpr.edu.br/~fabro/IF66J/Relatorios_Finais/2014_1/Sistema%20de%20Monitoramento%20de%20Consumo%20de%20%C3%81gua%20Dom%C3%A9stico%20com%20a%20Utiliza%C3%A7%C3%A3o%20de%20um%20Hidr%C3%B4metro%20Digital.pdf> Acesso em: 01 mai. 2018.

UPTON, EBEN; HALFACREE, GARETH. **Raspberry Pi: Guia do Usuário.** 4. Ed. Rio de Janeiro: Alta Books 2017.

SMARTARDUINO.ESP12E- Manual do Usuário. Disponível em: <<https://smartarduino.gitbooks.io/user-manual-for-esp-12e-devkit/content/>>. Acesso em: 13 out. 2017.