



UNIVERSIDADE FEDERAL DO PARÁ  
CAMPUS CASTANHAL  
FACULDADE DE COMPUTAÇÃO

ANTONIO MISAEL LOPES DE FREITAS

**MONITORAMENTO REMOTO DE CONSUMO ENERGÉTICO RESIDENCIAL:** uso de IoT's para auxiliar no consumo consciente de energia.

CASTANHAL  
2018

ANTONIO MISAEL LOPES DE FREITAS

**MONITORAMENTO REMOTO DE CONSUMO ENERGÉTICO RESIDENCIAL:** uso de IoT's para auxiliar no consumo consciente de energia.

Monografia apresentada ao Curso de Bacharelado em Engenharia de Computação da Universidade Federal do Pará, como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação.

Universidade Federal do Pará  
Faculdade de Computação  
Curso de Bacharelado em Engenharia de Computação

Orientador: Prof. Dr. José Jailton Ferreira Junior

Castanhal  
2018

**ANTONIO MISAEL LOPES DE FREITAS**

**MONITORAMENTO REMOTO DE CONSUMO ENERGÉTICO RESIDENCIAL:** uso de IoT's para auxiliar no consumo consciente de energia.

Monografia apresentada ao Curso de Bacharelado em Engenharia de Computação da Universidade Federal do Pará, como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação.

**Aprovado em:** \_\_/\_\_/\_\_\_\_

**Banca Examinadora**

---

Prof. Dr. José Jailton Ferreira Junior  
Orientador – UFPA - Castanhal

---

Prof. Me Igor Ruiz Gomes  
UFPA – Castanhal

---

Prof. Me Renato Hidaka Torres  
IFPA - Paragominas

## **Agradecimentos**

Acima de tudo, agradeço a Deus por mais esta realização.

A minha família, que sempre me apoiou em minha busca por mais conhecimento, me incentivando em não desistir, diante das dificuldades encontradas durante a trajetória do curso.

Aos colegas de curso de Engenharia de Computação 2014, pelo companheirismo e senso de cooperação em diversas etapas do curso.

Ao meu orientador Professor Dr. José Jailton Ferreira Junior, pelas boas ideias e colaborações no decorrer do curso e no desenvolvimento deste trabalho.

Se clamares por conhecimento, e por  
inteligência alçares a tua voz,  
Se como a prata a buscares e como a tesouros  
escondidos a procurares,  
Então entenderás o temor do Senhor, e acharás  
o conhecimento de Deus.

Provérbios 2:3-5

## RESUMO

É uma exigência cada vez mais latente que a humanidade tenha uma postura mais consciente quanto ao consumo de fontes energéticas, em especial a eletricidade. Este trabalho tem como objetivo desenvolver um sistema que permita ao usuário ter uma visualização, através de um aplicativo móvel, do comportamento de sua rede elétrica (tensão, corrente e potência) e a compreensão de como o uso de determinados equipamentos impactam no consumo elétrico mensal. Para a construção deste projeto foi utilizado a placa de desenvolvimento IoT (Internet das coisas) NodeMCU, o módulo de medição de rede elétrica PZEM-004t e o banco de dados na nuvem FIREBASE, e para a construção do aplicativo móvel, foi usado a IDE Android Studio. Após a análise documental e montagem do sistema foram inicializados os testes, certificando a confiabilidade dos dispositivos, bem como a integridade e capacidade do banco de dados. Através dos testes foi possível concluir o quão importante é o desenvolvimento de recurso que permitam ao usuário a visualização de como seus hábitos de uso interferem no consumo energético residencial. Foi visualizado através da interface do aplicativo que o chuveiro elétrico alterou em mais de 600% o consumo elétrico, e quando ligado no mesmo momento que uma central de refrigeração, a rede atingiu um pico de 39,11 amperes, forçando o desligamento do disjuntor da residência, pois o mesmo era de 30 amperes. Com a visualização destas informações em tempo real, o usuário constrói a percepção da necessidade do uso moderado de determinados equipamentos elétricos, direcionando-o a um consumo mais consciente.

**Palavras-chaves:** rede elétrica, Internet das Coisas, consumo consciente.

## ABSTRACT

It is an increasingly latent demand that humanity has a more conscious attitude about the consumption of energy sources, especially electricity. This work aims to develop a system that allows the user to have a view, through a mobile application, of the behavior of their electrical network (voltage, current and power) and the understanding of how the use of certain equipment impacts monthly electricity consumption. For the construction of this project was used the IoT (Internet of Things) NodeMCU development board, the PZEM-004t network measurement module and the FIREBASE cloud database, and for the construction of the mobile application, the IDE Android Studio. After the documentary analysis and assembly of the system was initialized the tests, certifying the reliability of the devices as well as the integrity and capacity of the database. Through the tests it was possible to conclude how important is the development of resources that allow the user to see how their usage habits interfere in residential energy consumption. It was visualized through the application interface that the electric shower changed by more than 600% the electrical consumption, and when connected at the same time as a central cooling, the network reached a peak of 39.11 amperes, forcing the circuit breaker residence, as it was 30 amperes. By visualizing this information in real time, the user constructs the perception of the need for the moderate use of certain electrical equipment, directing it to a more conscious consumption.

**Keywords:** electric grid, Internet of Things, conscious consumption.

## LISTA DE FIGURAS

|                                                                                                                           |    |
|---------------------------------------------------------------------------------------------------------------------------|----|
| Figura 1 - Estrutura do medidor eletromecânico.....                                                                       | 17 |
| Figura 2 - Medidor eletromecânico. ....                                                                                   | 17 |
| Figura 3 - Medidor eletrônico de energia. ....                                                                            | 18 |
| Figura 4 - Esp8266 NodeMCU.....                                                                                           | 22 |
| Figura 5 - Esquema de disposição das conexões do Kit NodeMCU.....                                                         | 23 |
| Figura 6 - Pzem-004T.....                                                                                                 | 24 |
| Figura 7 - Bobina toroidal.....                                                                                           | 24 |
| Figura 8 - Base NodeMCU.....                                                                                              | 25 |
| Figura 9 - Esquema de funcionamento do divisor de tensão.....                                                             | 33 |
| Figura 10 -Divisor de tensão antes de solda e isolamento .....                                                            | 33 |
| Figura 11 - Divisor de tensão após solda e isolamento .....                                                               | 33 |
| Figura 12 - Esquema de ligação dos componentes.....                                                                       | 34 |
| Figura 13 - Ligação NodeMCU e PZEM-004T .....                                                                             | 35 |
| Figura 14 - Visão da ligação dos componentes junto a rede elétrica.....                                                   | 35 |
| Figura 15 - Visão do sistema montado .....                                                                                | 36 |
| Figura 16 - Bobina toroidal junto a rede elétrica (leitura da corrente). ....                                             | 36 |
| Figura 17 - Adição da Url da placa NodeMCU.....                                                                           | 37 |
| Figura 18 - Configuração do NodeMCU. ....                                                                                 | 37 |
| Figura 19 - Inserção de bibliotecas.....                                                                                  | 38 |
| Figura 20 - Definição dos pinos do NodeMCU.....                                                                           | 39 |
| Figura 21 - Definições no código do NodeMCU.....                                                                          | 40 |
| Figura 22 – Definição de Variáveis globais no código do NodeMCU.....                                                      | 40 |
| Figura 23 - Código do Void Setup do NodeMCU.....                                                                          | 41 |
| Figura 24 -Verificação de conexão com o roteador wireless .....                                                           | 41 |
| Figura 25 - Definição do ArduinoJson.....                                                                                 | 42 |
| Figura 26 - Código do laço for para leitura de 30 segundos do PZEM-004T....                                               | 42 |
| Figura 27 - Bloco de construção e envio de mensagem ao banco de dados. ..                                                 | 43 |
| Figura 28 - Criando um banco de dados para o projeto .....                                                                | 44 |
| Figura 29 - Regras de leitura e escrita do Firebase.....                                                                  | 45 |
| Figura 30 - Estrutura do banco de dados. ....                                                                             | 45 |
| Figura 31 - Tela inicial.....                                                                                             | 47 |
| Figura 32 - Tela Home do Aplicativo.....                                                                                  | 47 |
| Figura 33 - Tela relatório de consumo.....                                                                                | 47 |
| Figura 34 - Tela relatório por período.....                                                                               | 47 |
| Figura 35 - Classe “SomaLeituras”. ....                                                                                   | 48 |
| Figura 36 - Inicialização do hardware.....                                                                                | 49 |
| Figura 37 - Conectado a rede wireless.....                                                                                | 49 |
| Figura 38 - Leitura com o aplicativo.....                                                                                 | 50 |
| Figura 39 - Leitura de corrente com amperímetro. ....                                                                     | 50 |
| Figura 40 - Leitura da tensão usando um multímetro digital.....                                                           | 51 |
| Figura 41 - Leitura da tensão pelo PZEM-004T.....                                                                         | 51 |
| Figura 42 - Total de leitura do banco de dados. ....                                                                      | 52 |
| Figura 43 - Tela do aplicativo com o chuveiro elétrico desligado(A) e ligado(B).<br>.....                                 | 55 |
| Figura 44 - Comportamento da rede elétrica com chuveiro elétrico e central de ar condicionado (desligados e ligados)..... | 56 |

## LISTA DE TABELAS

|                                                                                         |    |
|-----------------------------------------------------------------------------------------|----|
| Tabela 1 - Funcionalidades do Firebase.....                                             | 27 |
| Tabela 2 - Regras de Escrita do Firebase .....                                          | 28 |
| Tabela 3 - Equivalência de pinos do NodeMCU e Arduino.....                              | 39 |
| Tabela 4 - Tabela de consumo .....                                                      | 53 |
| Tabela 5 - Relação de consumo do chuveiro elétrico e Central de ar<br>condicionado..... | 54 |

## LISTA DE GRÁFICOS

|                                                 |    |
|-------------------------------------------------|----|
| Gráfico 1- Matriz Energética Brasileira .....   | 15 |
| Gráfico 2 - Consumo Energético residencial..... | 54 |

## LISTA DE EQUAÇÕES

|                                               |    |
|-----------------------------------------------|----|
| Equação 1 - Divisor de tensão.....            | 32 |
| Equação 2 - Calculo do divisor de tensão..... | 33 |

## LISTA DE ACRÔNIMOS

|              |                                                                             |
|--------------|-----------------------------------------------------------------------------|
| <b>ANEEL</b> | Agencia Nacional de Energia Elétrica.                                       |
| <b>API</b>   | Application Programming Interface (Ambiente de Programação de Aplicativos). |
| <b>IDE</b>   | Integrated Development Environment (Ambiente Integrado de Desenvolvimento). |
| <b>IoT</b>   | Internet of Things (Internet das coisas).                                   |
| <b>IP</b>    | Internet Protocol (Protocolo de Internet).                                  |
| <b>JSON</b>  | JavaScript Object Notation (Notação de Objetos JavaScript).                 |
| <b>LED</b>   | Light Emitting Diode (Diodo Emissor de Luz).                                |
| <b>NTP</b>   | Network Time Protocol (Protocolo de Tempo em Rede).                         |
| <b>URL</b>   | Uniform Resource Locator (Localizador padrão de recursos).                  |
| <b>TX</b>    | Transmissão de Dados                                                        |
| <b>RX</b>    | Recepção de Dados                                                           |
| <b>SQL</b>   | <i>Structured Query Language</i> (Linguagem de Consulta Estruturada)        |
| <b>USB</b>   | Universal Serial Bus (Porta serial universal)                               |
| <b>UTF</b>   | Unicode Transformation Format (formato de transformação Unicode)            |

# SUMÁRIO

|                                                                                |           |
|--------------------------------------------------------------------------------|-----------|
| <b>LISTA DE FIGURAS .....</b>                                                  | <b>8</b>  |
| <b>LISTA DE TABELAS .....</b>                                                  | <b>9</b>  |
| <b>LISTA DE GRÁFICOS .....</b>                                                 | <b>9</b>  |
| <b>LISTA DE EQUAÇÕES .....</b>                                                 | <b>9</b>  |
| <b>LISTA DE ACRÔNIMOS .....</b>                                                | <b>10</b> |
| <b>I - INTRODUÇÃO .....</b>                                                    | <b>12</b> |
| <b>1 – CONTEXTUALIZAÇÃO .....</b>                                              | <b>12</b> |
| <b>1.2– MOTIVAÇÃO .....</b>                                                    | <b>13</b> |
| <b>1.3 – OBJETIVO GERAL.....</b>                                               | <b>14</b> |
| <b>1.4 – OBJETIVOS ESPECÍFICOS .....</b>                                       | <b>14</b> |
| <b>II - O SISTEMA ELÉTRICO BRASILEIRO .....</b>                                | <b>15</b> |
| <b>2.1 – ESTRUTURA E FUNCIONAMENTO DA REDE ELÉTRICA BRASILEIRA. ....</b>       | <b>16</b> |
| <b>2.2 – O REGISTRO DO CONSUMO DE ENERGIA ELÉTRICA. ....</b>                   | <b>16</b> |
| <b>2.3 – CRISE ENERGÉTICA BRASILEIRA DE 2001.....</b>                          | <b>20</b> |
| <b>III - METODOLOGIA .....</b>                                                 | <b>21</b> |
| <b>3.1 - HARDWARE UTILIZADO .....</b>                                          | <b>21</b> |
| <b>3.1.1 – NODEMCU ESP-12E.....</b>                                            | <b>22</b> |
| <b>3.1.2 – MODULO PZEM-004T.....</b>                                           | <b>24</b> |
| <b>3.1.3 – SHILD BASE DE ALIMENTAÇÃO PARA NODEMCU .....</b>                    | <b>25</b> |
| <b>3.2 – BANCO DE DADOS FIREBASE. ....</b>                                     | <b>25</b> |
| <b>3.2.1 – CARACTERÍSTICAS GERAIS DO FIREBASE.....</b>                         | <b>26</b> |
| <b>3.2.2 – CAPACIDADE DE ARMAZENAMENTO E ACESSO .....</b>                      | <b>27</b> |
| <b>3.2.3 – REGRAS DE ESCRITA E USO .....</b>                                   | <b>28</b> |
| <b>3.2.4 – SEGURANÇA NO ACESSO AO BANCO .....</b>                              | <b>29</b> |
| <b>3.3 – ANDROID STUDIO .....</b>                                              | <b>30</b> |
| <b>IV - DESENVOLVIMENTO DO PROJETO (PROTOTIPO) .....</b>                       | <b>31</b> |
| <b>4.1 – MONTAGEM E CONFIGURAÇÃO DOS DISPOSITIVOS.....</b>                     | <b>32</b> |
| <b>4.2 – CODIGO FONTE DO NODEMCU. ....</b>                                     | <b>36</b> |
| <b>4.2.1 – ADICIONANDO BIBLIOTECAS .....</b>                                   | <b>37</b> |
| <b>4.2.2 – DIFINIÇÕES GERAIS .....</b>                                         | <b>39</b> |
| <b>4.2.3 – CONFIGURAÇÃO DE INICIALIZAÇÃO DO NODEMCU (“void setup()”) .....</b> | <b>40</b> |
| <b>4.2.4 – VOID LOOP.....</b>                                                  | <b>41</b> |

|                                                         |           |
|---------------------------------------------------------|-----------|
| <b>4.3 – BANCO DE DADOS DATABASE FIREBASE. ....</b>     | <b>44</b> |
| <b>4.4 – DESENVOLVIMENTO DO APLICATIVO ANDROID.....</b> | <b>46</b> |
| <b>V - RESULTADOS.....</b>                              | <b>49</b> |
| <b>VI - CONSIDERAÇÕES FINAIS.....</b>                   | <b>57</b> |
| <b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>                 | <b>60</b> |
| <b>ANEXOS.....</b>                                      | <b>62</b> |

## **I - INTRODUÇÃO**

### **1 – CONTEXTUALIZAÇÃO**

A descoberta da eletricidade no século VI, por Talles de Mileto, trouxe grandes benefícios a humanidade, permitindo o desenvolvimento de máquinas elétricas, dispositivos de comunicação e acelerando a produção de bens e serviços nos mais diversos setores. A partir da segunda etapa da revolução industrial (1860 – 1900), a quantidade de equipamentos eletrônicos cresceu substancialmente, onde foi observado a troca gradativa de equipamentos manuais por mecânicos elétricos, em virtude disso a produção de energia elétrica, bem como o seu consumo tem sido de fundamental importância para o desenvolvimento de diversas nações.

A energia, nas suas mais diversas formas, é indispensável à sobrevivência da espécie humana. E mais do que sobreviver, o homem procurou sempre evoluir, descobrindo fontes e formas alternativas de adaptação ao ambiente em que vive e de atendimento às suas necessidades. Dessa forma, a exaustão, escassez ou inconveniência de um dado recurso tendem a ser compensadas pelo surgimento de outro(s). Em termos de suprimento energético, a eletricidade se tornou uma das formas mais versáteis e convenientes de energia, passando a ser recurso indispensável e estratégico para o desenvolvimento socioeconômico de muitos países e regiões.

ANEEL, 2002, p.19.

Como citado no documento da ANEEL, pensar em um mundo sem eletricidade é algo desnorteador, tendo em vista que cada vez mais dependemos de equipamentos, cuja fonte energética é a eletricidade. Ações que hoje fazemos de forma natural e desatenta de seu funcionamento, como assistir televisão, usar um micro-ondas, ter um ambiente refrigerado, são possíveis devido o uso de eletricidade, no entanto, a falta de visibilidade de como/quanto consumimos energia elétrica cotidianamente não nos permite um controle mais consciente sobre gastos energéticos.

Cada dia mais tem-se desenvolvido uma consciência a respeito do consumo de recursos energéticos, seja ela por motivos ambientais e/ou econômicos. No entanto, observa-se a falta de mecanismos que permita ao usuário a visualização do seu consumo real e instantâneo, pois é disponibilizado ao usuário, na conta de energia, somente o consumo total mensal, não permitindo ao consumidor a visualização do seu consumo real e os períodos em que tem maior consumo, bem como a fuga de energia.

Dante tal circunstancia, faz-se necessário a criação de mecanismos que permitam ao usuário a visualização de como se dá o consumo de energia elétrica residencial, bem como a observação do comportamento de sua rede elétrica (tensão, corrente e potencia), dando condições aos usuários de visualizar como determinados equipamentos interferem no consumo elétrico mensal e consequentemente no valor da conta de energia.

## **1.2– MOTIVAÇÃO**

Segundo o código de defesa do consumidor, artigo 6º, III, da Lei 8078/90, o fornecedor de serviços deve disponibilizar todas as informações necessárias, para que o usuário possa decidir pelo uso ou não de determinado serviço ou produto. No trato ao consumo de energia elétrica, a prestadora de serviços elétricos do estado do Pará (CELPA), disponibiliza um talão contendo informações relativas ao consumo total mensal atual e dos 12 meses anteriores, os tributos e a tensão nominal contratada, no entanto, o usuário não consegue visualizar em que período do dia ou do mês ele teve um maior consumo, ou que equipamentos interferem mais no consumo, cabendo ao usuário fazer anotações do consumo diário, caso queira ter um relatório mais detalhado de consumo.

Atualmente existem equipamentos que podem ser instalados na rede elétrica para fazer análise e registro das informações estatísticas de consumo, no entanto, são mais indicados a indústria, tendo em vista o alto valor agregado, e o uso de informações muito técnicas, tornando mais dificultosa a análise pelo consumidor residencial. Em contrapartida, nos últimos anos, o termo “internet das coisas” (IoT – Internet of Things) tem ganhado cada vez mais destaque, onde o mesmo refere-se a conexão de diversos equipamentos de uso comum, a internet, através de sensores inteligentes e softwares que transmitem dados para uma rede. A criação de dispositivos usando conhecimento de IoT’s se tornou uma opção para a solução de problemas analíticos, usando sensores, que são facilmente implementados na tecnologia, e apresentam um custo financeiro adequado a pequenos projetos.

Uma das vantagens do uso de IoT’s é a facilidade de uso e geralmente os dispositivos (sensores, módulos embarcados, interfaces, entre outros) são baratos e encontrados facilmente em loja online.

### 1.3 – OBJETIVO GERAL

A partir das questões abordadas anteriormente, este trabalho visa desenvolver um sistema, baseado em conceitos referentes a IoT's, que permita ao consumidor residencial a visualização do comportamento da rede elétrica (tensão, corrente e potência), garantindo assim, transparência e consciência quanto ao uso de energia elétrica.

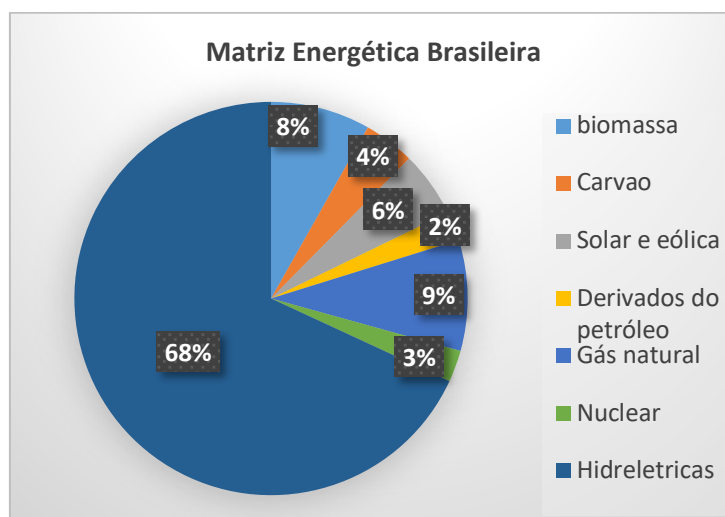
### 1.4 – OBJETIVOS ESPECÍFICOS

- Identificar quais equipamentos (*hardwares*) são mais viáveis ao desenvolvimento do projeto, tendo como base o custo financeiro, confiabilidade, acessibilidade e precisão na leitura;
- Definir o banco de dados apropriado, visando o uso gratuito, capacidade de armazenamento e persistência dos dados por grande período;
- Desenvolver um aplicativo para dispositivos móveis que permita ao usuário o acesso as informações da rede elétrica residencial remotamente, através da internet;
- Garantir que somente o usuário registrado tenha acesso as informações de sua rede elétrica, dando segurança ao sistema;
- Analisar como alguns eletrodomésticos interferem no consumo energético residência.

## II - O SISTEMA ELÉTRICO BRASILEIRO

Atualmente a principal fonte de obtenção de energia elétrica no mundo, dá-se através do uso de combustíveis fósseis (carvão, óleo e gás natural) em termelétricas, sendo responsável por 66,3% da energia gerada no mundo, seguida das hidrelétricas (16%) e nuclear (10,6%). O Brasil, por possuir uma grande quantidade de rios de porte elevado, sua matriz energética<sup>1</sup> diverge da maior parte do mundo, obtendo 68,1% da energia consumida por meio de hidrelétricas, e o restante é obtida de termelétricas, solar, eólica e nuclear, como demonstrado no gráfico 1.

Gráfico 1- Matriz Energética Brasileira



Fonte: Elaborada pelo autor.

O uso de energia elétrica no Brasil data de 1879, quando Dom Pedro II concedeu a Thomaz Alva Edison a permissão de instalação de dispositivos destinados a iluminação pública, sendo instalada e inaugurada a primeira rede elétrica permanente na Estação Central da Estrada de Ferro Dom Pedro II, atual Central do Brasil, no Rio de Janeiro. Em 1881 foi instalada a primeira iluminação externa pública no trecho da atual Praça da República na cidade do Rio de Janeiro, e 1883 entrou em operação a primeira usina hidrelétrica no Brasil, localizada no Ribeirão do Inferno, afluente do rio Jequitinhonha, na cidade de Diamantina, no estado de Minas Gerais.

O Brasil é o país da América Latina, pioneiro na produção e distribuição de energia elétrica, possuindo hoje uma malha elétrica que atinge 107,4 mil quilômetros

<sup>1</sup> Representa toda energia disponibilizada para ser distribuída, transformada e consumida nos processos produtivos, uma representação quantitativa dos recursos energéticos disponíveis em um país ou região.

de extensão, sendo um sistema formado por linhas de alta tensão (230mil volts ou mais) e demais instalações que operam entre 69 mil volts e 138 mil volts).

## **2.1 – ESTRUTURA E FUNCIONAMENTO DA REDE ELÉTRICA BRASILEIRA.**

Segundo dados da Agencia Nacional de Energia Elétrica (ANEEL), hoje o Brasil tem a capacidade de produzir aproximadamente 160 GigaWatts de energia, através de 1323 usinas hidrelétricas e 2990 termelétricas.

O sistema brasileiro de energia elétrica é formado pelas seguintes estruturas:

- Geradoras de energia: Hidrelétricas, termelétricas e eólica;
- Linhas de transmissão: alta tensão;
- Subestações: reduz a tensão das linhas de transmissão, adequando a necessidade dos usuários;
- Linha de transmissão de baixa tensão: direciona a energia da subestação ao consumidor final.

Existem dois subsistemas de transmissão de energia nacional, que até 1999 eram independentes, o sistema Sul-Sudeste-Centro-Oeste e o Norte-Nordeste, e que foram interligados, permitindo que uma rede supra a necessidade da outra rede, na ocasião de baixa produção em decorrência alterações climáticas, tendo em vista que, apesar de possuímos muitas termelétricas, elas produzem apenas 29,5% do que é consumido no país, e 68% provem de hidrelétricas, assim a escassez de chuvas em determinada região pode afetar drasticamente a produção energia regional.

## **2.2 – O REGISTRO DO CONSUMO DE ENERGIA ELÉTRICA.**

O consumo energético residencial/empresarial é feito através de registradores de consumo, popularmente conhecido como relógio de energia, o qual é instalado segundo um padrão das prestadoras de serviço. Atualmente existem dois tipos de medidores de consumo de energia, o eletrônico e o eletromecânico.

O medidor de energia mais antigo e ainda utilizado é o eletromecânico (figura 1), que possui um disco de metal (figura 2), que gira quando a eletricidade passa pelas bobinas, gerando um campo magnético, sendo que a velocidade de giro é diretamente proporcional ao campo magnético gerado, e a cada giro é determinado uma

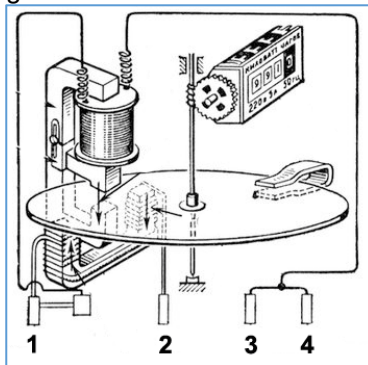
quantidade de watts, que é registrado e mostrado através de um ciclômetro, conjunto de relógios, ou mostrador digital.

Figura 1 - Estrutura do medidor eletromecânico.



Fonte: Site Saber Eletrônica.<sup>2</sup>

Figura 2 - Medidor eletromecânico.



Fonte: Blog Nova Eletrônica.<sup>3</sup>

O outro modelo de registrador de energia é o eletrônico (figura 3), que utiliza um circuito elétrico para determinar a quantidade de energia consumida, e permite o monitoramento remoto. Esse modelo é mais moderno possuindo conexões que permitem o acoplamento de dispositivos de envio de informações via radio frequência, fibra ótica ou cabo rj45, diretamente para a estação de monitoramento das prestadoras, no entanto, na maioria das casas o registro no banco de dados das prestadoras de serviço é feito manualmente, onde todo mês um funcionário passa para coletar os dados de consumo.

<sup>2</sup> Disponível em: <<https://www.sabereletrica.com.br/medidor-de-energia-eletrica/>>. Acesso em out. 2018.

<sup>3</sup> Disponível em: <<http://blog.novaeletronica.com.br/medidor-de-energia-eletrica/>>. Acesso em out. 2018

Figura 3 - Medidor eletrônico de energia.



Fonte: Site Minulight.<sup>4</sup>

O consumo de energia é baseado em quilowatts, onde faz-se a subtração da leitura do relógio atual pela leitura feita em um mês anterior, assim gerando o valor consumido e o valor a pagar. Cabe salientar que o valor do quilowatt-hora é formado a partir da análise de 3 fatores, definido pela Agencia Nacional de Energia Elétrica - ANEEL, os quais são:

- Custo para geração da energia;
- Transporte de energia;
- Encargos setoriais.

Os encargos setoriais referem-se a contribuições instituídas por lei, cujo repasse ao consumidor, tem como objetivo garantir o equilíbrio econômico financeiro, sendo que o valor de tais encargos é incorporado diretamente no valor dos quilowatts, não aparecendo de forma clara na conta de energia. Os encargos setoriais referentes a produção e transmissão de energia são:

- Conta de Desenvolvimento Energético – CDE;
- Programa de Incentivo à Fontes Alternativas de Energia Elétrica – PROINFA;
- Compensação Financeira pela Utilização de Recursos Hídricos – CFURH;
- Encargos de Serviços do Sistema – ESS e de Energia de Reserva – EER;
- Taxa de Fiscalização dos Serviços de Energia Elétrica – TFSEE;

<sup>4</sup> Disponível em: <[https://www.minulight.com.br/wp-content/uploads/2013/10/medidor\\_eletronico\\_cronos\\_6001\\_60211.png](https://www.minulight.com.br/wp-content/uploads/2013/10/medidor_eletronico_cronos_6001_60211.png)>. Acesso em out. 2018.

- Pesquisa e Desenvolvimento – P&D e Programa de Eficiência Energética – PEE; e
- Contribuição ao Operador Nacional do Sistema – ONS

A partir de 2015 a forma para determinar o valor de pagamento do consumo de energia elétrica modificou, e foi incorporado o Sistema de Bandeira Tarifárias, que estão ligadas diretamente aos custos de produção de energia elétrica. As bandeiras adotadas foram verde, amarela e vermelha, as quais indicam se haverá aumento ou não da tarifa. Cada bandeira possui as seguintes características:

- Verde: condições favoráveis a geração de energia, e a tarifa não sofre nenhum acréscimo;
- Amarela: condições menos favoráveis de geração de energia, e é acrescido R\$ 0,01 para cada quilowatt-hora(kWh) consumido;
- Vermelha (Patamar 1): custo elevado para produção de energia e ocorre acréscimo de R\$ 0,03 para cada quilowatt-hora;
- Vermelha (Patamar 2): custo bastante elevado para geração de energia, sendo acrescido R\$ 0,05 para cada quilowatt-hora consumido.

Cabe salientar que a bandeira vermelha está intimamente ligada ao uso de termelétricas, pois em caso de baixa na produção de energia pelas hidrelétricas, algumas termelétricas que estão inativas, ou que estão funcionando em capacidade mínima, passam a funcionar, ou aumentam sua capacidade de produção, porém o custo de geração de energia nelas, em comparação com as hidrelétricas, é elevadíssimo, e este custo é repassado ao consumidor.

O valor final da energia elétrica é composto por 53,5% referente a compra de energia, transmissão e encargos setoriais, 17% do valor é pagamento para distribuição e 29,5% trata de encargos tributários como ICMS, PIS/COFINS, e contribuição para iluminação pública.

## **2.3 – CRISE ENERGÉTICA BRASILEIRA DE 2001.**

No ano de 2001 o Brasil passou por uma grande crise energética, com risco de ocorrer apagões, em virtude da capacidade de produção de energia não ser capaz de suprir a necessidade de consumo.

Neste período, 90% da produção de energia elétrica no Brasil era feita através de hidrelétricas, e o país enfrentava um período de estiagem nas principais represas, diminuindo o nível de água dos reservatórios, limitando a capacidade de produção das hidrelétricas. Outro agravante para a crise refere-se a questões governamentais, na qual limitava o investimento do setor energético. Em contrapartida, ocorria o aumento contínuo do consumo energético populacional e o crescimento da indústria brasileira, desta forma a relação produção x consumo encontra-se praticamente equiparada.

Como medidas para evitar um “apagão”, o governo brasileiro tomou medidas emergenciais, no intuito de diminuir o consumo energético, dentre elas destacam-se:

- Aumento de produção de energia por termelétricas;
- Imposição de limites de consumo para população, onde foi definido certa quantidade de quilowatts para cada consumidor, e cobrança de multas para quem não economizasse energia;
- Incentivo para troca de lâmpadas incandescentes por fluorescentes;
- Indústrias trocaram máquinas elétricas por funcionamento a gás;
- Leilões de energia futura no Mercado Atacadista de Energia (MAE), no intuito de forçar investimentos no setor;

Para aumentar a capacidade energética brasileira deu-se início à construção de novas hidrelétricas, entre elas, Santo Antônio e Jirau no estado de Rondônia e Belo Monte no estado do Pará, investimento em outras formas de produção de energia, como eólica e solar, e a melhoria nas redes de transmissão.

Foi implantado políticas de incentivo ao consumo consciente de energia elétrica, através de propagandas nas mídias nacionais, e programas de trocas de itens residências que consomem bastante energia, como lâmpadas incandescentes e refrigeradores antigos.

Atualmente, segundo a ANEL, é descartada a possibilidade de um novo apagão, tendo em vista que a produção de energia elétrica brasileira é suficiente para suprir a necessidade energética nacional.

### III - METODOLOGIA

A partir da problemática, foi buscado maneiras de desenvolver um sistema (*hardware*, *software* e banco de dados) usando conceitos relacionados a internet das coisas(*IoT*), tendo em foco o uso de *hardwares* de baixo custo e com fácil implementação e o uso de um banco de dados na nuvem e gratuito, porém que tivesse uma capacidade de armazenamento, manutenção e consistência dos dados por grande período e que não viesse comprometer a confiabilidade do sistema.

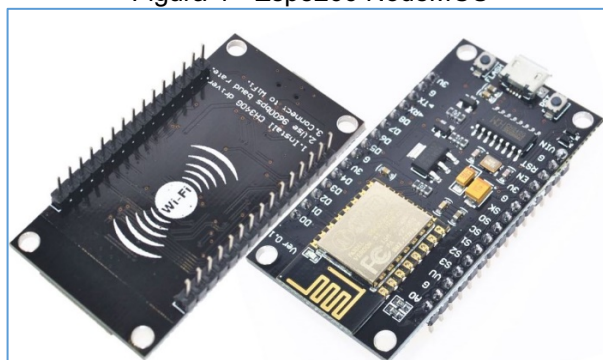
#### 3.1 - HARDWARE UTILIZADO

No início do desenvolvimento do projeto, foi optado em usar a placa Arduino Uno, o módulo wireless ESP8266-01 e o módulo de leitura de tensão e corrente PZEM004T. A escolha inicial do Arduino Uno se deu em vista que já possuía uma peça disponível e ter sido usada como recurso na disciplina Projeto de Engenharia I, sendo assim, já tinha familiaridade e conhecimento para a manipulação da mesma, a qual é uma interface desenvolvida com o uso do micro controlador ATmega328. Da mesma forma, a escolha do ESP8266-01, se deu em virtude de já ter utilizado em um projeto de monitoramento de luminosidade e que foi apresentado na V Jornada de Inclusão Digital, na Universidade Federal do Pará no campus da cidade de Castanhal.

Depois de alguns testes, foi possível determinar a perfeita usabilidade dos dispositivos, no entanto, o código precisava adaptar a placa do Arduino, pois, tanto o módulo wireless quanto o PZEM-004T usam a comunicação serial, desta forma, seria necessário fazer a alternância das portas seriais, a cada leitura de dados da rede elétrica e envio de dados para o banco na nuvem. A partir da análise de requisitos do sistema foi possível determinar que os *hardwares* ideais ao projeto eram o módulo NodeMCU ESP-12E Ch340 (ESP8266-12E) com uma base de alimentação e o módulo PZEM-004T.

### 3.1.1 – NODEMCU ESP-12E.

Figura 4 - Esp8266 NodeMCU



Fonte: Site Picclick.<sup>5</sup>

O NodeMCU é um kit de desenvolvimento e *firmware* de código aberto, baseado no chip Esp8266 que possui uma interface usb-serial e um regulador de tensão de 3.3V. Este dispositivo apresenta grande vantagem no uso como interface para o desenvolvimento de projetos IoT, quando comparado com o Arduino, tendo em vista que o mesmo já possui integrado a capacidade de conexão em redes wireless, podendo ser utilizado como ponto de acesso ou estação, além de um poder computacional superior ao arduino, tendo em vista que o Arduino possui uma CPU AVR de 8bits (RISC) com um *clock* de 16MHz o ESP8266-12E é incorporado com o micro controlador Tensilica L106 de 32bits (RISC) de baixo consumo e um *clock* de 80MHz podendo atingir um valor máximo de 160MHz.

O chip esp8266 foi projetado pela empresa Espressif Systems, e é uma solução de rede *wireless* autossuficiente, oferecendo uma ponte de micro controlador existente para *wifi* e também capaz de executar aplicações independentes, desta forma torna desnecessário o uso de outro microcontrolador/placa (como o arduino), para a execução de projetos IoT.

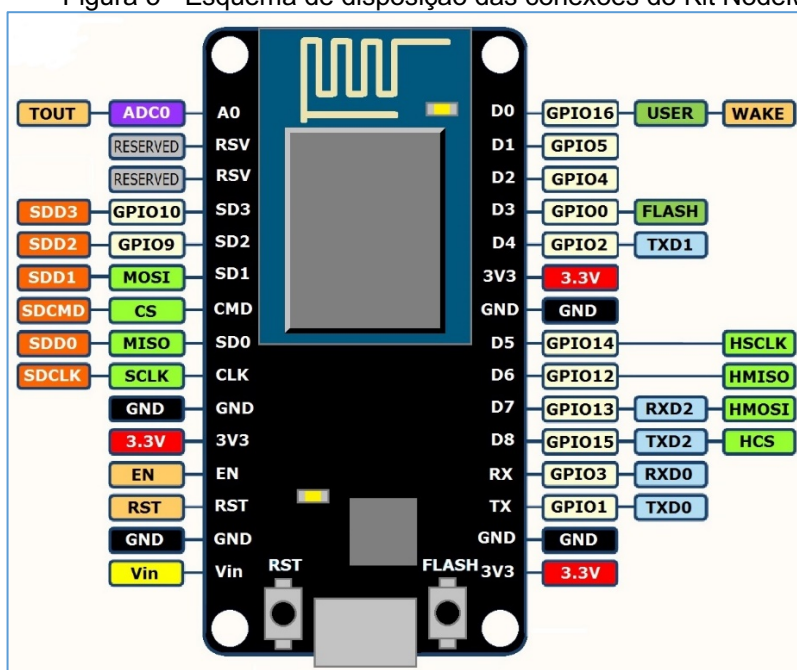
As especificações técnicas do NodeMCU são as seguintes:

- Tensão de trabalho: 3.3v;
- Corrente: 10uA ~ 170mA;
- Memória *flash*: 4MByte
- Conexão usb-ttl integrada com chip CH340G;

<sup>5</sup> Disponível em: <<https://picclick.com/NodeMcu-Lua-ESP8266-ESP-12E-CH340-WIFI-V3-Internet-162346546916.html>>. Acesso em out. 2018.

- Modo de operação: *Wifi Direct* (P2P), ponto de acesso ou ambos;
- Wifi 2,4GHz, com suporte ao modo de segurança WPA/WPA2;
- Suporte a múltiplas conexões;
- D0 ~ D8, SD1 ~ SD3: usado como GPIO, PWM, IIC;
- Taxa de transferência: 10-460800bps;
- Suporte a interface de comunicação de dados UART/GPIO;
- Atualização remota de *firmware* (OTA);
- Suporte a redes inteligentes *Smart link*;
- Temperatura de trabalho: -40 °C ~ +125 °C;

Figura 5 - Esquema de disposição das conexões do Kit NodeMCU



Fonte: Site Filipflop.<sup>6</sup>

Como pode ser observado na figura 4, o kit NodeMCU possui pinos na parte inferior, referentes as portas mostradas na figura 5, que permitem a conexão direta com uma *protoboard*, facilitando a implementação e a inserção de outros componentes.

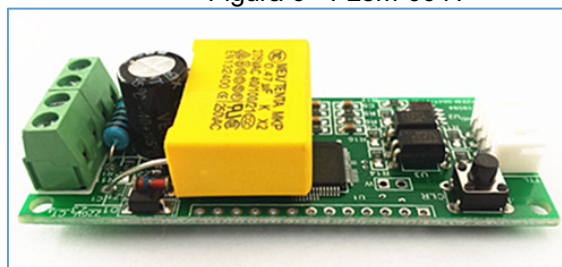
<sup>6</sup> Disponível em: <<https://www.filipeflop.com/blog/esp8266-nodemcu-como-programar/>> Acesso em nov. 2018.

### 3.1.2 – MODULO PZEM-004T

Para fazer a leitura da rede elétrica e obter os valores de tensão, corrente e potência, foi escolhido o modulo PZEM-004T, por ele ser capaz de fazer todas as leituras de uma só vez, desta forma dando mais exatidão e confiabilidade ao sistema, tendo em vista que a alternativa seria o uso do modulo sensor de corrente SCT-013-000, porém, ele lê somente a corrente e a tensão da rede teria que ser adicionado no código, baseado na tensão padrão indicado pela concessionaria de energia, no entanto, ao fazer algumas leituras na rede, foi possível verificar um variação constante na tensão, desta forma, o valor real do consumo energético ficaria comprometido, caso usa-se um valor fixo de tensão, causando um grande desnível entre o que a concessionaria de energia indicasse como consumo e o registrado pelo dispositivo.

O modulo PZEM-004T é capaz de ler a tensão, corrente, potência ativa e energia, e para isso ele utiliza uma bobina toroidal (figura 8) para ler a corrente, e outros dois fios que devem ser ligados a rede elétrica, para medir a tensão e a partir destes dados, o modulo define também a potencia ativa. Ele possui capacidade de ler tensões entre 80 e 260 VAC, e correntes que variam entre 0 a 100 amperes.

Figura 6 - Pzem-004T



Fonte: Site Mercado Libre.<sup>7</sup>

111

Figura 7 - Bobina toroidal



Fonte: Site Mercado libre.<sup>8</sup>

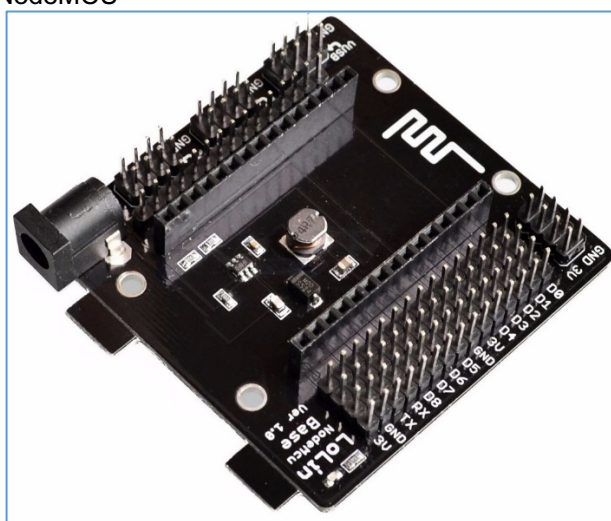
<sup>7</sup> Disponível em: <[https://articulo.mercadolibre.com.mx/MLM-609277210-medidor-de-energia-ttl-pzem-004t-\\_JM](https://articulo.mercadolibre.com.mx/MLM-609277210-medidor-de-energia-ttl-pzem-004t-_JM)>. Acesso em nov. 2018.

<sup>8</sup> Disponível em< [https://articulo.mercadolibre.com.mx/MLM-609277210-medidor-de-energia-ttl-pzem-004t-\\_JM](https://articulo.mercadolibre.com.mx/MLM-609277210-medidor-de-energia-ttl-pzem-004t-_JM)>. Acesso em nov. 2018.

### 3.1.3 – SHIELD BASE DE ALIMENTAÇÃO PARA NODEMCU

Para utilizar os dispositivos do projeto seria necessário o uso de duas fontes de alimentação diferentes, tendo em vista que o NodeMCU opera com 3.3V enquanto que o PZEM-004T usa 5v, desta forma foi preferível utilizar a base de alimentação para o NodeMCU (figura 5), pois a mesma pode usar uma fonte de alimentação entre 7 e 12v, e possui saídas dedicadas de 5v e 3.3v, ideal para alimentar o PZEM-004T e o NodeMCU.

Figura 8 - Base NodeMCU



Fonte: Site Mercado Livre.<sup>9</sup>

## 3.2 – BANCO DE DADOS FIREBASE.

Para o perfeito funcionamento do projeto fazia-se necessário a implementação de um banco de dados na nuvem e gratuito capaz de armazenar certa quantidade de dados, tendo em vista que o dispositivo ficaria conectado a ele de forma permanente, enviando as leituras do sensor de energia a cada 30 segundos. Após a análise das opções disponíveis o banco de dados mais adequado ao projeto foi o *Real Time Database Firebase*, da empresa Google.

<sup>9</sup> Disponível em: <[https://produto.mercadolivre.com.br/MLB-832374664-shield-base-alimentaco-para-esp8266-nodemcu-v3-\\_JM?quantity=1](https://produto.mercadolivre.com.br/MLB-832374664-shield-base-alimentaco-para-esp8266-nodemcu-v3-_JM?quantity=1)> Acesso em nov. 2018.

### 3.2.1 – CARACTERÍSTICAS GERAIS DO FIREBASE.

O *Realtime Database* Firebase é uma base de dados na nuvem em que os dados são guardados utilizando *Javascript Object Notation* (JSON), sendo um banco de dados NoSQL, podendo ser sincronizados em tempo real com todos os dispositivos conectados, sejam eles IOS, Android, JavaScript e outros.

Por ser um banco de dados NoSQL, o FIREBASE possui otimizações e funcionalidade diferentes de um banco de dados relacional, onde o mesmo foi desenvolvido para autorizar operações que possam ser executadas com rapidez, e permita uma experiência de tempo real, atendendo milhões de usuários, não comprometendo a capacidade de resposta instantânea, semelhante a aplicativos já reconhecidos mundialmente como Whatsapp e Facebook.

Como o Firebase utiliza JSON, a estruturação do banco de dados dá-se através de armazenamento e transmissão de informações no formato texto, onde sempre se faz necessário a caracterização do par de informações, sendo que para cada valor representado é necessário atribuir um nome (ou rotulo), que descreve seu significado. As informações podem ser atribuídas a três tipos: *booleano*, *string* e numérico (inteiro ou real), mas a partir dos tipos básicos é possível desenvolver *arrays* e objetos.

Com o FIREBASE é possível criar aplicativos avançados e colaborativos, dando acesso seguro ao banco de dados, implementando regras de segurança compatíveis com diversos tipos de usuários.

Os principais recursos disponíveis no Real Time Database Firebase, podem ser resumidos na Tabela 1, a seguir.

Tabela 1 - Funcionalidades do Firebase.

|                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Em tempo real                          | Em vez de solicitações HTTP típicas, o <i>Firebase Realtime Database</i> usa a sincronização de dados. Sempre que os dados são alterados, todos os dispositivos conectados recebem essa atualização em milissegundos. Crie experiências colaborativas e imersivas sem se preocupar com códigos de rede.                                                                                                                                                                                                                                                                                         |
| Off-line                               | Os apps do Firebase permanecem responsivos mesmo <i>off-line</i> , pois o SDK do <i>Firebase Realtime Database</i> mantém seus dados em disco. Quando a conectividade é restabelecida, o dispositivo cliente recebe as alterações perdidas e faz a sincronização com o estado atual do servidor.                                                                                                                                                                                                                                                                                                |
| Acessível em dispositivos clientes     | O <i>Firebase Realtime Database</i> pode ser acessado diretamente de um dispositivo móvel ou navegador da <i>Web</i> , sem um servidor de aplicativos. A segurança e a validação de dados estão disponíveis por meio de regras de segurança baseadas em expressão do <i>Firebase Realtime Database</i> , executadas quando os dados são lidos ou gravados.                                                                                                                                                                                                                                      |
| Escalonar entre vários bancos de dados | Com o <i>Firebase Realtime Database</i> no plano de preços do <i>Blaze</i> , você pode oferecer suporte em grande escala às necessidades de dados do seu app. Para isso, você pode dividir seus dados entre várias instâncias de banco de dados no mesmo projeto do <i>Firebase</i> . Simplifique a autenticação no seu projeto com o <i>Firebase Authentication</i> e autentique usuários nas suas instâncias de banco de dados. Controle o acesso às informações em cada banco de dados com regras personalizadas do <i>Firebase Realtime Database</i> para cada instância de banco de dados. |

Fonte: Site Firebase.<sup>10</sup>

### 3.2.2 – CAPACIDADE DE ARMAZENAMENTO E ACESSO

O FIREBASE garante a possibilidade de uso em grande e baixa escala, desta forma, é disponibilizado uma conta gratuita para teste e uma conta paga caso seja necessário um grande fluxo de dados, com cobrança baseada no uso. Como o projeto destina-se a desenvolvimento a baixo custo, foi adotado a conta gratuita, porém a capacidade de armazenamento e acesso foi extremamente satisfatório.

Ao criar a conta gratuita, a Google disponibiliza a seguinte capacidade de uso:

<sup>10</sup> Disponível em: <<https://firebase.google.com/docs/database/?authuser=0>>. Acesso em nov. 2018.

- Conexões simultâneas: 100 usuários;
- Capacidade de armazenamento: 1 Gigabyte;
- Downloads dos dados do banco: 10Gigabytes por mês;
- Operações de upload: 20000 por dia;
- Operações de download: 50000 por dia;
- Operação de autenticação;

As cotas referentes a operações de download e upload são redefinidas a meia noite do horário do pacífico.

### 3.2.3 – REGRAS DE ESCRITA E USO

A documentação do bando de dados Firebase é bastante detalhado quanto as regras de uso e escrita, permitindo uma boa elaboração e estruturação do banco, as quais seguem as normas detalhadas na Tabela 2, a seguir.

Tabela 2 - Regras de Escrita do Firebase

| LIMITES                                | DETALHES                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Restrições em códigos de coleções      | <ul style="list-style-type: none"> <li>• Precisam ser caracteres UTF-8 válidos</li> <li>• Não podem ter mais de 1.500 bytes</li> <li>• Não podem conter uma barra (/)</li> <li>• Não podem consistir apenas em um único ponto (.) ou em pontos duplos (..)</li> <li>• Não podem corresponder à expressão regular <code>__.*__</code></li> </ul>     |
| Profundidade máxima das sub-coleções   | 100                                                                                                                                                                                                                                                                                                                                                 |
| Restrições em códigos de documentos    | <ul style="list-style-type: none"> <li>• Precisam ser caracteres UTF-8 válidos</li> <li>• Não podem ter mais de 1.500 bytes</li> <li>• Não podem conter uma barra (/)</li> <li>• Não podem consistir unicamente em um único ponto (.) ou em pontos duplos (..)</li> <li>• Não podem corresponder à expressão regular <code>__.*__</code></li> </ul> |
| Tamanho máximo do nome de um documento | 6KiB                                                                                                                                                                                                                                                                                                                                                |
| Tamanho máximo de um documento         | 1 MiB (1.048.576 bytes)                                                                                                                                                                                                                                                                                                                             |

|                                           |                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Restrições em nomes de campo              | Precisam ser caracteres UTF-8 válidos                                                                                                                                                                                                                                                                                                                                                            |
| Tamanho máximo de um nome de campo        | 1.500 bytes                                                                                                                                                                                                                                                                                                                                                                                      |
| Restrições em caminhos de campo           | <ul style="list-style-type: none"> <li>Os nomes de campo precisam ser separados por um único ponto (.)</li> <li>Cada nome de campo precisa estar entre crases, a menos que atenda aos seguintes requisitos: <ul style="list-style-type: none"> <li>O nome do campo contém apenas os caracteres a-z, A-Z, 0-9 e sublinhado (_)</li> <li>O nome do campo não começa com 0-9</li> </ul> </li> </ul> |
| Tamanho máximo de um caminho de campo     | 1500 bytes                                                                                                                                                                                                                                                                                                                                                                                       |
| Tamanho máximo do valor de um campo       | 1 MiB — 89 bytes (1.048.487 bytes)                                                                                                                                                                                                                                                                                                                                                               |
| Profundidade máxima dos campos em um mapa | 20                                                                                                                                                                                                                                                                                                                                                                                               |

Fonte: Site Firebase.<sup>11</sup>

### 3.2.4 – SEGURANÇA NO ACESSO AO BANCO

Ao criar um aplicativo que utilizará o banco de dados FIREBASE, é possível utilizar o próprio FIREBASE para autenticar e permitir o acesso ao banco, viabilizando e facilitando o gerenciamento de usuários. Através do *Firebase Authentication* é possível gerenciar os usuários por e-mail, números de telefone e provedores de identidade federados como Google, *Facebook*, *Twitter* entre outros.

Através das regras de segurança do Firebase é possível definir níveis de acesso ao banco, permitindo somente a leitura ou leitura e escrita dependendo do usuário, onde, após o *login* do usuário é possível acessar as informações básicas do perfil e controlar o acesso dele aos dados armazenados, e é possível também usar o *token* de autenticação fornecidos para verificar a identidade dos usuários.

Cabe salientar, que já existem bibliotecas prontas para diversas IDEs e linguagens de programação, permitindo o serviço em vários tipos de dispositivos, sejam eles Android, IOs, Web, entre outros.

<sup>11</sup> Disponível em: <<https://firebase.google.com/docs/firestore/quotas?hl=pt-br>>. Acesso em nov. 2018.

### 3.3 – ANDROID STUDIO

Para a criação do aplicativo para dispositivo móvel, no intuito de verificação e monitoramento em tempo real dos dados da rede elétrica, foi escolhido do Android Studio como Ambiente de Desenvolvimento Integrado (IDE), tendo em vista a familiaridade com a linguagem de programação da IDE, que é o JAVA, desta forma, foi criado um aplicativo *android*, além de que para teste do aplicativo foi utilizado o dispositivo móvel *Galaxy S8* da Samsung, cujo sistema operacional é o android 8.0.

O android Studio é a IDE oficial para desenvolvimento de aplicativos android, e é baseado no *IntelliJ IDEA*, o qual permite uma maior facilidade na escrita de códigos, pois tem uma abordagem não-intrusiva intuitiva, facilitando a escrita, depuração, refatoração e teste.

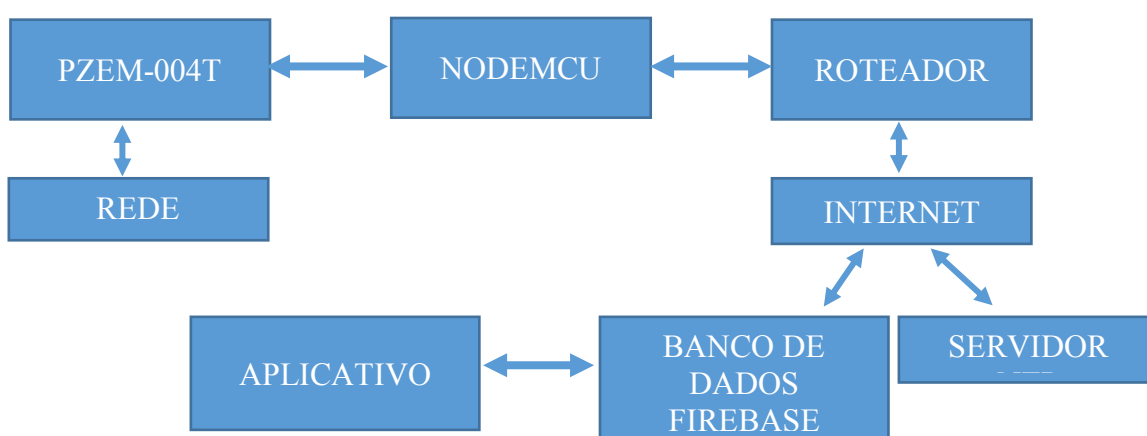
Graças a junção do IntelliJ IDEA, a escrita de códigos no Android Studio se torna muito prática e intuitiva, haja vista que, conforme ocorre a escrita o sistema vai identificado palavras chaves e classes mais utilizadas, e fazendo a predição de possível métodos, palavras ou variáveis a serem utilizados e ou replicados ao código.

A construção de telas se dá de duas formas, escrevendo um código em XML padrão ou usando a interface visual, onde é possível arrastar os ícones, ferramentas e outros recursos diretamente na tela, visualizando diretamente como ficaria a disposição dos componentes visuais e posteriormente faz-se a integração das telas com o código escrito em Java, dando funcionalidade a botões, *labels* e outros recursos.

O Android Studio, dentre suas funcionalidades, tem implementado um simulador de dispositivos móveis, com uma grande variedade de simulação de hardwares, permitindo assim o teste contínuo dos aplicativos em desenvolvimento, diminuindo consideravelmente o tempo que seria utilizado ao precisar testar o aplicativo em um dispositivo real.

#### IV - DESENVOLVIMENTO DO PROJETO (PROTOTIPO)

O entendimento de como funciona um sistema é fator fundamental para a construção de projetos que apresentem uma boa usabilidade e eficácia. Para o desenvolvimento do projeto foi primeiramente feito a análise dos requisitos, onde segundo SOMMERVILLE (2007, p. 49), tem como princípio, a definição dos objetivos do sistema, suas funcionalidades, as necessidades estruturais e as restrições do projeto. Partindo das informações obtidas foi possível a construção do fluxograma seguinte, que norteou o desenvolvimento do projeto.



O funcionamento se dará da seguinte forma:

- O NodeMCU conecta ao roteador através da rede wireless e acessa um servidor NTP (*Network Time Protocol*) obtendo data/hora e acessa também o banco de dados FIREBASE, através de uma chave de acesso;
- O NodeMCU requisita do módulo PZEM-004T as informações de tensão e corrente;
- O módulo PZEM-004T faz a leitura da rede elétrica e obtém a tensão e a corrente da rede e envia estas informações para o NodeMCU;
- O NodeMCU fica registrando e acumulando os dados enviados pelo PZEM-004T;
- Após 30 segundos o NodeMCU envia as informações registradas (data, hora, tensão, corrente, consumo elétrico de 30 segundos) para o banco de dados FIREBASE, e zera os registros;
- O FIREBASE receberá os dados a cada 30 segundos e armazenará;

- O aplicativo instalado no dispositivo móvel requisitará ao FIREBASE o acesso as informações através de um processo de autenticação, e em caso positivo, o banco de dados fornecerá o que for requisitado, e o aplicativo mostrará ao usuário através da interface.

#### 4.1 – MONTAGEM E CONFIGURAÇÃO DOS DISPOSITIVOS

Os *hardwares* utilizados no projeto foram o NodeMCU esp8266-12E (wifi), a base do NodeMCU e o modulo de medição de tensão, corrente e potência PZEM-004T. Após análise dos *datasheets*<sup>12</sup> dos dispositivos, foi verificado uma divergência quanto a tensão de trabalho entre o NodeMCU e o PZEM-004t, pois, enquanto o NodeMCU opera com 3.3v o PZEM-004t utiliza 5v, desta forma, quando o módulo PZEM-004T envia informação através do TX, a tensão de trabalho é 5 v, e caso fosse utilizado a ligação direta entre os dispositivos, seria possível, em algum momento o comprometimento da operação normal do NodeMCU, sendo assim, foi desenvolvido um divisor de tensão, reduzindo a tensão de 5v para 3.3 v, gerando uma estrutura de proteção para o sistema.

O divisor de tensão foi construído baseado na equação padrão para cálculo de tensão de saída (lei de Kirchhoff e Ohm), onde a tensão tem relação com as resistências e a tensão de entrada, desta forma:

Equação 1 - Divisor de tensão.

$$V_{out} = \frac{R_2}{(R_1 + R_2)} \times V_{in}$$

Fonte: Elaborado pelo autor.

Onde:

$V_{in}$  = tensão de entrada

$V_{out}$  = tensão de saída

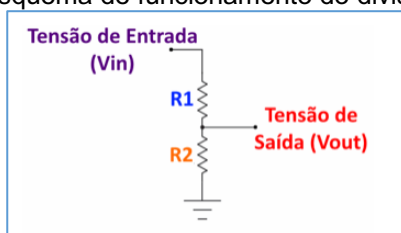
$R_1$  = resistência 1

$R_2$  = resistência 2

---

<sup>12</sup> Documento que especifica as características técnicas de um dispositivo eletrônico ou software, permitindo uma melhor manipulação e obtenção dos recursos disponíveis.

Figura 9 - Esquema de funcionamento do divisor de tensão



Fonte: Elaborado pelo autor.

O divisor (figura 10) de tensão foi montado utilizando um resistor de 1000 ohms como  $R_1$  e como  $R_2$  foi utilizado dois resistores de 1000ohms em série, já que não possuía um resistor de 2000ohm, desta forma, como a tensão de entrada é de 5v, a tensão de saída seria 3.33v, sendo a tensão ideal para o NodeMCU.

Equação 2 - Calculo do divisor de tensão.

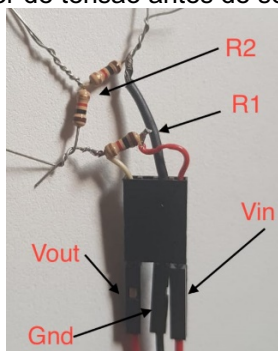
$$V_{out} = \frac{2000}{1000 + 2000} \times 5$$

$$V_{out} = 3.33 \text{ v}$$

Fonte: Elaborado pelo autor.

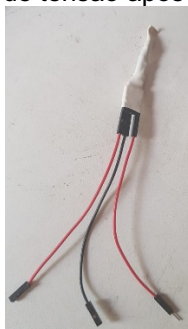
Para facilitar a conexão com os pinos do NodeMCU e do PZEM-004t foi soldado os resistores em um conector tipo b, removido de uma placa mãe queimada, ficando estruturado como mostrado na figura 10 e 11.

Figura 10 -Divisor de tensão antes de solda e isolamento.



Fonte: Elaborado pelo autor.

Figura 11 - Divisor de tensão após solda e isolamento.

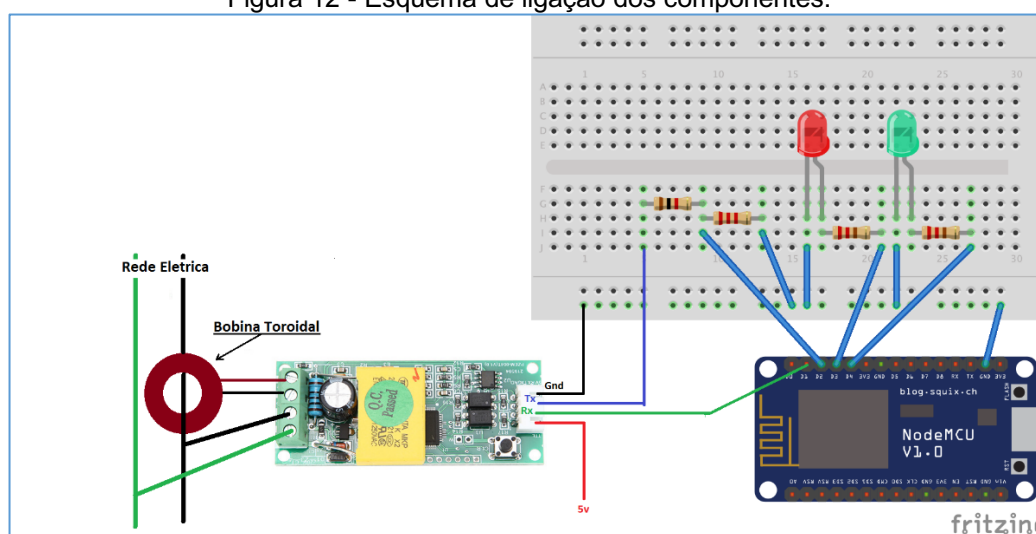


Fonte: Elaborado pelo autor.

Após o planejamento e adequação elétrica dos componentes foi feito a interligação dos mesmos, e para isso foi utilizado a carcaça de um receptor de tv por assinatura, que estava danificado, reutilizando também os leds já existentes na estrutura. A definição das ligações (figura 12) seguiu o seguinte padrão:

- Pino D2 do NodeMCU = Rx;
- Pino D3 do NodeMCU = Tx;
- Pino D4 do NodeMCU = Led Vermelho;
- Pino D5 do NodeMCU = Led Verde;

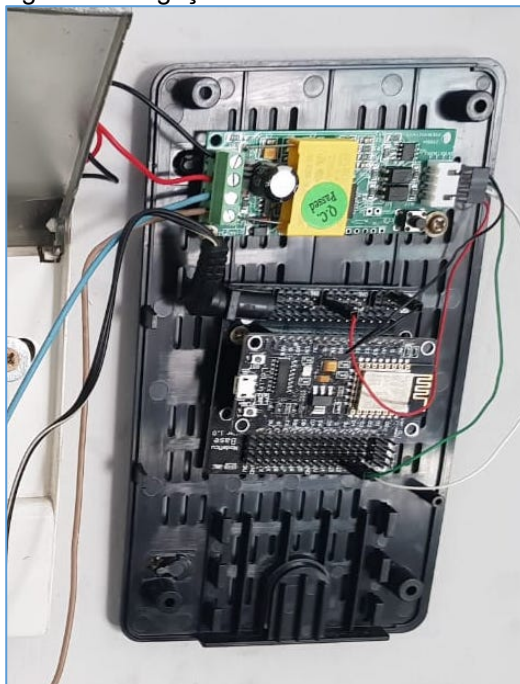
Figura 12 - Esquema de ligação dos componentes.



Fonte: Elaborado pelo autor.

Com o uso da carcaça do receptor foi possível uma melhor fixação do dispositivo junto a rede elétrica, sendo fixado ao lado da caixa de distribuição (figuras 13, 14, 15) da residência de teste, onde a bobina foi colocada dentro da caixa de distribuição (figura 16). Para proteção na manipulação dos componentes junto a rede elétrica, foi instalado uma tomada, para ligar a fonte de alimentação e um interruptor para ligar e desligar (figuras 14 e 15 parte inferior) a linha conectada no PZEM-004T, usado para medir a tensão da rede.

Figura 13 - Ligação NodeMCU e PZEM-004T



Fonte: Elaborado pelo autor.

Figura 14 - Visão da ligação dos componentes junto a rede elétrica.



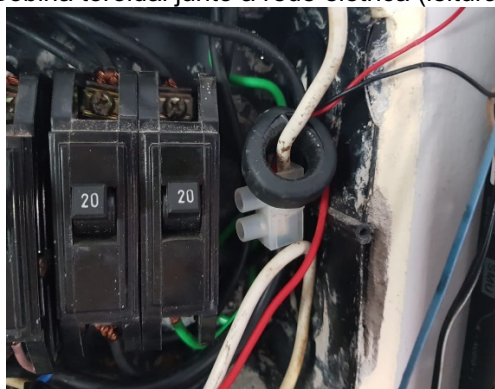
Fonte: Elaborado pelo autor.

Figura 15 - Visão do sistema montado



Fonte: Elaborado pelo autor.

Figura 16 - Bobina toroidal junto a rede elétrica (leitura da corrente).

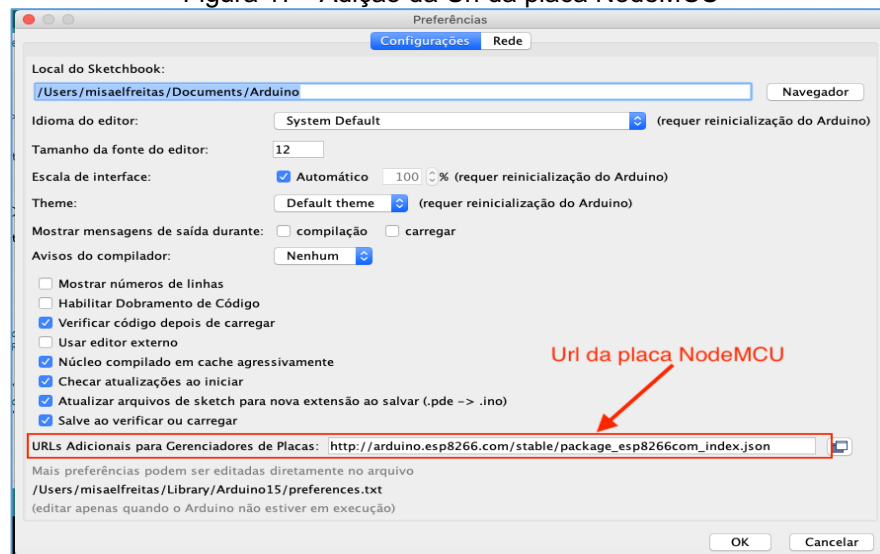


Fonte: Elaborado pelo autor.

## 4.2 – CODIGO FONTE DO NODEMCU.

Apesar do NodeMCU, ser originalmente construído para uso da linguagem de programação LUA, o código foi desenvolvido usando a IDE Arduino. Tendo em vista que a linguagem LUA é baseada em linguagem C, a codificação na IDE Arduino é totalmente compatível, sendo preciso apenas instalar a biblioteca do NodeMCU na IDE Arduino, utilizando o gerenciador de placas da IDE, e adição da URL [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json), nas configurações gerais (figura 17).

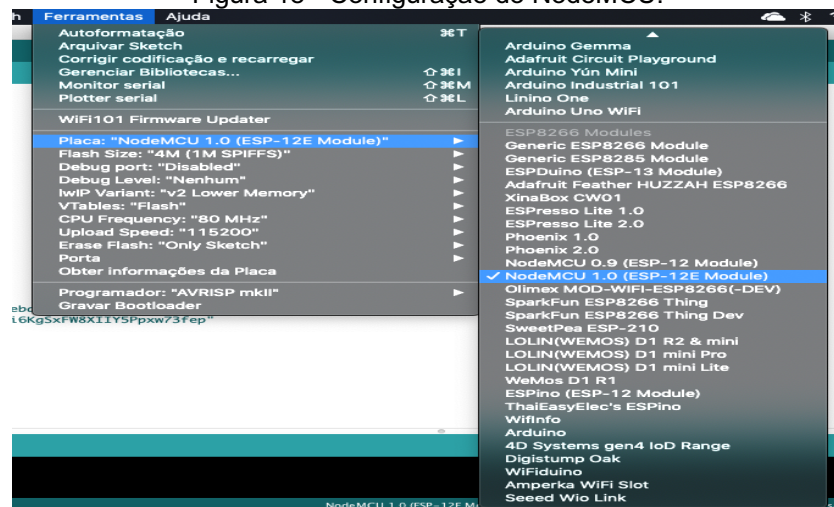
Figura 17 - Adição da Url da placa NodeMCU



Fonte: Elaborado pelo autor.

A partir da instalação da placa, no menu Ferramentas->Placas, já é possível a visualização do NodeMCU 1.0(ESP-12E Module) (figura 18) como opção de desenvolvimento.

Figura 18 - Configuração do NodeMCU.

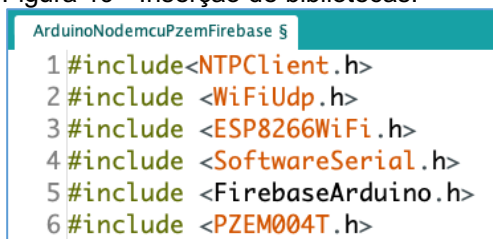


Fonte: Elaborado pelo autor.

### 4.2.1 – ADICIONANDO BIBLIOTECAS

Para facilitar o uso e melhorar o desenvolvimento do projeto foi inserido algumas bibliotecas ao código, a saber:

Figura 19 - Inserção de bibliotecas.



```
ArduinoNodemcuPzemFirebase 5
1 #include <NTPClient.h>
2 #include <WiFiUdp.h>
3 #include <ESP8266WiFi.h>
4 #include <SoftwareSerial.h>
5 #include <FirebaseArduino.h>
6 #include <PZEM004T.h>
```

Fonte: Elaborado pelo autor.

- Biblioteca NTPClient.h: Usada para configurar os dados recebidos do servidor NTP;
- Biblioteca WiFiUdp.h: Facilita a conexão com o servidor NTP do sitio <https://www.pool.ntp.org/zone/br;>
- Biblioteca ESP8266WiFi.h: responsável pelo gerenciamento da conexão wireless, bem como o envio e recebimento de dados da internet;
- Biblioteca SoftwareSerial.h: utilizado para permitir a visualização e manipulação de dados usando o terminal da IDE Arduino;
- Biblioteca FirebaseArduino.h: responsável pela manipulação dos dados do banco de dados Firebase Database. Através dela é feito a autenticação para acesso ao banco, configuração da mensagem com os dados a serem inseridos no banco e envio de mensagem solicitando o retorno de dados gravados no banco;
- Biblioteca PZEM004T.h: usada para facilitar a comunicação serial com NodeMCU, permitindo o uso de poucas linhas de código para obtenção dos dados usando o modulo;

#### 4.2.2 – DIFINIÇÕES GERAIS - CODIGO NODEMCU

É preciso informar que a pinagem do Arduino difere da nomenclatura utilizada no NodeMCU, desta forma foi necessário definir a equivalência dos pinos e seus respectivos identificadores, como pode ser observado na tabela 2 a seguir.

Tabela 3 - Equivalência de pinos do NodeMCU e Arduino

| NodeMCU   | Arduino |
|-----------|---------|
| <b>D0</b> | 16      |
| <b>D1</b> | 5       |
| <b>D2</b> | 4       |
| <b>D3</b> | 0       |
| <b>D5</b> | 2       |
| <b>D5</b> | 14      |
| <b>D6</b> | 12      |
| <b>D7</b> | 13      |
| <b>D8</b> | 15      |
| <b>A0</b> | 17      |

Fonte: Elaborado pelo autor.

Como o modulo PZEM004T utiliza comunicação serial, foi preciso definir quais pinos do NodeMCU seriam utilizados como Rx e Tx (figura 20), desta forma, foi utilizado os pinos 4 como Rx e o 5 como Tx. Para o acionamento dos LED's definiu-se os pinos 0 para o led verde e o pino 2 para o led vermelho.

Figura 20 - Definição dos pinos do NodeMCU.

```

8 PZEM004T pzem(4,5); // d2=4 Rx d1=5 Tx
9 IPAddress ip(192,168,1,1);
10
11 int ledverde = 0; //d4
12 int ledvermelho = 2; //d3

```

Fonte: Elaborado pelo autor.

Por definição da biblioteca do PZEM004T, é utilizado o comando “IPAddress ip(192,168,1,1);” no entanto, ele não tem relação com conexões de internet, é apenas um padrão utilizado para inicialização do modulo. O termo “ip” foi utilizado apenas para facilitar a reutilização no código, mas não significa “protocolo de internet”.

No bloco inicial é definido (figura 21) também o nome da rede wireless e sua senha, como também é definido a URL de acesso ao banco de dados Firebase, a chave de autenticação e o nome de referência do banco de dados a ser utilizado.

Figura 21 - Definições no código do NodeMCU

```

23 #define FIREBASE_HOST "fir-app-d753d.firebaseio.com" //url de acesso ao Firebase
24 #define FIREBASE_AUTH "ynRARPzVysMMtCqxFi6KgSxFW8XIIY5Ppxw73fep" // chave de autenticação
25 #define WIFI_SSID "Nome da Rede" // nome da rede wireless
26 #define WIFI_PASSWORD "Senha da Rede" // senha da rede wirelles
27 #define TABLE_NAME "leitura" // referência de escrita e leitura no Firebase

```

Fonte: Elaborado pelo autor.

Foi definido também algumas variáveis globais observados na figura 22, a seguir:

Figura 22 – Definição de Variáveis globais no código do NodeMCU.

```

14 WiFiUDP ntpUDP;
15 NTPClient timeClient(ntpUDP);
16
17 String formatarDataHora, data, hora;
18 float consumoAcumulado=0;
19 float tensao=0;
20 float corrente=0;

```

Fonte: Elaborado pelo autor.

#### 4.2.3 – CONFIGURAÇÃO DE INICIALIZAÇÃO DO NODEMCU (“void setup()”)

No início é definido a velocidade de comunicação (*baud rate*) para 115200 (figura 23 – A) do NodeMCU tendo em vista que esta é a velocidade padrão utilizada pelo módulo PZEM004T, evitando assim problemas de comunicação entre os dispositivos. É definido os pinos dos leds vermelho e verde como saída (figura 23 – B) e que eles inicializarão em modo *high* (acessos)(figura 23-C).

Na figura 23-D faz-se a inicialização do acesso a rede wireless, onde é inserido um loop que permite que o NodeMCU só passe para próxima etapa da configuração após conectar na rede. Após conectar na rede wireless, é inicializado a comunicação com o servidor NTP (figura 23 – E), inserindo a indicação para alteração do fuso horário padrão para o da cidade de Belém, e o comando “timeCliente.forceUpdate()” força a atualização para o horário atual.

Para conexão com o banco de dados FIREBASE (figura 23 – F) é utilizado o comando “Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH)”, sendo “FIREBASE\_HOST” o endereço URL indicado no início do código e “FIREBASE\_AUTH” a chave de autenticação obtida ao se construir o banco de dados. Para estabelecer a comunicação com o módulo PZEM004T usa-se o comando “pzem.setAddress(ip)”.

Figura 23 - Código do *Void Setup* do NodeMCU.

```

31 void setup() {
32   Serial.begin(115200); // baud rate A
33
34   pinMode(ledverde, OUTPUT); // define o ledverde como saída B
35   pinMode(ledvermelho, OUTPUT); // define o ledvermelho como saída
36
37   digitalWrite(ledvermelho, HIGH); //ao iniciar o led vermelho acende C
38   digitalWrite(ledverde, HIGH); //ao iniciar o led verde acende
39   delay(2000);
40
41   WiFi.begin(WIFI_SSID, WIFI_PASSWORD); // inicializa a conexão wireless D
42   while (WiFi.status() != WL_CONNECTED) { // aguarda a conexão com a rede wireless
43     delay(200);
44   }
45
46   timeClient.begin(); //Inicializa a operação para obtenção de data e hora E
47   timeClient.setTimeOffset(-10800); //seta o fuso horário de Belém
48   timeClient.forceUpdate(); // força a atualização para o fuso horário local
49
50   Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); //Inicializa a conexão com do FIREBASE, F
51   //e faz a autenticação.
52   pzem.setAddress(ip); //inicializa a comunicação com o módulo PZEM004T G
53 }

```

Fonte: Elaborado pelo autor.

#### 4.2.4 – VOID LOOP

Logo de início foi inserido um bloco de código (figura 24) para verificar se ocorreu a conexão com o roteador wireless, em caso positivo o led verde acende e o vermelho apaga, em caso negativo o led vermelho acende e o led verde apaga, e é chamada a função que faz a conexão com o roteador.

Figura 24 -Verificação de conexão com o roteador wireless

```

79 if (WiFi.status() == WL_CONNECTED){
80   digitalWrite(ledvermelho, LOW); //apaga o led vermelho
81   digitalWrite(ledverde, HIGH); // acende o led verde
82
83 }else{
84   digitalWrite(ledvermelho, HIGH); //acende o led vermelho
85   digitalWrite(ledverde, LOW); //apaga o led verde
86   WiFi.begin(WIFI_SSID, WIFI_PASSWORD); // caso esteja desconectado ele tenta conectar automaticamente
87   delay(2000);
88 }

```

Fonte: Elaborado pelo autor.

Como seria necessário o envio de dados relativos a tensão, corrente, data, hora, e consumo para o banco de dados Realtime Databases Firebase, que utiliza Javascript Object Notation (JSON), foi necessário a utilização da biblioteca ArduinoJson, permitindo o envio de dados usando o protocolo de transmissão de informação no formato texto, onde foi definido um tamanho de buffer igual a 500 (figura 25), evitando um estouro de memória e o comprometimento dos dados, caso a mensagem ultrapassa-se o tamanho do buffer.

Figura 25 - Definição do ArduinoJson.

```

64 StaticJsonBuffer<500> jsonBuffer;
65 JsonObject& dados = jsonBuffer.createObject();

```

Fonte: Elaborado pelo autor.

Para evitar que a mensagem gerada ultrapassasse o tamanho do buffer, foi evitado o uso de cálculos de divisão, impedindo assim, a geração de *floats* com casas decimais demasiadamente grande. Desta forma, a manipulação dos cálculos referentes ao consumo e valor de pagamento, ficou sobre responsabilidade do aplicativo.

No intuito de diminuir a quantidade de dados a ser inseridos no banco de dados, foi definido um loop em que o envio de informações para o bando de dados é feito em um intervalo de 30 segundos (figura 26).

Figura 26 - Código do laço for para leitura de 30 segundos do PZEM-004T.

```

112 for (int i =0; i<15; i++){ A
113     tensao = pzem.voltage(ip);
114     corrente = pzem.current(ip);
115
116     if(tensao < 0.0){tensao = pzem.voltage(ip); B
117     } else{if(tensao<0.0){ tensao = 0;}}
118
119     if(corrente < 0.0){corrente = pzem.current(ip); C
120     }else{if(corrente<0.0){corrente=0;}};
121
122     consumoReal = (tensao * corrente); //em wats por hora
123     consumoPeriodo +=consumoReal; //consumo acumulado no intervalo de 30segundo D
124     delay(1000);
125 }

```

Fonte: Elaborado pelo autor.

Foi inserido a estrutura logica representado na figura 26 – B e C, como forma de não permitir que leituras com valores negativos de tensão e corrente, pudessem ser gerados, comprometendo os dados, tendo em vista, que ao fazer alguns testes, foi verificado que, caso a rede elétrica estivesse desligada, o terminal exibia valores negativos de tensão. Como o consumo é um produto entre a tensão e a corrente, o consumo energético seria negativo, caso um dos dois valores fosse negativo, desta forma ao fazer a somatória dos valores consumidos em determinado período (figura 26 - D), certos valores seriam subtraídos, comprometendo seriamente o valor do consumo real, causando distorção entre a leitura do dispositivo e a leitura do relógio da concessionaria de energia.

Após 30 segundos de leitura dá-se a criação da mensagem a ser enviada ao banco de dados, através do uso do objeto JSON “dados” (figura 25) no inicio do *loop*.

Para isso são definidos nomes que caracterizarão os valores no banco, como verificado figura 27 a seguir:

Figura 27 - Bloco de construção e envio de mensagem ao banco de dados.

```

138 dados["data"] = data;
139 dados["hora"] = hora;
140 dados["tensao"] = tensao;
141 dados["corrente"] = corrente;
142 dados["whora"] = consumoReal;
143 dados["w30seg"] = consumoPeriodo*2; // consumo no periodo de 30 segundos
144
145 Firebase.push(TABLE_NAME, dados); // Comando para postar no firebase
146
147 if (Firebase.failed()) {
148     Serial.print(String(TABLE_NAME) + " erro ao enviar");
149     digitalWrite(ledvermelho, HIGH); digitalWrite(ledverde, LOW);
150     delay(1000);
151     Serial.print("Status Firebase:"); Serial.println(Firebase.error());
152 }

```

Fonte: Elaborado pelo autor.

O termo “push” visualizado na figura 27 – B, é utilizado para gerar uma chave de identificação da mensagem, pois diferente de um banco de dados relacional, o FIREBASE não utiliza chaves primarias na construção do banco, e a própria documentação do Firebase recomenda o uso do termo “push”, para gerar a chave de identificação das informações, evitando que seja criado identificadores com mesmo nome, caso o usuário defina algum termo como identificador, desta forma garante-se uma melhor organização do banco e redundância na criação de identificadores.

Por fim, é inserido um verificador de envio de informações ao banco de dados (figura 27-C), onde caso o comando “Firebase.failed()” retorne “true”, é mostrado no terminal (caso o computador esteja conectado ao NodeMCU) “erro ao enviar”, o led verde apaga e o led vermelho acende.

Cabe salientar que apesar do “laço for”, da figura 26-A, indicar que a variável “i” terá tamanho 15, o intervalo para cada interação é de 2 segundos, caracterizando assim um total de 30 segundos para encerramento do “laço”. Em virtude do tempo de interação do laço, foi necessário multiplicar por 2 o consumo obtido no intervalo de 30 segundos, já que neste período foram feitas apenas 15 leituras, ou seja, uma leitura a cada 2 segundos.

### 4.3 – BANCO DE DADOS DATABASE FIREBASE.

Como já descrito na metodologia, o Firebase Realtime Database é um banco de dados hospedado na nuvem, e os dados são armazenados como JSON. Diferentemente dos bancos de dados SQL, no Firebase não há tabelas, nem registros, e os dados são armazenados como objetos JSON, em uma árvore de objetos.

Quando se adiciona dados ao banco, esses dados se tornam um node na estrutura JSON com uma chave associada, onde o usuário pode definir as chaves, como códigos de usuários e nomes semânticos, ou pode solicitar que o próprio Firebase determine a chave, sendo a opção recomendada pela documentação oficial.

Para construir o banco de dados do projeto, foi necessário apenas fazer o *login* no site <https://firebase.google.com>, usando uma conta da Google e clicar em “Adicionar projeto” (figura 28), adicionar um nome ao projeto e aceitar os termos contratuais. A partir daí, já é possível adicionar dados ao banco, necessitando apenas fornecer ao dispositivo que irá enviar os dados, a identificação do banco, através de uma ID, identificada no código do NodeMCU URL (figura 21) como “FIREBASE\_HOST” e de uma chave secreta (FIREBASE\_AUTH) que autoriza a inserção de dados.

Figura 28 - Criando um banco de dados para o projeto



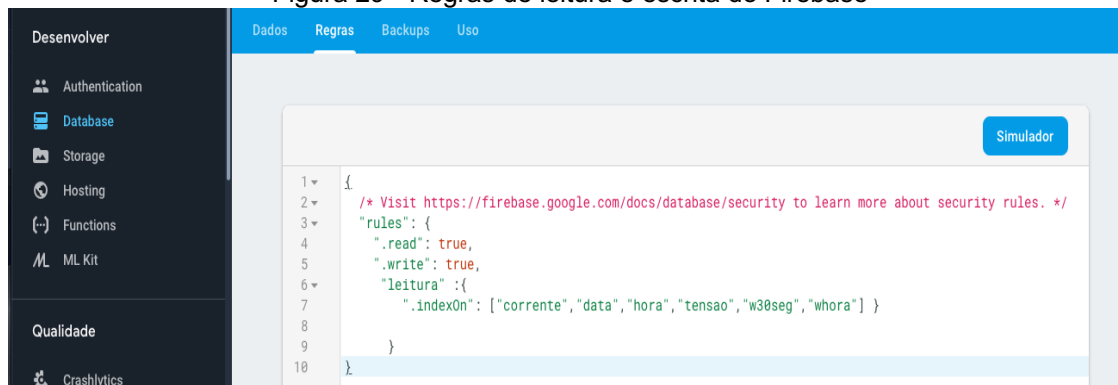
Fonte: Site Firebase.<sup>13</sup>

Além da chave de acesso é recomendado a utilização de regras de autenticação, a qual foi selecionado apenas o uso de “Email/senha”. Quanto as regras

<sup>13</sup> Disponível em: <<https://console.firebase.google.com/?hl=pt-br>>. Acesso em nov. 2018.

de leitura e escrita no banco de dados foi definido que qualquer usuário pode ler e gravar no banco (figura 29), tendo em vista que se trata de um protótipo, tendo apenas dois usuários.

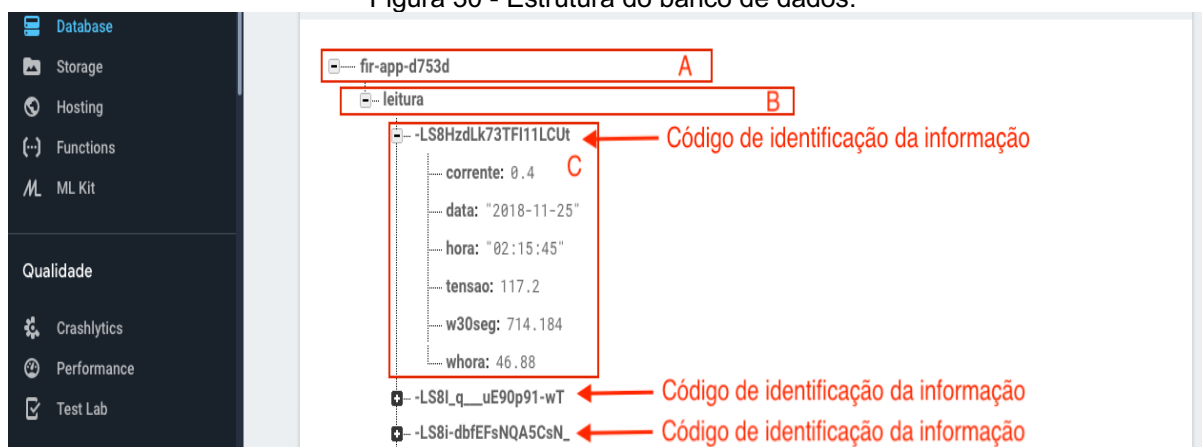
Figura 29 - Regras de leitura e escrita do Firebase



Fonte: Site Firebase.<sup>14</sup>

Após as configurações recomendadas pela documentação do Firebase, o banco já pode receber e enviar dados, sendo que os mesmos são aninhando segundo a figura 30. A estrutura do banco segue o padrão de uma árvore JSON, onde é definido um nó raiz (figura 30-A), e um outro nó definido como “pai” (figura 30-B) que foi nomeado como “leitura”, desta forma todos os dados serão inseridos no nó “pai”, e os dados inseridos (figura 30-C) são denominados de filhos, e são identificados por código único, gerado automaticamente pelo Firebase.

Figura 30 - Estrutura do banco de dados.



Fonte: Site Firebase.<sup>15</sup>

<sup>14</sup> Disponível em: <<https://console.firebase.google.com/project/fir-app-d753d/database/fir-app-d753d/rules>>. Acesso em nov. 2018.

<sup>15</sup> Disponível em: <<https://console.firebase.google.com/project/fir-app-d753d/database/fir-app-d753d/data?hl=pt-br>>. Acesso em nov. 2018.

#### 4.4 – DESENVOLVIMENTO DO1111 APLICATIVO ANDROID

Para desenvolver o aplicativo foi utilizado a IDE *Android Studio*, em virtude da familiaridade do desenvolvedor com a linguagem e o ambiente da IDE. A princípio foi buscado definir a arquitetura do programa, bem como as funções e classes necessárias. Cabe salientar que o aplicativo se destina a apenas exibir dados contidos no banco Firebase Database e calcular o valor a pagar, desta forma não foi necessário o desenvolvimento de classes que apresentassem uma complexidade computacional elevada e que requeressem o uso de arquiteturas de softwares mais elaboradas.

A partir da análise de requisitos foi verificado a necessidade de criação de quatro telas, sendo elas: Login, Home, Relatório por Período e Relatório de Consumo.

Através do aplicativo é possível executar as seguintes operações:

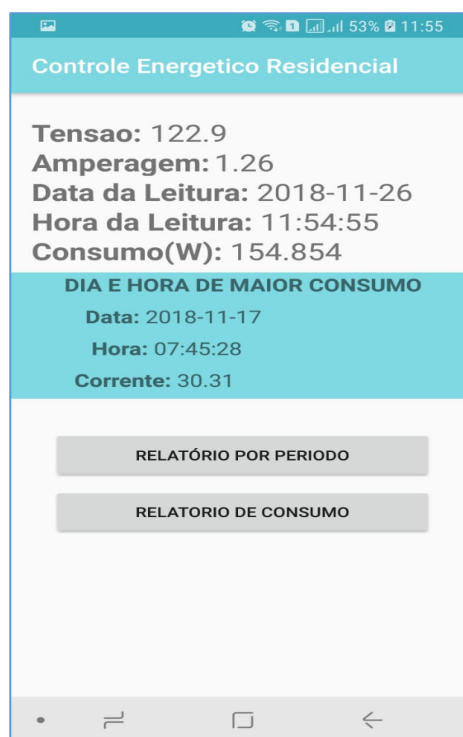
- Criar usuário para acesso ao banco de dados Firebase, através do botão cadastro (figura 31-A);
- Fazer login no banco de dados Firebase (figura 31-B) através do recurso *Firebase Authentication*;
- Após o login é exibido a tela Home (figura 32), onde é mostrado ao usuário a tensão, corrente (amperagem), a data e a hora da última leitura da rede elétrica. Na tela *home* existe dois botões que abrem as telas de exibição de relatório da rede elétrica por período (com vários dados) e relatório do consumo, que mostra apenas a quantidade de quilowatts consumido e o valor a pagar do dia selecionado;
- Na tela de relatório de consumo (figura 33) é mostrado um calendário, onde o usuário seleciona a data (um dia específico) que deseja visualizar o consumo em quilowatts e o valor a pagar;
- Na tela relatório por período (figura 34) é possível selecionar o período em que deseja saber os dados de tensão, corrente, consumo em watss/hora, o total de leituras, o consumo total do período e o valor a pagar. Para exibir os dados é necessário que o usuário insira o período, clicando nos campos denominados “Data inicial” e “Data final”, onde é possível digitar a data ou selecionar no calendário que será exibido, caso ele clique 2 vezes no mesmo campo. Após a seleção do período, basta clicar em um dos botões da tela, sendo que é possível exibir apenas as ultimas 20 leituras ou todas as leituras do período;

Figura 31 - Tela inicial.



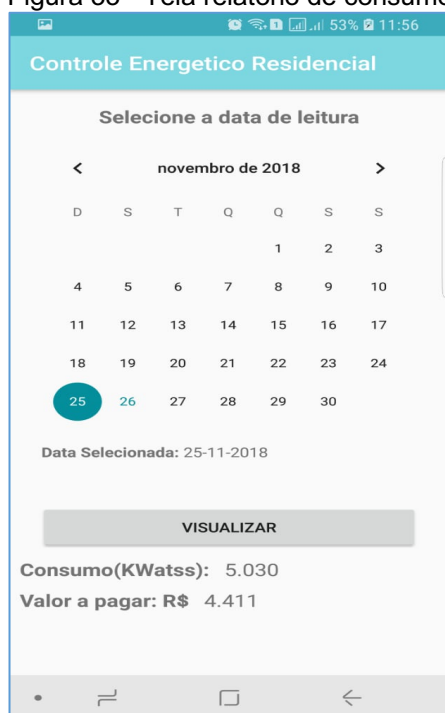
Fonte: Elaborado pelo autor.

Figura 32 - Tela Home do Aplicativo



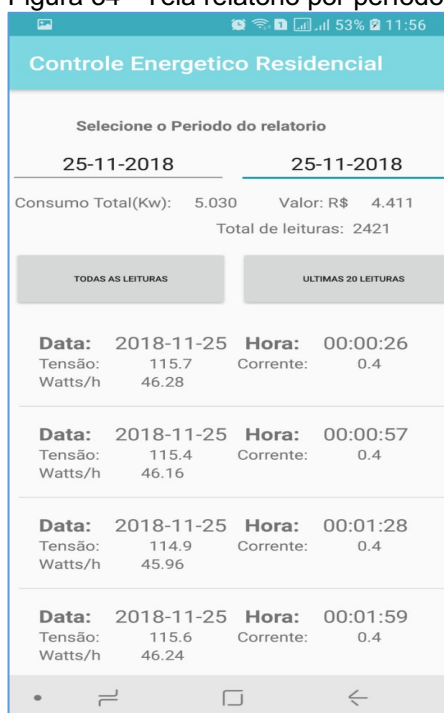
Fonte: Elaborado pelo autor.

Figura 33 - Tela relatório de consumo.



Fonte: Elaborado pelo autor.

Figura 34 - Tela relatório por período

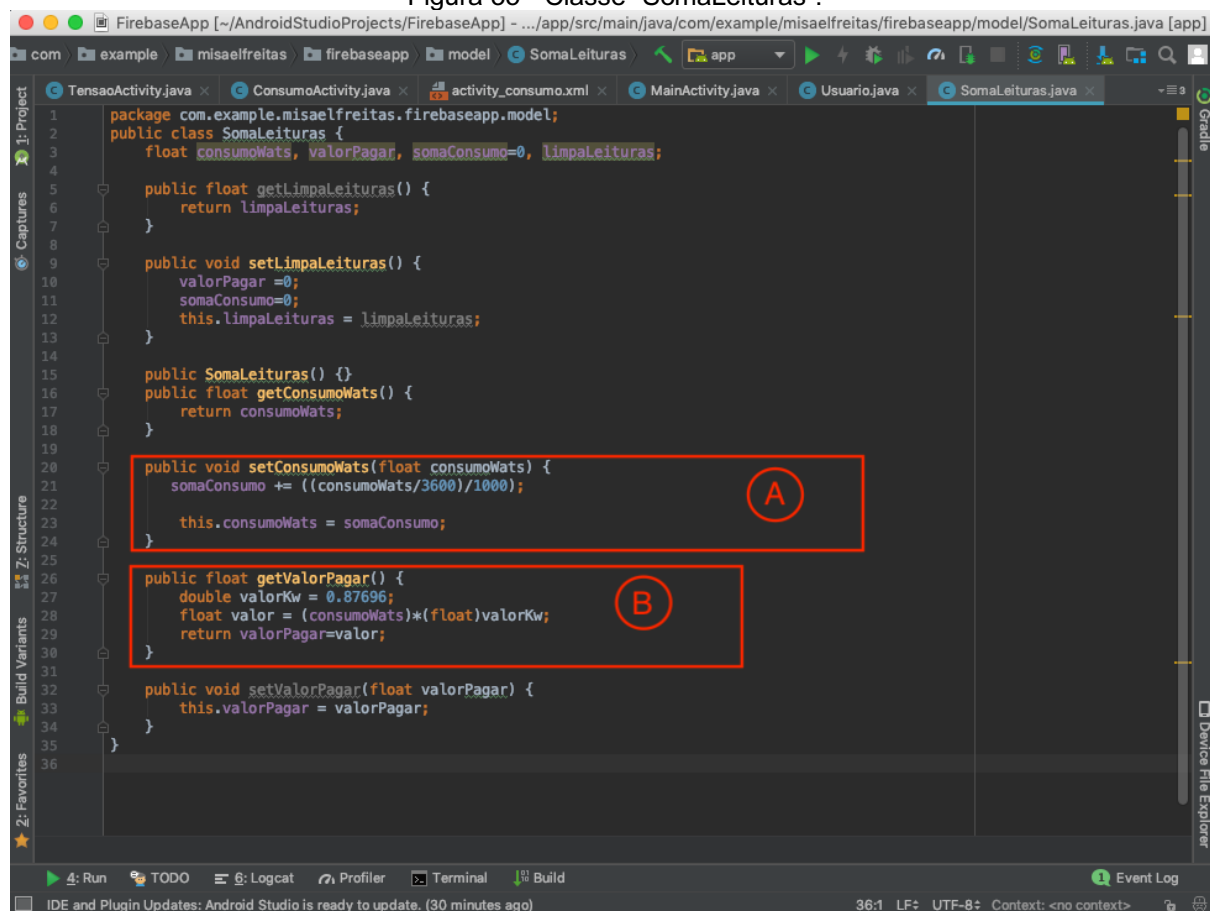


Fonte: Elaborado pelo autor.

Para determinar a quantidade de quilowatts consumida foi necessário converter o valor de watts/hora, que é a forma que o modulo NodeMCU mede, para watts/segundo, dividindo o valor fornecido pelo bando de dados por 3600 (tendo em vista que 1 hora possui 3600 segundos) e depois dividir o resultado por 1000,

transformando em quilowatts/hora (figura 35-A). Para a execução dos cálculos do consumo energético e valor a pagar foi criada uma classe denominada “SomaLeituras” (figura 35), onde o valor a pagar (figura 35-B) é calculado considerando o valor dos quilowatts em 0.87696, obtido do site da concessionária de energia elétrica do estado do Pará (<http://www.celpe.com.br/residencial/informacoes/cobranca-de-tarifas>), CELPA<sup>16</sup>, e que consta na resolução homologatória número 2.433/2018 da ANEEL.

Figura 35 - Classe “SomaLeituras”.



Fonte: Elaborado pelo autor.

<sup>16</sup> CELPA – Centrais elétricas do Pará. Concessionaria responsável pela distribuição de energia no estado do Pará.

## V - RESULTADOS

Com o hardware, o aplicativo e o banco de dados estruturados, deu-se início aos testes, para verificar a usabilidade e confiabilidade do sistema (Hardware e software). O primeiro item a ser testado foi o NodeMCU, verificando a capacidade de conexão e reconexão com o roteador wireless.

Após fazer o *upload* do código fonte para o NodeMCU, no qual foi adicionado o *SSID* da rede *wireless* e a senha, o comportamento esperado do *hardware*, seria que ao ligar na rede elétrica os dois leds deveriam acender e após conectar na rede *wireless*, o led vermelho deveria apagar, ficando apenas o led verde aceso, e assim ocorreu, como mostrado na figura 38 e 39.

Figura 36 - Inicialização do hardware



Fonte: Elaborado pelo autor.

Figura 37 - Conectado a rede wireless



Fonte: Elaborado pelo autor.

Como teste de reconexão, foi desligado o roteador e mantido ligado o NodeMCU, desta forma o led vermelho deveria acender e o verde apagar, e como esperado, o procedimento foi executado com sucesso. Após religar o roteador, o NodeMCU reconectou automaticamente a rede wireless, sendo confirmado através dos leds, já que o led vermelho apagou e led verde acendeu. Desta forma fica garantido a autonomia de operação, não necessitando de intervenção humana para

reconectar o sistema. Caso ocorra uma queda de energia e desligamento do sistema, ocorre a reconexão assim que a rede elétrica e a conexão com a internet entrar em operação.

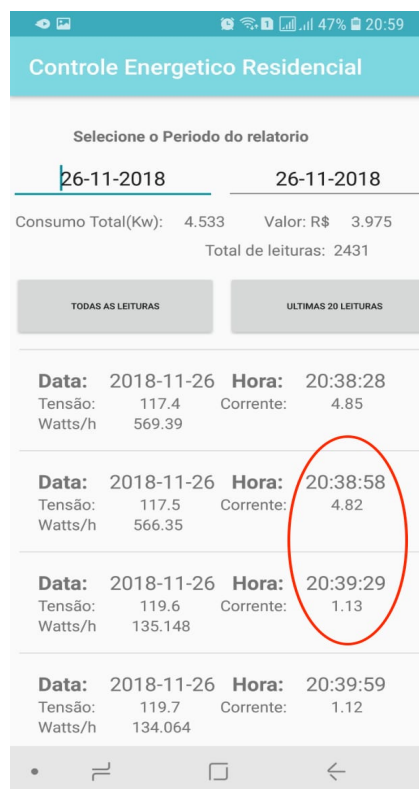
Outro teste executado foi em relação a precisão de leitura da rede elétrica, usando o modulo PZEM-004T, e para isso foi utilizado o alicate amperímetro THOMPSON e um aspirador de pó da Back+Decker de 750w. Ao ligar o aspirador de pó o alicate amperímetro registrou 3,4 amperes as 20 horas e 38 minutos (figura 39) e ao comparar com o que foi registrado pelo sistema (Figura 40) no mesmo horário, verificou-se uma amperagem igual 3,69 amperes, demonstrando a precisão do modulo PZEM-004T. Cabe ressaltar que o teste não foi executado em um sistema isolado, e sim usando toda a rede elétrica da casa, onde foi subtraído o valor da amperagem quando o aspirador de pó estava ligado pelo momento registrado após desligar o aspirador de pó, portanto, a diferença de 0.29 amperes da comparação dos dois medidores é perfeitamente compreensível, levando em consideração as alterações naturais da rede.

Figura 39 - Leitura de corrente com amperímetro.



Fonte: Elaborado pelo autor.

Figura 38 - Leitura com o aplicativo



Fonte: Elaborado pelo autor.

Para medir a tensão foi utilizado o multímetro digital ET-1002 da IMINIPA, e quando comparado com a tensão medida pelo sistema desenvolvido, ocorreu uma diferença de apenas 0.8 volts, sendo que o multímetro registrou 117,5 volts (figura 40) e o PZEM-004T 116,7 volts (figura 41), confirmando a precisão do módulo.

Figura 40 - Leitura da tensão usando um multímetro digital.



Fonte: Elaborado pelo autor.

Figura 41 - Leitura da tensão pelo PZEM-004T.

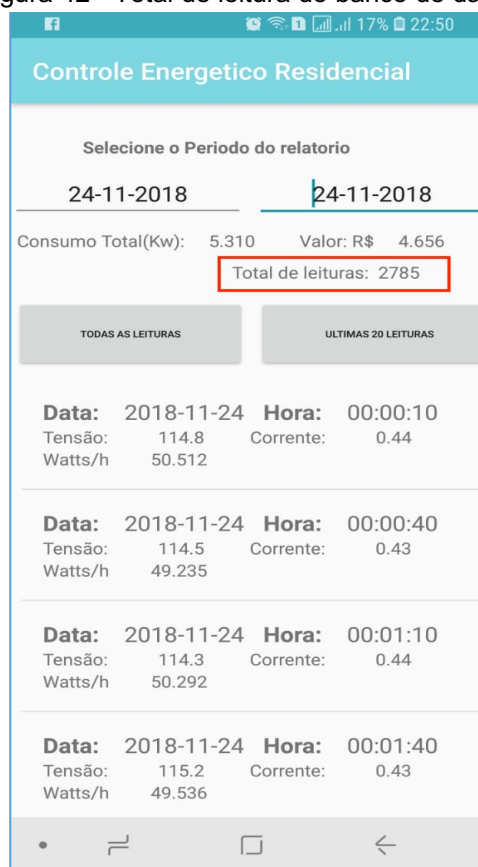


Fonte: Elaborado pelo autor.

Por fim, foi feito o teste do banco de dados Firebase, quanto a quantidade de dados registrados diariamente. Como o sistema envia dados para o Firebase a cada 30 segundos, espera-se que após um dia a quantidade de leitura seja igual a 2880

leituras. Ao solicitar o relatório do dia 24 de novembro de 2018 foi verificado que foi feito o registro de 2785 leitura (figura 42), dando uma diferença de 95 leituras a menos, ou seja, deu um erro de 2,73%. Após análise dos dados foi observado que não existe uma precisão quanto ao intervalo de alimentação do banco de dados, ocorrendo uma variação quanto ao tempo de envio, onde ocorre inserção no banco com 30, 31 e 32 segundos, desta forma, pode-se afirmar que a diferença real entre a quantidade de leituras registradas e o esperado é inferior ao calculado.

Figura 42 - Total de leitura do banco de dados.



Fonte: Elaborado pelo autor.

A partir a análise da quantidade de dados inseridos no banco de dados Firebase, foi possível verificar que a leitura de um dia ocupa apenas aproximadamente 400kbytes, garantindo assim, a possibilidade de fazer uso gratuitamente do banco de dados na nuvem por no mínimo 2,6 anos.

Conforme ajustado e testado o sistema, deu-se inicio ao período de arquivamento dos dados da rede elétrica, no intuito de comparar o consumo registrado em determinado período pelo relógio da concessionaria de energia e pelo sistema desenvolvido. A alimentação do banco de dados iniciou-se no dia 08 de novembro de

2018, às 14 horas, mas, para efeito de análise, passou-se a considerar os dados lidos a partir das 0 horas do dia 08 de novembro de 2018, até às 23 horas e 59 minutos do dia 27 de novembro de 2018. O dia 28 de novembro foi escolhido como data limite, pois, foi o dia em a concessionaria de energia registrou o último consumo energético da residência, e gerou o documento tarifário (Anexo1).

Para obter os dados inseridos no Firebase foi utilizado o aplicativo desenvolvido no projeto, denominado “CEnergy – Controle Energético”, através da tela “Relatório de consumo”. Foi obtido o consumo energético de cada um dos dias do intervalo e inseridos em uma planilha do programa “Excel”, gerando a tabela 1 e o gráfico 1.

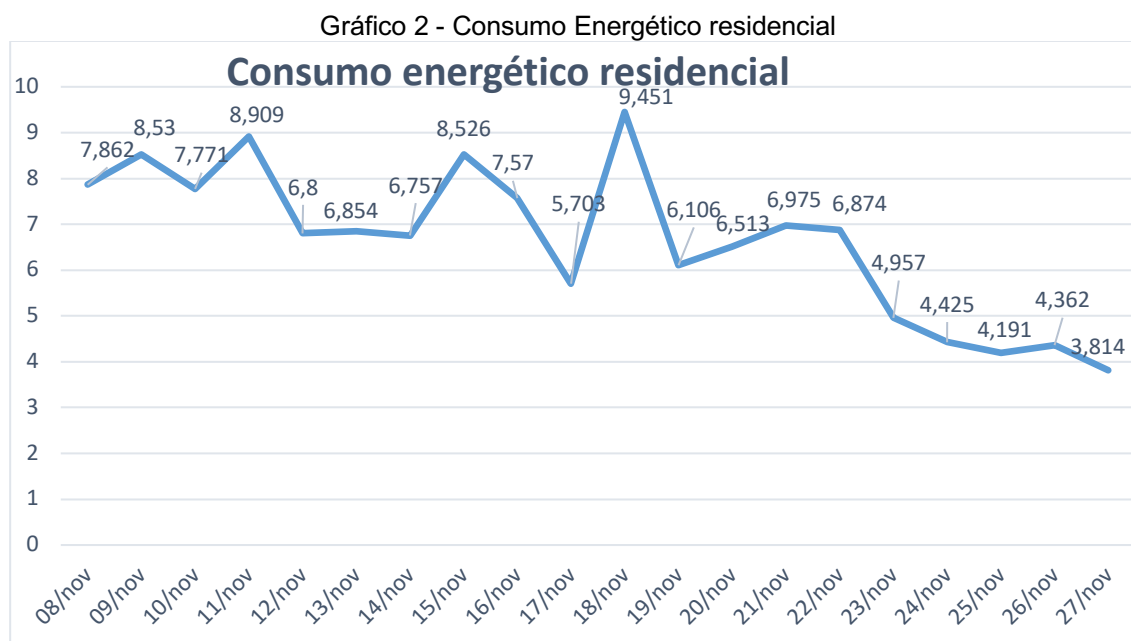
Tabela 4 - Tabela de consumo

| Data da leitura | Consumo diário (KW) | Dia da Semana | Media Semanal (KW) | Consumo semanal (KW) | Media diária (KW) | Consumo total (KW) | Total de dias lidos |
|-----------------|---------------------|---------------|--------------------|----------------------|-------------------|--------------------|---------------------|
| 08/nov          | 7,862               | quinta        | 7,7877             | 46,726               | 6,6475            | 132,95             | 20                  |
| 09/nov          | 8,53                | sexta         |                    |                      |                   |                    |                     |
| 10/nov          | 7,771               | sábado        |                    |                      |                   |                    |                     |
| 11/nov          | 8,909               | domingo       |                    |                      |                   |                    |                     |
| 12/nov          | 6,8                 | segunda       |                    |                      |                   |                    |                     |
| 13/nov          | 6,854               | terça         |                    |                      |                   |                    |                     |
| 14/nov          | 6,757               | quarta        | 7,2323             | 50,626               |                   |                    |                     |
| 15/nov          | 8,526               | quinta        |                    |                      |                   |                    |                     |
| 16/nov          | 7,57                | sexta         |                    |                      |                   |                    |                     |
| 17/nov          | 5,703               | sábado        |                    |                      |                   |                    |                     |
| 18/nov          | 9,451               | domingo       |                    |                      |                   |                    |                     |
| 19/nov          | 6,106               | segunda       |                    |                      |                   |                    |                     |
| 20/nov          | 6,513               | terça         | 5,0854             | 35,598               |                   |                    |                     |
| 21/nov          | 6,975               | quarta        |                    |                      |                   |                    |                     |
| 22/nov          | 6,874               | quinta        |                    |                      |                   |                    |                     |
| 23/nov          | 4,957               | sexta         |                    |                      |                   |                    |                     |
| 24/nov          | 4,425               | sábado        |                    |                      |                   |                    |                     |
| 25/nov          | 4,191               | domingo       |                    |                      |                   |                    |                     |
| 26/nov          | 4,362               | segunda       |                    |                      |                   |                    |                     |
| 27/nov          | 3,814               | terça         |                    |                      |                   |                    |                     |

Fonte: Elaborado pelo autor.

Ao analisar a tabela 3 com o documento tarifário da concessionaria de energia, é possível verificar certa compatibilidade entre os dados, tendo em vista que a concessionaria registrou o consumo de 212KW, entre 26/10/2018 e 27/11/2018, totalizando 32 dias, enquanto que o sistema desenvolvido registrou em 20 dias um total de 132,95 KW, porém, se adotarmos a média diária de consumo, demonstrado

na tabela, multiplicado pelos 32 dias lidos pela concessionária, teremos um total de 212,704 KW, se equiparando ao registrado pelo relógio da concessionária.



Fonte: Elaborado pelo autor.

Observando o gráfico 2, é possível notar um decréscimo no consumo energético residencial diário, se tornando mais acentuado a partir do dia 19 de novembro, isso se justifica, pelo fato da diminuição do uso de uma central de refrigeração, tendo em vista que neste período, iniciou-se fortes chuvas no local, diminuindo a temperatura da região, tornando desnecessário o uso da mesma.

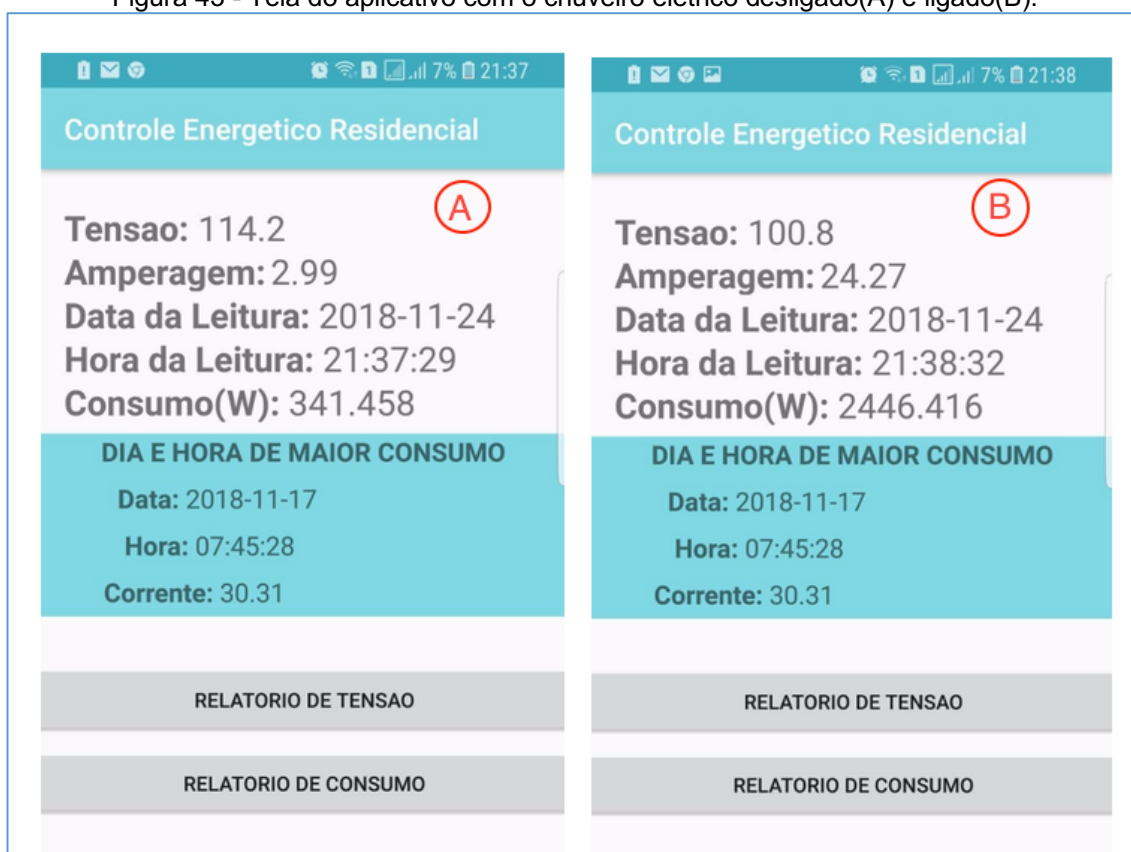
Para verificar como certos equipamentos interferem no consumo energético, foi testado um chuveiro elétrico da marca LORENZET, modelo Bella Ducha 4T de 5000w e uma central de refrigeração da marca Consul, modelo Bem Estar de 9000btus. Os resultados obtidos estão expressos na Tabela 4 e na figura 45.

Tabela 5 - Relação de consumo do chuveiro elétrico e Central de ar condicionado

|                    | Desligado | Ligado   | Aumento  |
|--------------------|-----------|----------|----------|
| Chuveiro           | 341,458   | 2446,416 | 616,46%  |
| Central de Ar      | 200,941   | 918,85   | 357,27%  |
| Chuveiro e Central | 200,26    | 3703,717 | 1749,45% |

Fonte: Elaborado pelo autor.

Figura 43 - Tela do aplicativo com o chuveiro elétrico desligado(A) e ligado(B).



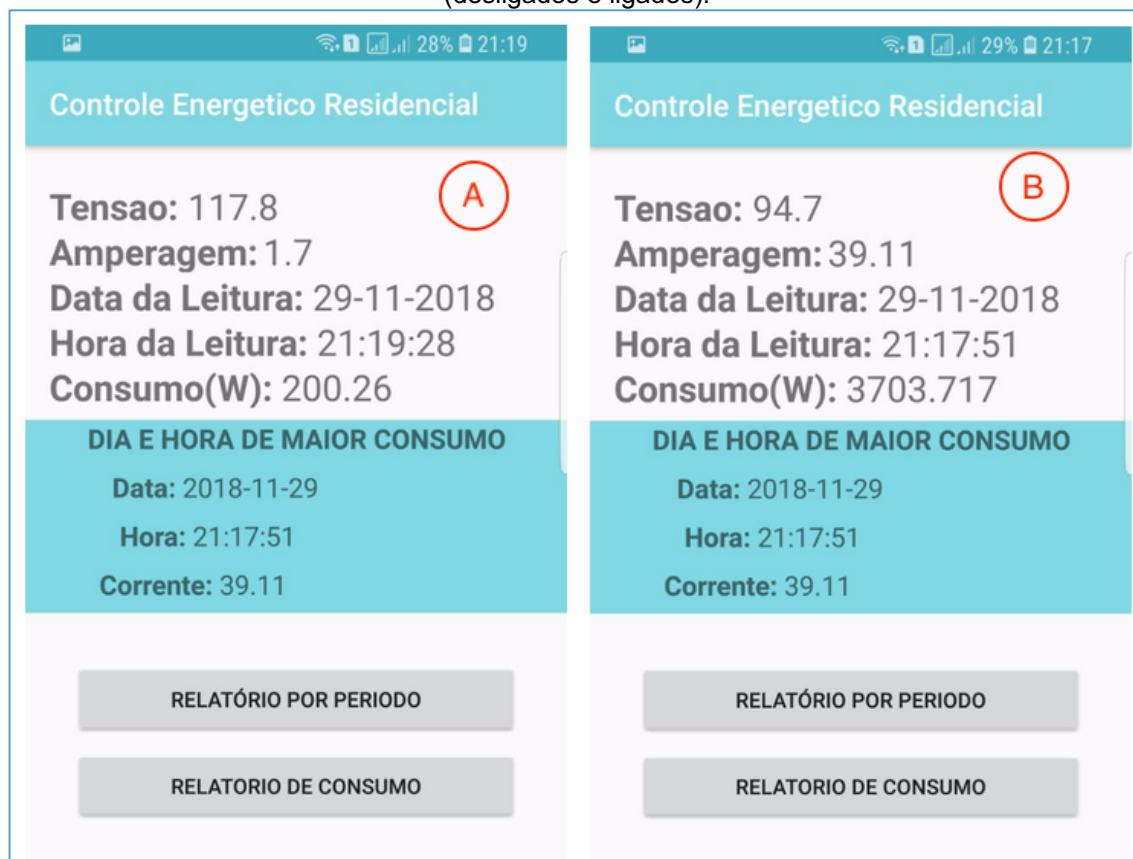
Fonte: Elaborado pelo autor.

Para verificar o consumo do chuveiro elétrico foi medido o consumo da rede elétrica antes de ligar (figura 43-A), na qual possuía uma geladeira e 11 lâmpadas ligadas, e depois de ligar (figura 43-B). Ao comparar a figura 43-A com a 43-B, pode ser observado que ao ligar o chuveiro elétrico, o consumo da rede aumentou 616,46%. A verificação de consumo da central de ar seguiu o mesmo padrão, e verificou-se o aplicativo antes de depois de liga-la, onde foi observado um aumento de 357,27%.

No intuito de verificar o impacto dos dois equipamentos na rede elétrica, foi observado o comportamento da rede elétrica com a central de ar e o chuveiro desligados (figura 44-A) e após ligar (figura 44-B). Pode-se concluir que os dois equipamentos (central de ar e chuveiro elétrico) são os maiores consumidores de energia elétrica da residência, como pode ser comprovado também através do gráfico 1, onde o período em que não foi usado a central de ar, o consumo residencial caiu cerca de 34%. Um outro fato interessante é em relação a corrente da rede elétrica quando o chuveiro elétrico e a central de ar estão ligados, pois foi atingido um corrente de 39,11 amperes, justificando o motivo que em alguns momentos, anteriores ao teste, o disjuntor da residência foi desligado, pois o disjuntor é de 30 amperes, desta forma,

é natural que ele desligue, demonstrando que ele está mau dimensionado para a residência.

Figura 44 - Comportamento da rede elétrica com chuveiro elétrico e central de ar condicionado (desligados e ligados).



Fonte: Elaborado pelo autor.

Foi possível concluir com os testes, que o sistema está operando de forma satisfatória, e é necessário ter uma maior consciência quanto ao uso do chuveiro elétrico e da central de ar condicionado, pois os mesmos consomem grande quantidade de energia, quando comparados com os outros equipamentos da residência.

## VI - CONSIDERAÇÕES FINAIS

Sem duvida o uso de energia elétrica trouxe grandes benefícios a humanidade, permitindo o desenvolvimento de muitos recursos tecnológicos, que nos dias atuais são primordiais para nosso bem estar, como climatizadores de ambientes, refrigeradores de alimentos, equipamentos de entretenimento (televisão, radio, computadores, celulares) entre outros, no entanto, a produção de energia tem um custo, tanto financeiro quanto ambiental.

O consumo consciente de recursos energéticos é cada vez mais necessário para a manutenção e garantia de serviços e produtos que necessitam de energia elétrica, tendo em vista que a maior parte da energia no mundo ainda não é obtida a partir de recursos renováveis, e mesmo em países, como o Brasil, onde a maior parte da energia elétrica é gerada através de hidrelétricas, a baixa nos reservatórios hídricos, pode causar situações em que requerem medidas drásticas, como racionamentos, semelhantes ao ocorridos entre 2001 e 2002.

Apesar de noticiários informarem da necessidade de uso consciente de energia elétrica, a falta de recursos para visualizar como a energia é consumida desfavorece a compreensão de como determinados equipamentos impactam no consumo elétrico. Partindo dessa premissa, o projeto foi desenvolvido, onde através do uso dos conceitos relacionados a internet das coisas, computação na nuvem e o uso de dispositivos móveis, foi criado uma forma do usuário visualizar, como o seu comportamento, quanto ao uso de energia, impacta no consumo elétrico e consequentemente na conta que irá pagar.

Mesmo sendo amplamente divulgado que certos equipamentos consomem muita energia, como chuveiro elétrico, ferro de passar, entre outros, somente através do uso do aplicativo desenvolvido foi possível a verificação de quão impactante é o consumo elétrico destes equipamentos. O uso de um chuveiro elétrico, por exemplo, aumentou em 8 vezes a amperagem da rede, passando de 2,99 amperes para 24,27 amperes, ou seja, o consumo elétrico que estava em 379,73 Watts/h passa para 3082,29 Watts/h. Esse impacto visual, causado pela grande alteração no consumo, gera no usuário a compreensão e a percepção de como o equipamento é realmente um “vilão” no consumo energético, tornando real e visual o que apenas era teórico, criando assim a consciência da necessidade do uso moderado do equipamento.

A visualização dos dados mostrados, só foram possíveis devido a ampliação de conhecimento relacionado a internet das coisas, onde o desenvolvimento de recursos computacionais, antes disponível somente a pessoas e empresas com recursos avançado, tornou possível a usuários comuns, e apresentando custos reduzidos. Neste projeto, por exemplo, foi gasto aproximadamente 170 reais, e obtido recursos e funcionalidade que estão disponíveis em equipamentos, demasiadamente caros, como é o caso do medidor e analisador elétrico com acesso remoto Dmi-P200, da empresa ISSO, que custa R\$ 3179,00. Desta forma, podemos compreender que o uso de IoT's é uma excelente alternativa para soluções tecnológicas de baixo custo.

Pode-se concluir que o modulo PZEM-004T, é excelente para o desenvolvimento de projetos que necessitam uma leitura precisa de redes elétricas, sendo superior ao alicate amperímetro utilizado no projeto, tendo em vista que ele conseguiu obter dados de correntes bastante pequenas, como a de uma lâmpada de 7 watts testada no projeto, dando 0,055 amperes, enquanto que o alicate amperímetro não obteve dados. Uma das dificuldades encontradas para o uso do modulo PZEM-004T foi em relação a manipulação do código, pois existe pouca documentação sobre o dispositivo, limitando as formas como ele pode ser utilizado.

Quanto ao banco de dados utilizado, o Firebase, pode se afirmar, que o mesmo apresenta uma ótima usabilidade, sendo uma excelente ferramenta no mundo IoT, podendo ser utilizada como banco de dados, como servidor de automação, servidor de autenticação, servidor para aplicativos de interação social, como o *Whatsapp* e *Messenger*, entre outras funcionalidades. O que mais chamou a atenção na plataforma da Google, foi a facilidade no uso, possuindo documentação de suporte para diversas linguagens de programação e sistemas operacionais, podendo ser utilizada para aplicativos IOs, Android, Windows e Web.

Considerando o poder computacional existente no NodeMCU, e na possibilidade de incluir novas funcionalidades ao aplicativo, a nova fase do projeto incluirá acoplamento de módulos atuadores no NodeMCU, permitido o controle remoto de equipamento da residência, como o acionamento de lâmpada, através do aplicativo, que exibirá também gráficos de consumo e outras opções de análise da rede elétrica.

Pode-se afirmar que o projeto foi desenvolvido de forma satisfatória, onde, os componentes usados apresentaram ótima funcionalidade, confiabilidade e acessibilidade. O aplicativo desenvolvido, apesar de básico, apresentou total

compatibilidade com o banco de dados na nuvem, garantindo a manipulação e análise precisa dos dados, permitindo a visualização do comportamento da sua rede elétrica residencial remotamente, e o consumo consciente.

## REFERÊNCIAS BIBLIOGRÁFICAS

ANEEL – Agência Nacional de Energia Elétrica. **Atlas de Energia Elétrica do Brasil**. 3. ed. Brasília: ANEEL, 2008. Disponível em: <<http://www.aneel.gov.br/arquivos/PDF/atlas3ed.pdf>>. Acesso em: 15 de outubro de 2018.

\_\_\_\_\_. **Atlas de Energia Elétrica do Brasil**. 1. ed. Brasília: ANEEL, 2002. Disponível em: <[http://www.aneel.gov.br/arquivos/pdf/livro\\_atlas.pdf](http://www.aneel.gov.br/arquivos/pdf/livro_atlas.pdf)>. Acesso em: 15 de outubro de 2018.

EPE – Empresa de Pesquisa Energética. Brasília. **Consumo anual de energia elétrica por classe (nacional) - 1995-2017**. Empresa de Pesquisa Energética, 2017. Disponível em: <<http://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/Consumo-Anual-de-Energia-Eletrica-por-classe-nacional>>. Acesso em 22 outubro de 2018.

\_\_\_\_\_. **Anuário Estatístico de Energia Elétrica 2017: ano base 2016**. Disponível em <http://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-160/topico-168/Anuario2017vf.pdf> Acesso em 01 de dezembro de 2018.

BLOG GEOGRAFIA CRITICA NA VEIA. **Maiores usinas hidrelétricas do Brasil**. Disponível em <https://geografiacriticanaveia.wordpress.com/2015/10/20/a-maiores-usinas-hidreletricas-do-brasil/> Acesso em 28 de novembro de 2018.

ESPRESSIF. **ESP8266EX Datasheet. Version 5.8, 2018 [English]**. Disponível em: <[https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf)>. Acesso em 02 de outubro 2018.

JACQUES, Luiz. **Saiba Verificar o Bom funcionamento do medidor de energia elétrica de sua instalação**. Saber Elétrica. Disponível em <<https://www.sabereletrica.com.br/medidor-de-energia-eletrica/>>. Acesso em 21 outubro 2018.

MANCINI, Mônica. **Internet das Coisas: História, Conceitos, Aplicações e Desafios**. USP, 2017. Disponível em: <<https://pmisp.org.br/documents/acervo-arquivos/241-internet-das-coisas-historia-conceitos-aplicacoes-e-desafios/file>>. Acesso em 24 setembro 2018.

MCROBERTS, Michael. **Arduino básico** - São Paulo - SP: Novatec Editora, 2011.

PINTO, Tales dos Santos. **O apagão energético de 2001**. *Brasil Escola*. Disponível em <<https://brasilecola.uol.com.br/historiab/apagao.htm>>. Acesso em 05 de dezembro de 2018.

SOKOLOV, Oleg. **Arduino communication library for Peacefair PZEM-004T Energy Monitor**. GitHub, 2018. Disponível em: <<https://github.com/olehs/PZEM004T>>. Acesso em 26 set. 2018.

THOMAZINI, Ana Paula Nickel. **O princípio da transparência nas relações de consumo**; Boletim Jurídico. Disponível em <<https://www.boletimjuridico.com.br/doutrina/artigo/1332/o-principio-transparencia-nas-relacoes-consumo>>. Acesso em 25 de novembro de 2018.

THOMSEN, Adilson. **Como programar o módulo ESP8266 NodeMCU**. Filipeflop. Disponível em < <https://www.filipeflop.com/blog/esp8266-nodemcu-como-programar/>> Acesso em 16 de outubro de 2018.

SITE FIREBASE. **Cotas e limites**. Disponível em <https://firebase.google.com/docs/firestore/quotas?hl=pt-br> Acesso em 05 de outubro de 2018.

\_\_\_\_\_. **Regras de segurança**. Disponível em < <https://console.firebase.google.com/project/fir-app-d753d/database/fir-app-d753d/rules>>. Acesso em 05 de outubro de 2018.

SOMMERVILLE, I. **Engenharia de Software**. 8. ed. São Paulo: Pearson Addison Wesley, 2007.

VILLARINO, Julia. **Internet das coisas: um desenho do futuro**. Proof. Disponível em <<https://www.proof.com.br/blog/internet-das-coisas/>> Acesso em 15 de outubro de 2018.

## ANEXOS

## Anexo 1 – Conta de energia elétrica da residência do mês de novembro de 2018.

2ª Via P 1/1



**Centrais Elétricas do Pará**  
Rodovia Augusto Montenegro, km 8,5 Belém - PA  
CEP: 66.823-010 | Ins. Estadual: 150.744.80-3 | CNPJ: 04.895.728/0001-80



**ANTONIO MISAEL LOPES DE FREITAS**  
R. JATOBA, S/N LOTE 114  
S MIGUEL DO GUAMA 68660-000 SAO MIGUEL DO GUAMA  
- PA  
CPF

Conta de Energia Elétrica/Nota Fiscal ISérie B 002259761  
Nº da Fatura 0201811002259761 ICPOP: 5258/AA  
Instalação 95756301

|                |                   |                |
|----------------|-------------------|----------------|
| Conta do mês   | Vencimento        | Conta Contrato |
| <b>11/2018</b> | <b>04/12/2018</b> |                |

Para atendimento, informe este número

**Dados da Instalação**

Classificação: Residencial Pleno - MONOFÁSICO  
Nº Parceiro de Negócio: 93340299 Tensão Nominal (V): 127 V  
Grupo e Subgrupo de Tensão: B/B1 Unidade de Leitura: MM16B002  
Tipo de Tarifa: CONVENCIONAL MONÔMIA Nº Medidor: 2971665  
Fator de Potência: 0,00

**Datas**

|            |              |                          |
|------------|--------------|--------------------------|
| Emissão    | Apresentação | Previsão próxima leitura |
| 27/11/2018 | 27/11/2018   | 27/12/2018               |

**Informações do consumo do mês + Tarifa sem Tributos**

|           |                       |                    |           |                 |
|-----------|-----------------------|--------------------|-----------|-----------------|
| Constante | Data Leitura Anterior | Data Leitura Atual | Qtd. Dias | Resolução Aneel |
| 1,00      | 26/10/2018            | 27/11/2018         | 32        | 2433/18         |

|                  |                  |               |         |                     |
|------------------|------------------|---------------|---------|---------------------|
| Canal de Leitura | Leitura Anterior | Leitura Atual | Consumo | Tarifa sem Tributos |
| ATIVO TOTAL      | 15.473           | 15.085        | 212 kWh | 0,67098             |

**Histórico do Consumo (kWh)**



**Informações de tributos**

| Base          | Alíquota | Valor    |
|---------------|----------|----------|
| Tributos ICMS | 215,57   | 25,0000% |
| PIS           | 215,57   | 1,3157%  |
| COFINS        | 215,57   | 6,0920%  |

**Composição do Consumo (R\$)**

| Compra de Energia | Transmissão | Distribuição(CELPA) |
|-------------------|-------------|---------------------|
| 53,82             | 8,53        | 47,29               |

| Encargos  | Perda em | Tributos | Outros |
|-----------|----------|----------|--------|
| Setoriais | Energia  |          |        |
| 12,38     | 23,69    | 69,88    | 22,98  |

**Período Fiscal: 27/11/2018**

**Reservado ao Fisco**

5489.C27D.0785.2B68.2FB8.0ED2.E5E9.0D63

**Informações para o cliente**

**Demonstrativo do Faturamento**

| FORNECIMENTO             | QUANTIDADE | PREÇO    | VALOR(R\$) |
|--------------------------|------------|----------|------------|
| Consumo                  | 212        | 0,992783 | 210,47     |
| Adicional Band. Amarela  |            |          | 2,64       |
| Adicional Band. Vermelha |            |          | 2,46       |
| <b>ITENS FINANCEIROS</b> |            |          |            |
| Cip-Ilum Pub Pref Munic  |            |          | 22,95      |



**Total a pagar:** R\$ 238,52

**Reaviso de vencimento**

**Níveis de Tensão Fornecido**

| Tensão Nominal(Volts) | Faixa de valores para limites min e max |
|-----------------------|-----------------------------------------|
| 127                   | 116 a 133                               |
| 220                   | 201 a 231                               |

Clientes cujos indicadores padrões de continuidade tenham sido violados deverão receber uma compensação financeira através de crédito na conta de energia, conforme critérios definidos no módulo 08 do PRODUST/ANEEL.

As regras para a cobrança da CIP de cada município atendido encontram-se na área de acesso público do site da CELPA.

As informações sobre os atendimentos comerciais realizados para a sua Unidade Consumidora podem ser obtidas no site da CELPA.

As informações de apuração dos Indicadores de Continuidade e Limites Aplicáveis podem ser obtidas no site da CELPA: www.celpa.com.br.

Incidirão sobre a conta paga após o vencimento multa de 2%, juros de mora de 0,0333% ao dia (conforme a Lei 10438/02) e atualização monetária com base no IGP-M a serem incluídos na próxima fatura.

Conforme Resolução Normativa Aneel 581/2013 Arts 7º e 8º é seu direito solicitar a qualquer tempo a CELPA o cancelamento de cobrança relativa de outros serviços cobrados na fatura, bem como a emissão da nova fatura sem a cobrança dos serviços cancelados. Ressalta-se que o fornecimento poderá ser suspenso caso os valores referentes aos serviços de distribuição de energia não sejam devidamente pagos.

As informações sobre as condições de fornecimento, tarifas, produtos, serviços prestados, tributos e informações complementares encontram-se disponíveis para a consulta nas Agências de Atendimento e na área reservada ao consumidor no site da CELPA.

|                                                                                                                                                 |                                                                                                                         |                                                                       |                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| <b>Central de Atendimento:</b> 0800 091 0196<br>Atendimento gratuito 24h.<br>Central de Atendimento para deficientes auditivos<br>0800 721 6340 | <b>Ovidueta Celpa:</b> 0800 091 8500<br>Ligação gratuita de telefones fixo e móvel, de segunda a sexta, das 08h às 18h. | <b>ARCON-PA:</b> 0800 727 9107<br>Ligação gratuita de telefones fixos | <b>Agência Nacional de Energia Elétrica (ANEEL) 167:</b><br>Ligação gratuita de telefones fixos e móveis. |
|-------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|

## Anexo 2 – Documentação do módulo PZEM-004(v3.0)

### AC digital display Multifunction Meter

**Product Type: PZEM-004(V3.0)**

#### A. Function

1. Electrical parameter measurement function (voltage, current, active power, energy).
2. Overload alarm function (over power alarm threshold the power flash to alarm).
3. Power alarm threshold preset function (can set power alarm threshold).
4. The reset function of energy key.
5. Store data when power off (store the accumulated energy before power off).
6. Bright red digital display function (display voltage, current, active power, energy).
7. Serial communication function (with TTL serial interface itself, can communicate with a variety of terminal through the pin board, read and set the parameters).

#### B. Front display and key

##### I. Display Interface

Display interface is formed by four bright red digital tubes, used to display the voltage, current, power, energy parameters.

##### II. Display Format

1. Power: Test Range: 0 ~ 22kW
  - Within 0 ~ 10kW, the display format is 0.000 ~ 9.999;
  - Within 10 ~ 22kW, the display format is 10.00 ~ 22.00.
2. Energy: Test Range: 0 ~ 9999kWh
  - Within 0 ~ 10kWh, the display format is 0.000 ~ 9.999;
  - Within 10 ~ 100kWh, the display format is 10.00 ~ 99.99;
  - Within 100 ~ 1000kWh, the display format is 100.0 ~ 999.9;
  - 1000 ~ 9999kWh and above, the display format is 1000 ~ 9999.
3. Voltage: Test Range: 80 ~ 260VAC
  - Display Format is 110.0 ~ 220.0.
4. Current: Test Range: 0 ~ 100A
  - Display Format is 00.00 ~ 99.99.

#### III. Key

There is a key on the panel, it can be used to reset energy.

The method of reset energy: Long press the key for 5 seconds until the digital on energy display window flicker, then release the key. Short press the key again, then the energy data is cleared and quit the flickering state, now the reset operation is completed; if long press for 5 seconds again until no longer flicker, it means exit the reset state.

#### C. Wiring diagram

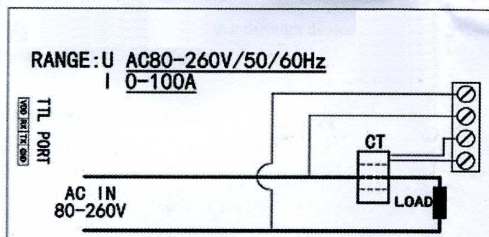


Figure 1 Wiring diagram

The wiring of this module is divided into two parts: the voltage and current test input terminal wiring and the serial communication wiring, as shown in Figure 1; according to the actual needs of the clients, with different TTL pin board to achieve communicate with different terminals.

#### D. Display Interface

The whole meter panel display window is formed by four windows, they are voltage, current, power and energy; the following are brief description of each parameter display:

##### 1. Voltage Display

Measure and display the current power frequency grid voltage.

##### 2. Current display

Measure and display the current load (appliances) current. There is supplementary instruction that the current test value is from the beginning of 10mA, but this module belongs to high power test equipment, if you care about the mA level current testing accuracy, it is not be recommended.

##### 3. Energy display

Measure and display the current accumulative power consumption. There is supplementary instruction that the minimum unit of the energy metering is 0.001kWh, which means it begins to accumulate from 1Wh, relatively speaking, the resolution is rather high, for the low-power (within 100W) load test, you can observe the accumulative process rather intuitively.

##### 4. Power display

Measure and display the current load power. There is supplementary instruction that the power test value is from the beginning of 0.001kW, which means it begins to test from 1W, but this module belongs to high power test equipment, if you have the requirement of the testing within 1W, it is not be recommended.

#### E. Serial communication

This module is equipped with TTL serial data communication interface, you can read and set the relevant parameters via the serial port; but if you want to communicate with a device which has USB or RS232 (such as computer), you need to be equipped with different TTL pin board (USB communication needs to be equipped with TTL to USB pin board; RS232 communication needs to be equipped with TTL to RS232 pin board), the specific connection type as shown in Figure 2. In the below table are the communication protocols of this module:

| NO. | function                      | Head | Data1- Data5                                                                             | Sum |
|-----|-------------------------------|------|------------------------------------------------------------------------------------------|-----|
| 1   | voltage                       | B0   | C0 A8 01 01 00 (Computer sends a request to read the voltage value)                      | 1A  |
|     |                               | A0   | 00 E6 02 00 00 (Meter reply the voltage value is 230.2V)                                 | 88  |
| 2   | current                       | B1   | C0 A8 01 01 00 (Computer sends a request to read the current value)                      | 1B  |
|     |                               | A1   | 00 11 20 00 00 (Meter reply the current value is 17.32A)                                 | D2  |
| 3   | Active power                  | B2   | C0 A8 01 01 00 (Computer sends a request to read the active power value)                 | 1C  |
|     |                               | A2   | 08 98 00 00 00 (Meter reply the active power value is 2200w)                             | 42  |
| 4   | Read energy                   | B3   | C0 A8 01 01 00 (Computer sends a request to read the energy value)                       | 1D  |
|     |                               | A3   | 01 86 9f 00 00 (Meter reply the energy value is 99999wh)                                 | C9  |
| 5   | Set the module address        | B4   | C0 A8 01 01 00 (Computer sends a request to set the address, the address is 192.168.1.1) | 1E  |
|     |                               | A4   | 00 00 00 00 00 (Meter reply the address was successfully set)                            | A4  |
| 6   | Set the power alarm threshold | B5   | C0 A8 01 01 14 (computer sends a request to set a power alarm threshold)                 | 33  |
|     |                               | A5   | 00 00 00 00 00 (Meter reply the power alarm threshold was successfully set)              | A5  |

#### Illustration of the communication protocol example:

##### 1. Set the communication address: 192.168.1.1

Send command: B4 C0 A8 01 01 00 1E

Reply data: A4 00 00 00 00 00 A4

### Anexo 3 – Código fonte de programação do NodeMCU.

```
//INCLUSAO DAS BIBLIOTECAS
#include<NTPClient.h>
#include <WiFiUdp.h>
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
#include <FirebaseArduino.h>
#include <PZEM004T.h>

//DEFINIÇÕES GERAIS
PZEM004T pzem(4,5);// d1=5 tx d2=4 rx
IPAddress ip(192,168,1,1);

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

String formatarDataHora, data, hora;
float consumoAcumulado=0;
float tensao=0;
float corrente=0;
int ledverde = 0;//d3
int ledvermelho = 2;//d4

#define FIREBASE_HOST "fir-app-d753d.firebaseio.com"
#define FIREBASE_AUTH "ynRARPzVysMMtCqxFi6KgSxFW8XIIY5Ppxw73fep"
#define WIFI_SSID "FREITASS"
#define WIFI_PASSWORD "casadeDeus10"
#define TABLE_NAME "leitura"

void setup() {
  Serial.begin(115200);
  pinMode(ledverde, OUTPUT);
  pinMode(ledvermelho, OUTPUT);

  digitalWrite(ledvermelho, HIGH);
  digitalWrite(ledverde, HIGH);

  delay(2000);

  // CONFIGURAÇÃO DA CONEXAO COM A REDE WIRELESS
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(200);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());

  //=====

  timeClient.begin();//INICIALIZA O SERVICO PARA OBTER DATA E HORA
  timeClient.setTimeOffset(-10800); //SETA O FUSO HORARIO DE BELEM
  timeClient.forceUpdate();

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);//FAZ A AUTENTICACAO NO FIREBASE
  pzem.setAddress(ip); //SETA O IP NO MÓDULO DE LEITURA DOS DADOS DA REDE ELETRICA
```

```

}

void loop() {
timeClient.forceUpdate();
  StaticJsonBuffer<500> jsonBuffer;
  JsonObject& dados = jsonBuffer.createObject();

if (WiFi.status() == WL_CONNECTED){
  digitalWrite(ledvermelho, LOW); //apaga o led vermelho
  digitalWrite(ledverde, HIGH); // acende o led verde

} else {
  digitalWrite(ledvermelho, HIGH); //acende o led vermelho
  digitalWrite(ledverde, LOW); //apaga o led verde
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD); // caso esteja desconectado ele tenta conectar automaticamente
  delay(2000);
}

  float consumoReal=0;
  float consumoPeriodo = 0;

//=====Bloco que obtem a data e hora atual da internet=====
  formatarDataHora = timeClient.getFormattedDate();

  int splitT = formatarDataHora.indexOf("T");
  data = formatarDataHora.substring(0, splitT);

  hora = formatarDataHora.substring(splitT+1, formatarDataHora.length()-1);

//Inicio da leitura da rede eletrica

for (int i =0; i<15; i++){
  tensao = pzem.voltage(ip);
  corrente = pzem.current(ip);

  if(tensao<0.0){ tensao = pzem.voltage(ip);
  } else{
  if(tensao<0.0){ tensao = 0;}}

  if(corrente <0.0){corrente = pzem.current(ip);
  }else{
  if(corrente<0.0){corrente=0;}}
  };
  Serial.println(hora);//MOSTRA A HORA NO TERMINAL
  Serial.print("Tensao: "); Serial.println(tensao);
  Serial.print("Corrente: "); Serial.println(corrente);
  consumoReal = (tensao * corrente);//em wats por hora
  consumoPeriodo +=consumoReal*2; //consumo acumulado no intervalo de 30segundo
  Serial.print("Watts/h: "); Serial.println(consumoReal);
  Serial.println(i);
  delay(1000)
}

  dados["data"] = data;
  dados["hora"] = hora;
  dados["tensao"] = tensao;

```

```
dados["corrente"] = corrente;
dados["whora"] = consumoReal;
dados["w30seg"] = consumoPeriodo;

Firebase.push(TABLE_NAME, dados); // Comando para postar no firebase

if (Firebase.failed()) {
    Serial.print( String(TABLE_NAME) + " erro ao enviar");
    digitalWrite(ledvermelho, HIGH); digitalWrite(ledverde, LOW);
    delay(1000);
    Serial.print("Status Firebase: ");
    Serial.println(Firebase.error());
}
}
```