

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETROTÉCNICA  
CURSO DE ENGENHARIA ELÉTRICA

GIOVANI ANDREY MAESTRELLI  
GUILHERME SZEMCZAK NAPOLEÃO

**SISTEMA SUPERVISÓRIO PARA MONITORAMENTO DE ENERGIA  
ELÉTRICA RESIDENCIAL**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2018

GIOVANI ANDREY MAESTRELLI  
GUILHERME SZEMCZAK NAPOLEÃO

**SISTEMA SUPERVISÓRIO PARA MONITORAMENTO DE ENERGIA  
ELÉTRICA RESIDENCIAL**

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do curso de Engenharia Elétrica, do Departamento Acadêmico de Eletrotécnica, da Universidade Tecnológica Federal do Paraná, como requisito parcial à obtenção ao título de Engenheiro Eletricista.

Orientador: Prof. Dr. Roberto Cesar Betini

CURITIBA

2018

Giovani Andrey Maestrelli  
Guilherme Szemczak Napoleão

## SISTEMA SUPERVISÓRIO PARA MONITORAMENTO DE ENERGIA ELÉTRICA RESIDENCIAL

Este Trabalho de Conclusão de Curso de Graduação foi julgado e aprovado como requisito parcial para a obtenção do Título de Engenheiro Eletricista, do curso de Engenharia Elétrica do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR).

Curitiba, 20 de junho de 2018.

---

Prof. Antonio Carlos Pinho, Dr.  
Coordenador de Curso  
Engenharia Elétrica

---

Profa. Annemahlen Gehrke Castagna, Mestre  
Responsável pelos Trabalhos de Conclusão de Curso  
de Engenharia Elétrica do DAELT

### ORIENTAÇÃO

---

Prof. Roberto Cesar Betini, Dr.  
Universidade Tecnológica Federal do Paraná  
Orientador

### BANCA EXAMINADORA

---

Prof. Roberto Cesar Betini, Dr.  
Universidade Tecnológica Federal do Paraná

---

Prof. Guilherme Luiz Moritz, Dr.  
Universidade Tecnológica Federal do Paraná

---

Prof. Celso Fabrício de Melo Júnior, Mestre  
Universidade Tecnológica Federal do Paraná

A folha de aprovação assinada encontra-se na Coordenação do Curso de Engenharia Elétrica

## RESUMO

MAESTRELLI, Giovani Andrey; NAPOLEAO, Guilherme Szemczak. SISTEMA SUPERVISÓRIO PARA MONITORAMENTO DE ENERGIA ELÉTRICA RESIDENCIAL. 94 f. Trabalho de Conclusão de Curso (Graduação) – Curso de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

O objetivo deste trabalho é apresentar o desenvolvimento de um sistema supervisório para monitoramento de energia elétrica residencial. Além disso, analisar a importância do sistema no enquadramento do usuário à tarifa branca. Para isso, utiliza-se um microcontrolador ESP32, o qual por meio de um programa, desenvolvido em linguagem C, possibilita comunicação com sensores de corrente, tensão e internet. Finalmente, desenvolve um software em linguagem de programação PHP para registrar os dados em um banco de dados MySQL, apresentando os resultados obtidos de forma gráfica, de fácil interação com o usuário, possibilitando ao mesmo decidir qual a melhor modalidade tarifária a ser adotada.

**Palavras chave:** Sistema supervisório, PHP, MySQL, ESP32, sensor de corrente.

## ABSTRACT

MAESTRELLI, Giovani Andrey; NAPOLEAO, Guilherme Szemczak. SUPERVISORY SYSTEM FOR THE MONITORING OF RESIDENTIAL ELECTRICAL ENERGY. 94 f. Trabalho de Conclusão de Curso (Graduação) – Curso de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

This paper's main objective is to present the development of a residential electric power monitoring system. Besides that, analyze the system's importance to the user's fitting into the "white tariff". To achieve that, a microcontroller ESP32 is used, which through a program made in C language, allows communication between the microcontroller, the current and voltage sensors and the internet. Finally, it develops a software in PHP programming language to register all data in a MySQL database, presenting obtained results in a visual manner, made easy for the user to understand, making it possible for them to decide the best electric tariff to adhere.

**Keywords:** Supervisory system, PHP, MySQL, ESP32, current sensor.

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 1: Projeção da demanda brasileira de energia elétrica de 2016 a 2026.....   | 11 |
| Figura 2: Volume útil do reservatório da usina hidrelétrica de Ilha Solteira.....  | 12 |
| Figura 3: Diagrama de blocos do sistema proposto.....                              | 15 |
| Figura 4: ESP-WROOM-32.....  | 18 |
| Figura 5: Medidor eletromecânico de indução monofásico.....                        | 22 |
| Figura 6: Medidor E22A.....  | 24 |
| Figura 7: Medidor E650 S4x da marca Landis+Gyr.....                                | 25 |
| Figura 8: Custo residencial da energia elétrica no Brasil.....                     | 26 |
| Figura 9: Comparação de renda e consumo per capita entre países.....               | 29 |
| Figura 10: Portas ESP-WROOM-32.....  | 32 |
| Figura 11: Sensor de corrente não-invasivo.....                                    | 33 |
| Figura 12: Polaridade de entrada e terminais de saída do sensor.....               | 34 |
| Figura 13: conector de saída do sensor.....  | 35 |
| Figura 14: Seleção do ESP32 Dev Module.....  | 37 |
| Figura 15: Sketch aberto na IDE do Arduino.....                                    | 38 |
| Figura 16: Exemplo de seleção de bibliotecas dentro do sketch do Arduino.....      | 39 |
| Figura 17: Interpretação do conteúdo HTML pelo navegador.....                      | 40 |
| Figura 18: Interpretação do conteúdo HTML com PHP pelo navegador.....              | 42 |
| Figura 19: Tela de acesso ao phpMyAdmin.....                                       | 43 |
| Figura 20: Ambiente de administração do phpMyAdmin.....                            | 44 |
| Figura 21: Dado inserindo dentro da tabela do banco de dados.....                  | 44 |
| Figura 22: Demonstração da utilização da linguagem JavaScript.....                 | 45 |
| Figura 23: Exemplo da dinâmica gerada pelo JavaScript.....                         | 45 |
| Figura 24: Exemplo de código AJAX.....   | 46 |
| Figura 25: Exemplo de dinâmica do código AJAX.....                                 | 47 |
| Figura 26: Exemplo de interpretação do código CSS pelo navegador.....              | 47 |
| Figura 27: Esquema do projeto a ser realizado.....                                 | 49 |
| Figura 28: Obtenção da relação de transformação.....                               | 50 |
| Figura 29: Circuito de medição de corrente utilizado.....                          | 51 |
| Figura 30: Circuito de medição de corrente para entrada na porta ADC do ESP32..... | 51 |
| Figura 31: Circuito de medição de tensão para entrada na porta ADC do ESP32.....   | 52 |
| Figura 32: Circuito de multiplexação de sinal de tensão.....                       | 53 |
| Figura 33: Esquematisação do circuito de multiplexação de tensão.....              | 53 |
| Figura 34: Esquematisação de multiplexação de tensão.....                          | 54 |
| Figura 35: Circuito de medição.....  | 54 |
| Figura 36: Calibração da corrente.....   | 56 |
| Figura 37: Calibração de tensão.....   | 56 |
| Figura 38: Listagem de redes obtidas pelo ESP32.....                               | 58 |
| Figura 39: Estrutura do banco de dados.....  | 60 |
| Figura 40: Estrutura da tabela “configuracao” no MySQL.....                        | 60 |
| Figura 41: Estrutura da tabela “consumo” no MySQL.....                             | 61 |

|   |    |
|---|----|
| Figura 42: Estrutura da tabela “usuario” no MySQL.....                                  | 61 |
| Figura 43: Exemplo da dado inserido pelo ESP32 na tabela “consumo” dentro do MySQL..... | 62 |
| Figura 44: Tela de login.....   | 63 |
| Figura 45: Informações da tela inicial.....   | 63 |
| Figura 46: Informações da tela de configuração. ....                                    | 64 |
| Figura 47: Seleção de data a ser visualizada.....                                       | 65 |
| Figura 48: Tela de Alarmes. ....  | 66 |
| Figura 49: Exemplo de email enviado pelo sistema de alarmes.....                        | 67 |
| Figura 50. Protótipo instalado.....   | 68 |
| Figura 51. Medições de corrente.....  | 69 |
| Figura 52. Medições de tensão. ....   | 69 |
| Figura 53. Histórico de consumo em Wh. ....   | 70 |

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 1: Limite de erro de acordo com a classe.....         | 23 |
| Tabela 2: Produção de energia elétrica no Brasil.....        | 30 |
| Tabela 3: Especificações do ESP-WROOM-32.....                | 31 |
| Tabela 4. Comparação entre as linguagens de programação..... | 41 |
| Tabela 5. Valores da tarifa branca e convencional.....       | 71 |
| Tabela 6. Comparação de custos entre as tarifas. ....        | 72 |
| Tabela 7. Relação de custos do protótipo. ....               | 74 |



## LISTA DE SIGLAS

|       |  |
|-------|--|
| ANEEL | Agência Nacional de Energia Elétrica               |
| AJAX  | <i>Asynchronous Javascript and XML</i>             |
| ASP   | <i>Active Server Pages</i>                         |
| CH_PD | <i>Chip Enable</i>                                 |
| CLP   | Controlador Lógico Programável                     |
| COPEL | Companhia Paranaense de Energia                    |
| CSS   | <i>Cascading Style Sheets</i>                      |
| EPE   | Empresa de Pesquisa Energética                     |
| GND   | <i>Ground</i>                                      |
| GPIO  | <i>General Purpose Input/Output</i>                |
| HTML  | <i>HyperText Markup Language</i>                   |
| I2C   | <i>Inter-Integrated Circuit</i>                    |
| I2S   | <i>Inter-IC Sound</i>                              |
| IDE   | <i>Integrated Development Environment</i>          |
| I/O   | <i>Input/Output</i>                                |
| JSP   | <i>JavaServer Pages</i>                            |
| LED   | <i>Light Emitting Diode</i>                        |
| NBR   | Norma Brasileira                                   |
| PHP   | <i>Hypertext Preprocessor</i>                      |
| PWM   | <i>Pulse-Width Modulation</i>                      |
| RDBMS | <i>Relational Database Management System</i>       |
| RST   | <i>Reset</i>                                       |
| RX    | Recepção de dados                                  |
| SCL   | <i>Serial Clock</i>                                |
| SDA   | <i>Serial Data</i>                                 |
| SDIO  | <i>Secure Digital Input Output</i>                 |
| SPI   | <i>Serial Peripheral Interface</i>                 |
| SQL   | <i>Structured Query Language</i>                   |
| TC    | Transformador de corrente                          |
| TTL   | <i>Transistor-Transistor Logic</i>                 |
| TX    | Transmissão de dados                               |
| UART  | <i>Universal Asynchronous Receiver Transmitter</i> |

## SUMÁRIO

|  |           |
|--|-----------|
| <b>1. INTRODUÇÃO .....</b>                                   | <b>11</b> |
| 1.1 TEMA .....   | 11        |
| 1.1.1 DELIMITAÇÃO DO TEMA .....                              | 14        |
| 1.2 PROBLEMAS E PREMISSAS:.....                              | 15        |
| 1.3 OBJETIVOS.....   | 15        |
| 1.3.1 OBJETIVO GERAL.....                                    | 15        |
| 1.3.2 OBJETIVOS ESPECÍFICOS:.....                            | 16        |
| 1.4 JUSTIFICATIVA:.....                                      | 16        |
| 1.5 PROCEDIMENTOS METODOLÓGICOS: .....                       | 17        |
| 1.6 ESTRUTURA DO TRABALHO.....                               | 19        |
| <b>2. FUNDAMENTAÇÃO TEÓRICA.....</b>                         | <b>20</b> |
| 2.1 AUTOMAÇÃO.....   | 20        |
| 2.2 SISTEMA SUPERVISÓRIO .....                               | 21        |
| 2.3 SISTEMAS DE MEDIÇÃO DE CONSUMO ELÉTRICO.....             | 21        |
| 2.3.1 MEDIDORES ELETROMECAÂNICOS .....                       | 22        |
| 2.3.2 MEDIDORES ELETRÔNICOS .....                            | 23        |
| 2.4 SISTEMAS TARIFÁRIOS PARA CONSUMIDORES RESIDENCIAIS ..... | 25        |
| 2.4.1 MODALIDADES.....                                       | 27        |
| 2.4.3 CONSUMO DE ENERGIA ELÉTRICA NO BRASIL.....             | 29        |
| 2.5 A PLACA ELETRÔNICA ESP32.....                            | 30        |
| 2.5.1 HARDWARE .....   | 31        |
| 2.5.2 SENSORES.....  | 33        |
| 2.5.2.1 CORRENTE.....  | 33        |
| 2.5.2.2 TENSÃO .....   | 35        |
| 2.6 SOFTWARE E LINGUAGEM DE PROGRAMAÇÃO .....                | 36        |
| 2.6.1 IDE ARDUINO .....                                      | 36        |
| 2.6.3 PHP .....  | 41        |
| 2.6.4 MySQL.....   | 42        |
| 2.6.7 CSS .....  | 47        |
| 2.7 COMUNICAÇÃO HARDWARE-SOFTWARE .....                      | 48        |
| <b>3. DESENVOLVIMENTO .....</b>                              | <b>50</b> |

|   |           |
|---|-----------|
| 3.1 TESTES INICIAIS .....   | 50        |
| 3.2 MONTAGEM DO HARDWARE .....  | 53        |
| 3.3 PROGRAMAÇÃO DO MICROCONTROLADOR .....                               | 55        |
| 3.3.1 CÓDIGO DE CALIBRAÇÃO .....  | 55        |
| 3.3.2 CONEXÃO <i>WiFi</i> .....   | 58        |
| 3.3.3 ARMAZENAMENTO DA VARIÁVEL NA URL .....                            | 59        |
| 3.5 MANIPULAÇÃO DE DADOS .....  | 59        |
| 3.5.1 ESTRUTURA DO BANCO DE DADOS MYSQL.....                            | 59        |
| 3.5.2 REGISTRO DE DADOS NO BANCO DE DADOS MYSQL .....                   | 61        |
| 3.6 INTERFACE GRÁFICA .....   | 62        |
| 3.6.1 TELA DE LOGIN.....  | 62        |
| 3.6.2 TELA INICIAL .....  | 63        |
| 3.6.3 TELA DE CONFIGURAÇÃO.....   | 64        |
| 3.6.4 TELA DE HISTÓRICO DE CORRENTE E TENSÃO .....                      | 65        |
| 3.6.5 TELA DE ALARMES .....   | 65        |
| <b>4 RESULTADOS E DISCUSSÕES.....</b>                                   | <b>67</b> |
| 4.1 TESTE REALIZADO .....   | 67        |
| 4.2 CONSUMO E CUSTO .....   | 70        |
| 4.2.1 COMPARAÇÃO DA TARIFA BRANCA COM A CONVENCIONAL .....              | 71        |
| 4.3 DIFICULDADES ENCONTRADAS .....                                      | 73        |
| 4.4 ANÁLISE DE CUSTO DO SISTEMA .....                                   | 73        |
| <b>5 CONCLUSÕES .....</b>   | <b>74</b> |
| 5.1 TRABALHOS FUTUROS .....   | 75        |
| 5.2 CONSIDERAÇÕES GERAIS .....  | 76        |
| <b>REFERÊNCIAS.....</b>   | <b>77</b> |
| <b>APÊNDICE A – CÓDIGO CALIBRAÇÃO ESP32.....</b>                        | <b>84</b> |
| <b>APÊNDICE B – CÓDIGO DE PROCURA DE REDES <i>WiFi</i> ESP32.....</b>   | <b>86</b> |
| <b>APÊNDICE C – CÓDIGO DE REQUISIÇÃO HTTP E IP ESTÁTICO ESP32 .....</b> | <b>87</b> |
| <b>APÊNDICE D – CÓDIGO FONTE ESP32.....</b>                             | <b>89</b> |
| <b>APÊNDICE E – ARQUIVO DE CONEXÃO CLASSES.PHP.....</b>                 | <b>93</b> |
| <b>APÊNDICE F – REGISTRO DE DADOS MYSQL BD.PHP .....</b>                | <b>94</b> |

## 1. INTRODUÇÃO

### 1.1 TEMA

O consumo de energia elétrica em escala mundial vem aumentando, no Brasil não é exceção. Estima-se que a demanda brasileira de energia elétrica se expandirá em um ritmo de 3,7% ao ano para a classe residencial e comercial e 2,9% para a classe industrial atingindo cerca de 744 TWh em 2026 (EPE, 2017a).

A Figura 1 mostra a projeção da demanda de energia elétrica de 2016 a 2026, sendo possível observar que houve uma amenização devido a eficiência energética (conservação), observa-se também que demanda de energia elétrica para 2026 será de 744 TWh, sendo 650 TWh supridos pela rede e 94 TWh supridos pela autoprodução, como por exemplo a geração distribuída.

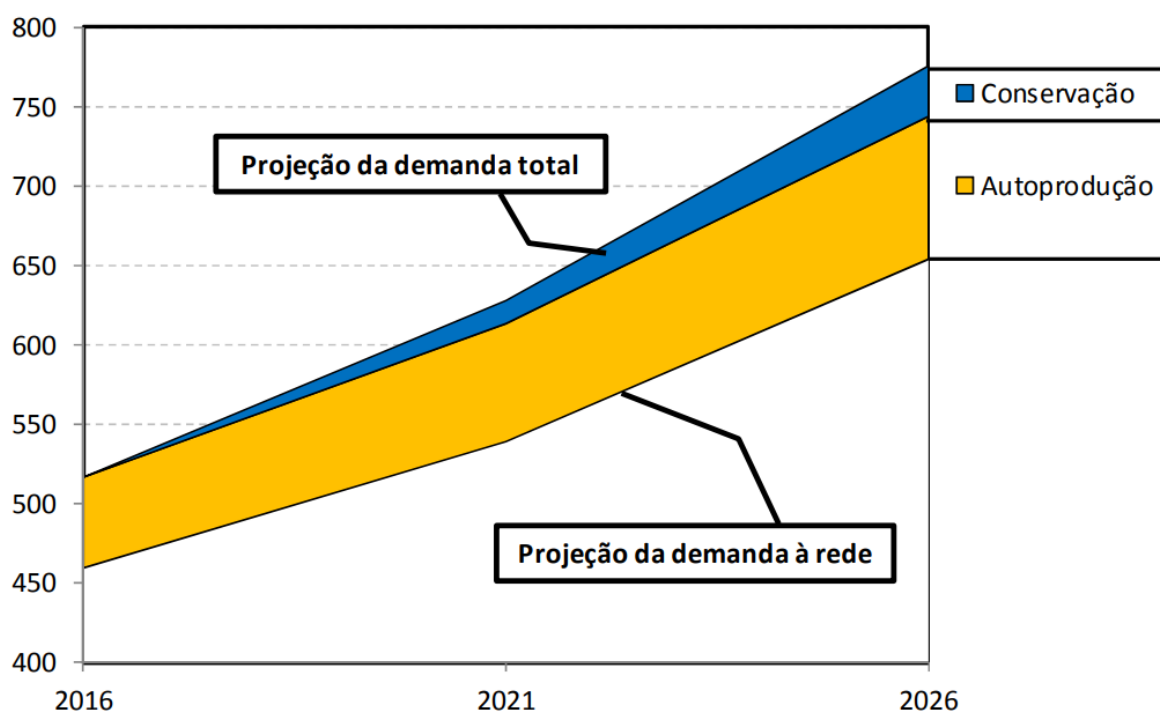


Figura 1: Projeção da demanda brasileira de energia elétrica de 2016 a 2026.

Fonte: EPE (2017a).

O Brasil possui um grande potencial de geração de energia renovável, principalmente para a implementação de usinas hidrelétricas, as quais causam menos impactos ao meio ambiente levando em conta a quantidade de energia gerada.

Atualmente a matriz de energia elétrica brasileira é composta por cerca de 61,08% de origem hídrica, 8,85% de biomassa, 7,17% de origem eólica, 16,44% de origem fóssil, 1,21% de origem nuclear e 0,25% de origem solar (ANEEL, 2017a).

Segundo Galvão e Bermann (2015), o Sistema Elétrico Brasileiro que é altamente dependente de usinas hidrelétricas, sofreu com redução dos níveis de seus reservatórios no ano de 2014, gerando uma crise na segurança energética do país. Para suprir esta demanda, o governo optou pela geração de energia através das usinas termoelétricas, cujo custo de geração é maior comparado com as usinas hidrelétricas.

De acordo com a Figura 2, a porcentagem do volume útil do reservatório da usina hidrelétrica de Ilha Solteira sofreu uma grande diferença comparando os anos de 2013 e 2014, enquanto em 2013 o reservatório mantinha um nível estável, em 2014 foi reduzido drasticamente.

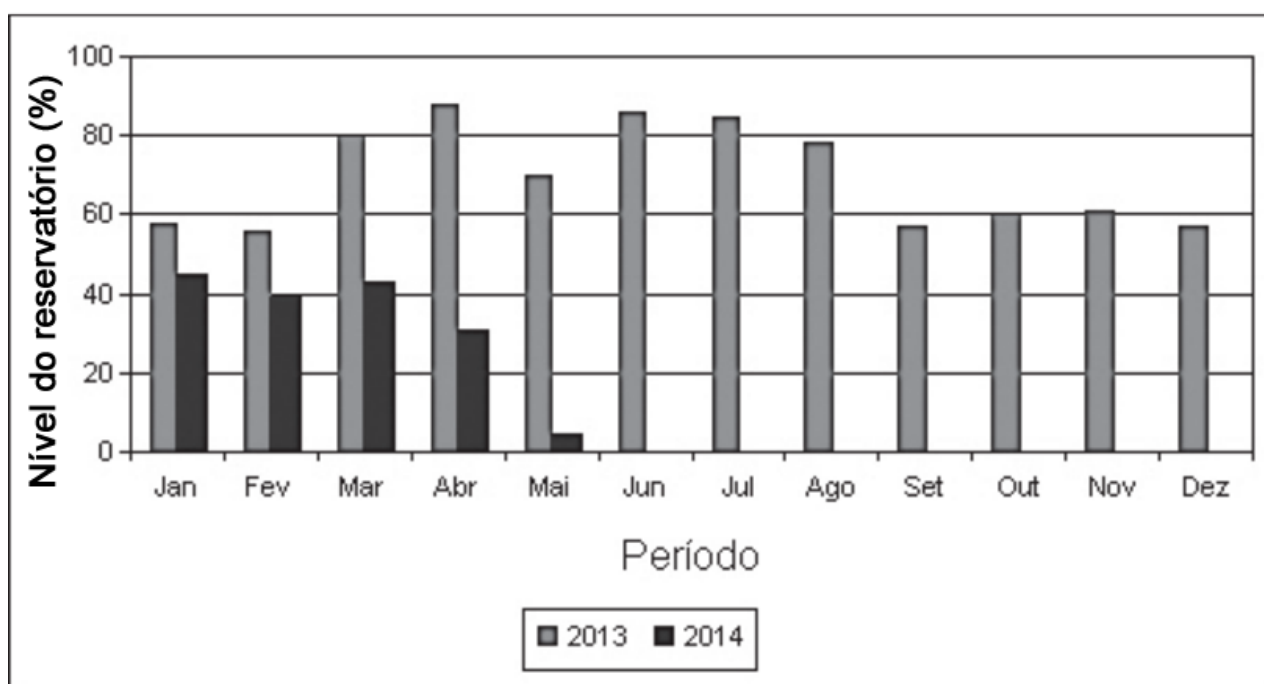


Figura 2: Volume útil do reservatório da usina hidrelétrica de Ilha Solteira.

Fonte: Adaptado de ONS (2015).

As usinas hidrelétricas que correspondem a maior parte da matriz energética brasileira vêm sofrendo com dificuldade de licenciamento ambiental devido a necessidade de alagamento de grandes áreas, sendo assim necessária a construção de usinas a fio d'água que não possuem reservatório. As usinas a fio d'água não geram garantias suficientes ao sistema, sendo necessária a utilização de outras fontes de energia, normalmente optando pela geração térmica (TANCREDI; ABBUD, 2010).

A utilização de termelétricas é indispensável no atual cenário energético brasileiro devido a sua estabilidade, porém o custo de geração é elevado, e traz consigo muitos impactos negativos ao meio ambiente por causa dos resíduos tóxicos gerados, principalmente por aquelas que utilizam óleo diesel e carvão como combustível (LAWSON; PEREIRA, 2017).

Esses combustíveis liberam na atmosfera gases como o dióxido de carbono ( $\text{CO}_2$ ) e o óxido nitroso ( $\text{N}_2\text{O}$ ), gases esses que agravam o efeito estufa, sendo uma das causas do aquecimento global. Apesar dos problemas, o seu uso para produção de energia na matriz energética do país é indispensável nos tempos atuais, criando assim um clima de instabilidade devido ao seu alto custo, que contribui para o aumento do preço dos energéticos, ao mesmo tempo em que a política ambiental exige redução dos níveis de emissões de gases que agravam o efeito estufa (BETINI, 2014).

Pode-se citar o aumento da demanda aliada a falta de planejamento da matriz energética brasileira como principais fatores para o aumento da tarifa de energia elétrica, afetando assim todos os setores da economia brasileira (TANCREDI; ABBUD, 2013). Com a tarifa de energia elétrica mais cara, a solução de curto prazo é uma melhor gestão do consumo de energia elétrica para evitar os desperdícios.

Segundo Goldemberg e Lucon (2007, p. 18), “a eficiência energética é, sem dúvida, a maneira mais efetiva de ao mesmo tempo reduzir os custos e os impactos ambientais locais e globais”. Existem leis e programas que incentivam a eficiência energética, porém a falta de conhecimento da população sobre como é calculado o consumo de energia faz com que continuem utilizando os equipamentos com baixa eficiência energética.

Na área de consumo residencial a gestão do consumo é dificultada pela falta de equipamentos que possibilitam o consumidor ter acessos as estatísticas de

consumo, então a comparação do consumo de energia só é realizada quando o consumidor está com a conta de energia elétrica em mãos, gerando assim muitas dúvidas de como se atingiu tal valor.

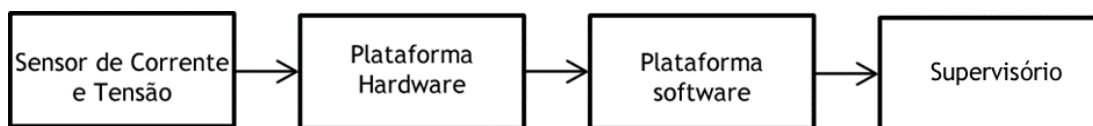
Os equipamentos disponíveis no mercado para fazer o monitoramento de energia no setor residencial permitem apenas o monitoramento de um equipamento, possuem poucas informações e o custo de investimento é alto levando em conta suas funcionalidades, cerca de R\$110,00 em média de acordo com uma breve pesquisa feita pela internet (ML, 2017).

Sendo assim a difusão de tais equipamentos em larga escala se torna inviável e de pouco interesse ao consumidor. Opta-se então por uma alternativa de baixo custo e mais atrativa, que visa auxiliar o consumidor a verificar como está sendo consumida a sua energia elétrica, possibilitando o monitoramento de vários circuitos simultaneamente, fornecendo alarmes de anormalidades, gráficos diários de níveis de tensão, corrente, consumo, assim como comparações de custos em modalidades tarifárias diferentes.

### 1.1.1 DELIMITAÇÃO DO TEMA

Apresentar o desenvolvimento de um sistema supervisor que visa auxiliar o consumidor residencial na utilização de energia elétrica de forma eficiente, através da medição de corrente e tensão de circuitos individuais, identificação do motivo de acionamento de disjuntores, gráficos detalhados de consumo, além do cálculo automático do custo de consumo baseado nas tarifas adotadas na região de implementação do sistema. Através de uma comparação do histórico de dados coletados, será implementado um alarme de anormalidades, que irá informar o usuário a existência de algum problema. O consumidor terá acesso a todos os dados através de qualquer dispositivo que possua conexão com um navegador de internet, como *smartphone*, *notebook* e *tablet*.

A Figura 3 consiste em um diagrama de blocos simplificando as diferentes etapas do processo de funcionamento do sistema supervisor, desde a aquisição de dados até a interface que se comunicará com o usuário.



**Figura 3: Diagrama de blocos do sistema proposto.**

**Fonte: Autoria própria.**

## 1.2 PROBLEMAS E PREMISSAS:

O problema identificado é a falta de conhecimento da população brasileira com relação ao consumo de energia elétrica de suas residências, bem como a dificuldade de obter ferramentas de baixo custo para supervisionar a quantidade de energia elétrica consumida em sua residência.

Existem simuladores que ajudam os consumidores a terem alguma perspectiva de como a energia está sendo consumida em sua residência, porém trata-se de sistemas simples, engessados e de baixa confiabilidade que não levam em conta as diversas variações do consumo causadas por fatores exógenos como variações de consumo em diferentes épocas do ano.

Existe também a dificuldade do consumidor de determinar qual será a melhor modalidade tarifária a ser escolhida devido à inclusão da tarifa branca que proporciona valores diferenciados para determinados períodos do dia. O cronograma da ANEEL de acordo com COPEL (2018b) estabelece que a partir de 2018 consumidores do grupo B que consomem acima de 500 kWh poderão optar pela tarifa branca, em 2019 poderão optar consumidores a partir de 250 kWh e em 2020 não haverá restrição.

## 1.3 OBJETIVOS

### 1.3.1 OBJETIVO GERAL

Desenvolver um sistema supervisório de baixo custo para monitoramento de consumo de energia elétrica residencial para quadros elétricos que possuam até 5



circuitos.

### 1.3.2 OBJETIVOS ESPECÍFICOS:

- Identificar qual tipo de *hardware* será mais viável para o problema proposto.
- Apresentar como os dados serão obtidos pelo *hardware*.
- Definir como será realizada a comunicação *software-hardware*.
- Identificar a estrutura e a linguagem de programação a ser utilizada no *software*.
- Demonstrar como toda a informação enviada ao *software* será processada.
- Realizar a montagem do protótipo.
- Descrever o cálculo do consumo de energia elétrica.
- Identificar a melhor modalidade tarifária a ser adotada.

Gerar relatórios diários de consumo e situação dos circuitos elétricos.

### 1.4 JUSTIFICATIVA:

O estudo e sistema desenvolvidos podem beneficiar a sociedade em aspectos econômicos e ambientais, uma vez que existe a preocupação com o uso de energia elétrica de forma eficiente. Sistemas disponíveis no mercado atualmente possuem um elevado custo de implementação e poucas funcionalidades, não sendo viáveis para uma difusão em larga escala. O estudo em questão propõe uma alternativa de baixo custo e com mais funcionalidades.

O consumidor terá acesso a interface gráfica através de uma página da internet que poderá ser disponibilizada por qualquer dispositivo que possua navegador *web*, o objetivo dessa característica do sistema é facilitar a vida do usuário possibilitando visualizar as informações de qualquer local a qualquer momento.

Com os relatórios de consumo que serão gerados, será possível determinar

qual o melhor tipo de modalidade tarifária a ser adotada, sendo ela a convencional ou branca, pois sem o sistema supervisor é muito difícil de obter dados precisos de consumo em determinados períodos do dia.

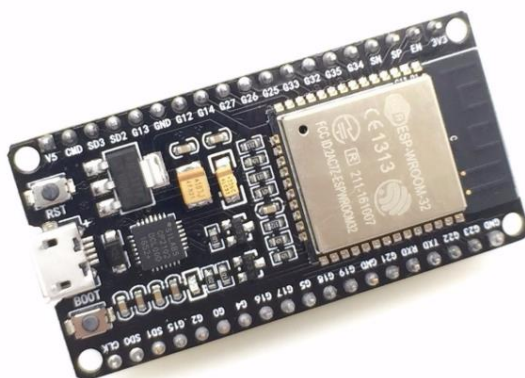
Em casos de atuação de disjuntores, será possível acompanhar o motivo da atuação do mesmo, se foi por sobrecorrente ou curto circuito, possibilitando ao consumidor tomar medidas preventivas ou corretivas quanto a suas instalações.

### 1.5 PROCEDIMENTOS METODOLÓGICOS:

O projeto proposto possui um caráter experimental, por se tratar do desenvolvimento de um protótipo real bem como a medição de grandezas físicas como potência elétrica e tensão. Apesar disso, no desenvolvimento do trabalho, será realizada uma vasta pesquisa bibliográfica através de livros, *datasheets* e pesquisas realizadas anteriormente.

Em uma primeira etapa de desenvolvimento do trabalho, serão realizadas pesquisas para obter-se o *hardware* mais apropriado para o projeto. A princípio, o protótipo será desenvolvido na plataforma ESP32 da empresa Espressif Systems, uma plataforma de prototipagem de hardware livre que possui o microcontrolador ESP32-D0WDQ6 baseado na linguagem de programação C/C++ (ESPRESSIF, 2018b), pois esse possui facilidade de acesso a componentes necessários para a realização deste projeto e também a referências bibliográficas. A aquisição dos dados deverá ser realizada através de sensores conectados a suas portas ADC, podendo os mesmos serem visualizados através de comunicação serial, sendo possível assim a aferição mais rápida dos dados pretendidos.

A Figura 4 mostra a plataforma ESP-WROOM-32, esta plataforma se destaca por possuir módulo *WiFi* integrado, tamanho reduzido, alto poder de processamento e podendo ser aplicada a mesma metodologia da programação das placas Arduino (ESPRESSIF, 2018c).



**Figura 4: ESP-WROOM-32.**

**Fonte: ESPRESSIF (2018a).**

O protótipo deverá ser calibrado através de comparações de dados obtidos pelo mesmo com dados obtidos por meio de equipamentos de medição usuais de alta confiabilidade. Após a calibração do protótipo, o mesmo deverá ser instalado em um quadro elétrico residencial, dando início a conexão com o *software* para que os dados monitorados sejam manipulados posteriormente.

Com o *hardware* pronto para uso, será desenvolvida a plataforma de *software*, que terá a sua conexão com o ESP32 estabelecida através do módulo *WiFi* integrado, enviando requisições via protocolo IEEE 802.11 e registrando as informações em um banco de dados para posteriormente serem utilizadas na manipulação.

As linguagens de programação que serão utilizadas na plataforma de *software* são linguagens de alto nível e de código aberto, evitando custos adicionais com aquisição de licenças, colaborando para a proposta de um protótipo de baixo custo. A plataforma será então hospedada em um servidor para que o acesso aos dados seja feito de qualquer local através da internet. Serão gerados relatórios de consumo, gráficos, monitoramento instantâneo e alarmes, que podem ser acessados através de qualquer equipamento que possua conexão a um navegador de internet, como *smartphones*, *notebooks* e *tablets*.

Em um último momento, serão avaliados os resultados do projeto, se as expectativas foram alcançadas bem como possíveis problemas encontrados durante o processo de desenvolvimento do protótipo.

## 1.6 ESTRUTURA DO TRABALHO

De forma a estruturar o desenvolvimento do projeto, o trabalho será dividido em 5 partes: Introdução, fundamentação teórica, desenvolvimento, discussão de resultados e conclusão.

**Introdução:** Serão feitas as considerações iniciais, apresentará o tema proposto, bem como o problema que levou ao seu desenvolvimento. Também serão discutidos os objetivos e procedimentos metodológicos.

**Fundamentação teórica:** Apresentará o embasamento teórico necessário para o desenvolvimento do sistema. Síntese de diferentes estudos como curvas de carga, consumo residencial de energia elétrica no Brasil, tipos de medidores eletrônico, modalidades tarifárias, informações sobre a placa eletrônica ESP-WROOM-32, sensores, módulos, linguagens de programação e o projeto do protótipo a ser realizado.

**Desenvolvimento:** Projeto e desenvolvimento do sistema supervisor, apresentará como os dados coletados pelos sensores serão tratados e enviados ao usuário de forma gráfica. Detalhamento do *software a ser* desenvolvido e arquitetura de *hardware* que será utilizada.

**Resultados e discussões:** Utilizando o protótipo desenvolvido, serão realizados testes e simulações do sistema, bem como a discussão dos resultados obtidos e dificuldades encontradas durante o seu desenvolvimento.

**Conclusão:** Comentários pertinentes ao término do projeto. Será avaliado se o objetivo foi atingido, sugestões e possíveis melhorias para trabalhos futuros.

## 2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo contém uma breve introdução à conceitos e tecnologias importantes para o desenvolvimento do protótipo. Ao longo do capítulo serão apresentados os principais conceitos dentro do âmbito de automação residencial, sistemas supervisórios, sistemas de medição de energia elétrica e sistemas tarifários. Além disso, explica-se como é a comunicação do ESP32 com os transdutores, assim como o desenvolvimento do software, a linguagem de programação.

### 2.1 AUTOMAÇÃO

O termo “automação” é polissêmico, ou seja, possui diferentes significados, porém, diferentes definições variam apenas em detalhes. Pode-se entender automação pela “substituição do trabalho do homem, manual, por sistemas previamente programados que se auto-controlam, regulam e realizam uma série de operações em velocidade superior à capacidade humana” (NEGREIROS; DIAS, 2007).

A automação não teve um ponto de origem específico, levando em consideração que o termo pode incluir qualquer instrumento criado para auxiliar o ser humano em suas tarefas do dia-a-dia, como por exemplo, o uso da roda d’água na automatização do processo de moagem, serrarias, ferrarias e trituração de grãos em geral (TEZA, 2002).

Tecnologias de automação podem ser empregadas em diversos ambientes, como residências, indústrias ou organizações, como é o caso de bancos. Para cada ambiente a automação possui características e tecnologias distintas. Para a automação residencial, por exemplo, o foco está em segurança, comunicação, gestão energética e conforto de uma habitação (MURATORI; HENRIQUE, 2011).

O sistema supervisório a ser desenvolvido se encaixa na área de gestão energética de uma residência, e tem por finalidade oferecer ao usuário informação o suficiente para que este tenha maior controle sobre a energia utilizada em sua residência, bem como auxiliar na escolha de tarifação, entre a tarifa convencional ou branca, como será discutido no item 2.4.1 na página 27.

## 2.2 SISTEMA SUPERVISÓRIO

Os sistemas de automação atuais atingiram tal nível que apenas a intuição humana não é mais suficiente para construir modelos bem definidos dos mesmos. Nestas circunstâncias, um ambiente de modelagem torna-se necessário para que se alcance esse objetivo. O software reconhecido como supervisório permite a operação e visualização através de telas gráficas elaboradas para qualquer processo residencial, industrial ou comercial, independentemente do tamanho do projeto (JURIZATO; PEREIRA, 2003).

Um sistema supervisório possui dois fatores principais, os algoritmos de controle e telas gráficas de supervisão. As telas gráficas são as telas visualizadas pelo homem, que possuem informações essenciais para a operação do sistema, bem como botões para realizar certas funções. Nos algoritmos de controle, são associados eventos a cada objeto. Como por exemplo, o acionamento de um botão em que uma das telas gráficas informa ao sistema, seja ele um CLP ou um microcontrolador, que queremos que determinada função ou elemento do campo seja ativado (JURIZATO; PEREIRA, 2003). Os algoritmos de controle são todos os dados não visualizados pelo operador das telas, e são essenciais para o funcionamento do sistema.

Os sistemas supervisórios vêm ganhando espaço no mercado, sendo desenvolvido para áreas que antigamente eram inviáveis, como é o caso de residências, fato que se deve principalmente a um fator financeiro.

Atualmente existem no mercado diversos softwares voltados para o desenvolvimento de sistemas supervisórios, com o intuito de auxiliar o projetista na criação de sistemas com diversas telas e dados na forma gráfica, como é o caso do software Elipse E3 (ELIPSE, 2018).

## 2.3 SISTEMAS DE MEDIÇÃO DE CONSUMO ELÉTRICO

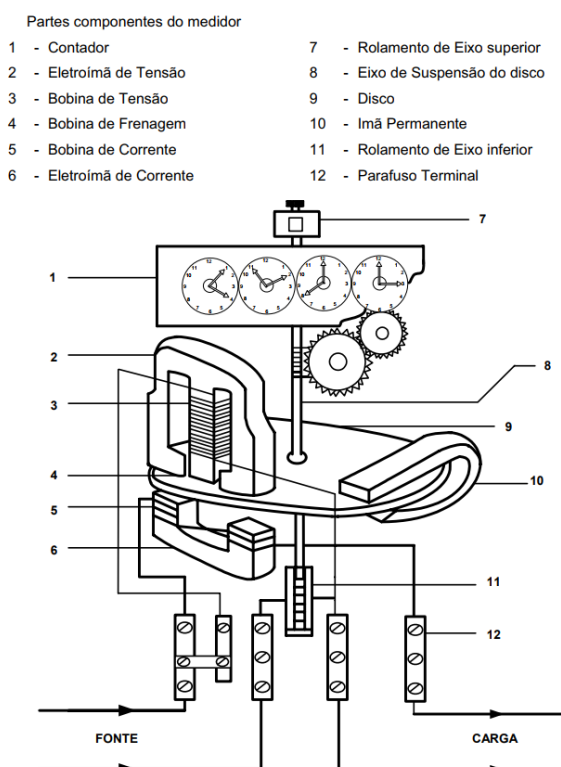
Para fazer a medição do consumo de energia elétrica em uma residência, indústria ou centro comercial, são utilizados instrumentos de medição que podem ser eletromecânicos e/ou eletrônicos, tais dispositivos apresentam o consumo de energia elétrica em um valor de quilowatt hora (kWh), isto é, a potência consumida em um

determinado tempo.

Dependendo da carga, medidores de consumo elétrico podem ser instalados diretamente na linha do consumidor, porém, para cargas mais elevadas, é necessário o uso de um transformador de corrente, para diminuir o valor da corrente no mesmo, por motivos de segurança. Neste item, serão apresentados o funcionamento de medidores eletromecânicos e eletrônicos, bem como suas principais características.

### 2.3.1 MEDIDORES ELETROMECAÑICOS

Os medidores eletromecânicos tiveram origem no final do século 19, e como o próprio nome sugere, são medidores de consumo de energia que possuem tanto uma parte elétrica, quanto mecânica. Os componentes de um medidor eletromecânico se encontram na Figura 5.



**Figura 5: Medidor eletromecânico de indução monofásico.**

**Fonte: VELASCO (2007).**

O medidor de indução monofásico é baseado princípio de Ferraris: “Um condutor percorrido por uma corrente elétrica, na presença de um campo magnético externo, fica submetido a uma força.”, através da corrente passando pelas bobinas, é

induzida uma corrente no disco de alumínio, denominada de corrente de Foucault, a interação entre o disco e o campo magnético resultante faz com que o mesmo gire, e através da velocidade de rotação do disco pode-se determinar a quantidade de energia elétrica consumida (VELASCO, 2007).

No Brasil, os medidores são regulamentados através da norma NBR 8377, 1995, tal norma especifica o tipo de medidor, materiais, valores de tensão e corrente bem como o erro admissível. De acordo com a norma, no Brasil os medidores de consumo devem se encaixar em uma classe de erro, que variam de acordo com a Tabela 1.

**Tabela 1: Limite de erro de acordo com a classe.**

| Classe de exatidão | Limite de erro |
|--------------------|----------------|
| 2                  | ±2%            |
| 1                  | ±1%            |
| 0,5                | ±0,5           |

**Fonte: NBR 8377 (1995).**

Os medidores eletromecânicos no Brasil se encaixam na classe 2, isto é, admitem um erro máximo de 2%. É possível encontrar medidores eletromecânicos de indução de classe 1 e até mesmo 0,5, porém o valor desses medidores é elevado pois não são fabricados no Brasil.

### 2.3.2 MEDIDORES ELETRÔNICOS

Medidores eletrônicos acrescentam novas funcionalidades ao medidor eletromecânico, primeiramente, o consumo de energia elétrica é mostrado através de um display de LCD ou LED, além disso, podem-se medir valores de potência instantânea, ativa ou reativa utilizada em certo período do dia, como horários de pico. Além de mostrar o valor de consumo de potência através do *display*, informações do medidor podem ser transmitidas para outros aparelhos eletrônicos de forma remota.

Os medidores eletrônicos utilizam circuitos eletrônicos para fazer a medição.



Cada modelo possui funcionalidades diferentes, para satisfazer diferentes consumidores. A Figura 6 mostra medidor eletrônico E22A da marca Landis+Gyr (2017a), muito utilizado para consumidores residenciais, pois foca apenas na energia ativa.



**Figura 6: Medidor E22A**

**Fonte: Landis+Gyr (2017).**

A Figura 7 mostra o medidor E650 S4x da mesma marca, possui maior capacidade de memória RAM, faz medições em quatro quadrantes, apresenta valores de potência ativa, reativa e aparente recebida e fornecida através de dois métodos diferentes de cálculo, para obter uma maior precisão no valor. Este medidor é empregado em indústrias, e possui um valor de mercado mais elevado.



**Figura 7: Medidor E650 S4x da marca Landis+Gyr.**

**Fonte: Landis+Gyr (2017).**

O manual de especificações técnicas para medidores eletrônicos de energia da COPEL (2015) apresenta informações detalhadas de construção e testes a serem realizados nos medidores fornecidos por fabricantes. Pode-se destacar o item 4.2.4 “pode ser solicitada ao fornecedor a disponibilização de 20 medidores, de sua linha de produção, para a realização de testes em campo. Estes não devem apresentar qualquer falha ou defeito em um período de 3 meses de avaliação”, o que mostra a rigorosidade dos testes e importância do medidor para uma empresa que fornece energia elétrica.

## 2.4 SISTEMAS TARIFÁRIOS PARA CONSUMIDORES RESIDENCIAIS

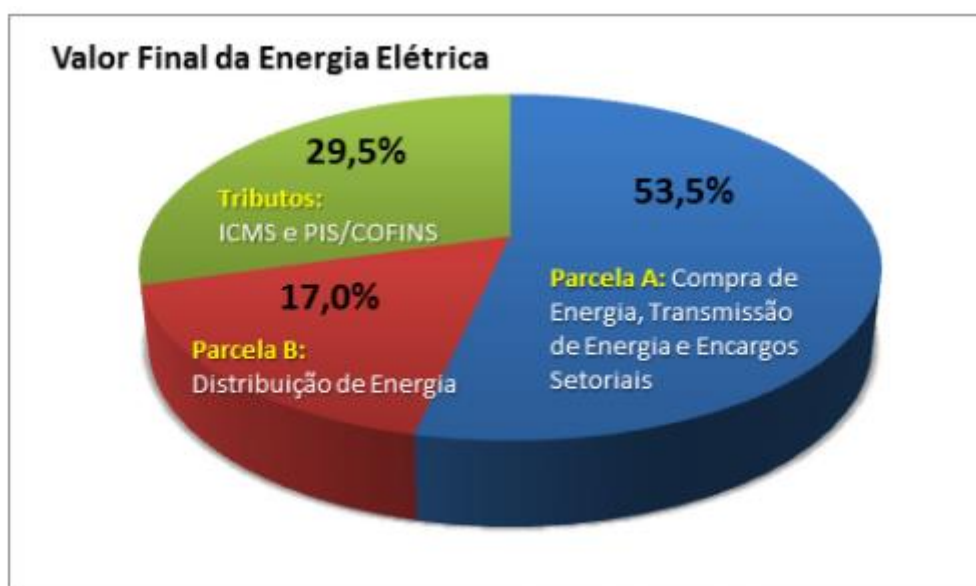
O protótipo a ser desenvolvido neste trabalho, não apenas fornecerá o valor de potência ativa consumida em uma residência, como também o preço a ser pago a concessionária pelo seu consumo. Para realizar esta conversão de maneira correta, é necessário um maior entendimento da cobrança de energia elétrica no país.

De acordo com a ANEEL (2017b): “A tarifa visa assegurar aos prestadores dos serviços receita suficiente para cobrir custos operacionais eficientes e remunerar investimentos necessários para expandir a capacidade e garantir o atendimento com qualidade. Os custos e investimentos repassados às tarifas são calculados pelo órgão

regulador, e podem ser maiores ou menores do que os custos praticados pelas empresas”.

Para que a qualidade do fornecimento de energia elétrica seja garantida, a distribuidora considera três custos que devem ser avaliados no cálculo da tarifa, são eles a energia gerada, transmissão/distribuição e encargos setoriais.

Para fins de cálculos tarifários, a composição dos custos da distribuidora se divide em dois tipos, A: Compra de energia, transmissão e encargos setoriais; e B: Distribuição de energia. Atualmente os custos para o consumidor se dão na proporção mostrada na Figura 8. Observa-se que além dos dois custos citados anteriormente, A e B, aproximadamente 30% do valor final da tarifa de energia consiste em impostos.



**Figura 8: Custo residencial da energia elétrica no Brasil.**

**Fonte: ANEEL (2017b).**

A partir do ano de 2015, as tarifas brasileiras passaram a contar com o sistema de bandeiras tarifárias, tal sistema consiste em aumentar ou diminuir o preço da energia elétrica no país de acordo com a condição de geração de energia no país em um dado momento. Por exemplo, em um mês com poucas chuvas em uma região, em que a geração de energia por hidrelétricas é prejudicada, é necessário a utilização mais intensa de termelétricas, tal ação acaba por encarecer a produção de energia elétrica, logo será necessária a aplicação de uma bandeira tarifaria apropriada para

que o consumidor pague a diferença.

Como certificado pela ANEEL (2017b), as bandeiras são:

Bandeira verde: condições favoráveis de geração de energia. A tarifa não sofre nenhum acréscimo;

Bandeira amarela: condições de geração menos favoráveis. A tarifa sofre acréscimo de R\$ 0,01 para cada quilowatt-hora (kWh) consumidos;

Bandeira vermelha - Patamar 1: condições mais custosas de geração. A tarifa sofre acréscimo de R\$ 0,03 para cada quilowatt-hora (kWh) consumido.

Bandeira vermelha - Patamar 2: condições ainda mais custosas de geração, estado mais crítico. A tarifa sofre acréscimo de R\$ 0,05 para cada quilowatt-hora (kWh) consumido.

Estes valores de bandeiras são os valores cobrados atualmente no país. Até outubro de 2017, os valores cobrados eram de R\$ 0,02 para a bandeira amarela, R\$ 0,03 para o patamar 1 da bandeira vermelha e R\$ 0,035 para o patamar 2 da bandeira vermelha. De acordo com a ANEEL, as mudanças foram feitas pois os valores antigos não eram suficientes para pagar pelo custo extra da energia com o uso mais intenso de termelétricas.

#### 2.4.1 MODALIDADES

Para entenderem-se as modalidades tarifárias aplicadas no Brasil, deve-se ter conhecimento dos grupos de consumidores. Qualquer consumidor de energia elétrica no Brasil entra em um de dois grupos, A ou B. Cada grupo possui seus subgrupos, com o objetivo de separar mais a fundo os diferentes tipos de consumidores. De acordo com a COPEL, os grupos de consumidores na região são:

Grupo A, unidades consumidoras com fornecimento em tensão igual ou superior a 2,3 kV:

- A1 – tensão de fornecimento igual ou superior a 230 kV;

- A2 – tensão de fornecimento de 88 kV a 138 kV;
- A3 – tensão de fornecimento de 69 kV;
- A3a – tensão de fornecimento de 30 kV a 44 kV;
- A4 – tensão de fornecimento de 2,3 kV a 25 kV;
- AS – tensão de fornecimento inferior a 2,3 kV, a partir de sistema subterrâneo de distribuição.

Grupo B, consumidores com fornecimento em tensão inferior a 2,3 kV:

- B-1 - Residencial;
- B-1- Residencial Baixa Renda;
- B-2 - Rural;
- B-2 - Cooperativa de utilização rural;
- B-2 - Serviço público de irrigação;
- B-3 - Demais classes;
- B-4 - Iluminação Pública.

A partir dos diferentes grupos de consumidores, aplicam-se diferentes tarifas, chamadas de modalidades tarifárias, de acordo com a ANEEL (2017b), são elas:

Azul: aplicada às unidades consumidoras do grupo A, caracterizada por tarifas diferenciadas de consumo de energia elétrica e de demanda de potência, de acordo com as horas de utilização do dia;

Verde: modalidade tarifária horária verde: aplicada às unidades consumidoras do grupo A, caracterizada por tarifas diferenciadas de consumo de energia elétrica, de acordo com as horas de utilização do dia, assim como de uma única tarifa de demanda de potência;

Convencional Binômia: aplicada às unidades consumidoras do grupo A caracterizada por tarifas de consumo de energia elétrica e demanda de potência, independentemente das horas de utilização do dia. Esta modalidade será extinta a partir da revisão tarifária da distribuidora;

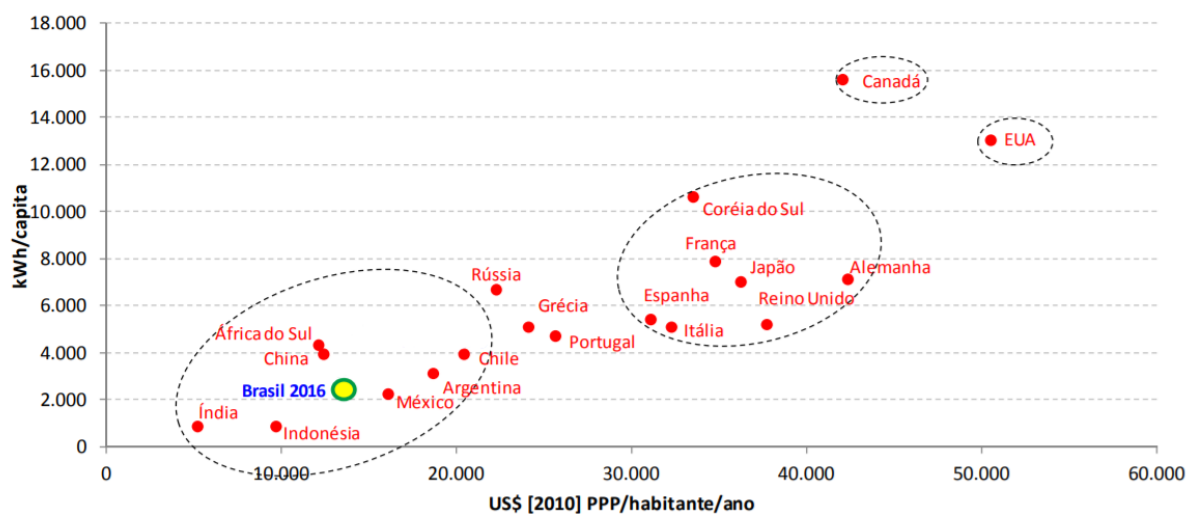
Convencional Monômia: aplicada às unidades consumidoras do grupo B, caracterizada por tarifas de consumo de energia elétrica, independentemente das horas de utilização do dia;

Branca: aplicada às unidades consumidoras do grupo B, exceto para o subgrupo B4 e para as subclasses Baixa Renda do subgrupo B1, caracterizada por tarifas diferenciadas de consumo de energia elétrica, de acordo com as horas de utilização do dia.

Para a Copel, o horário de pico, horário em que a demanda de energia é mais elevada, é entre as 18h e 21h (19h às 22h em horário de verão), com exceção de sábados, domingos e feriados definidos por lei federal.

### 2.4.3 CONSUMO DE ENERGIA ELÉTRICA NO BRASIL

O consumo de energia elétrica é um dos maiores indicadores de desenvolvimento econômico de um país, fato que pode ser visualizado ao comparar a variação da renda per capita com o consumo de energia entre países, como mostrado na Figura 9.



**Figura 9: Comparação de renda e consumo per capita entre países.**

**Fonte: Adaptado de EPE (2017a).**

De acordo com a EPE (EPE, 2017b), no ano de 2016 foram produzidos no Brasil 578.898 GWh, correspondendo a uma diminuição de 0,4% em comparação com o ano anterior (2015), deste valor, a geração hidroelétrica foi responsável pela produção de 380.911 GWh (65,8% do valor total). A Tabela 2 contém a composição da produção de energia elétrica no Brasil em GWh, através de dados levantados pela EPE nos anos de 2015 e 2016.

**Tabela 2: Produção de energia elétrica no Brasil.**

|                             | 2015           | 2016           | Part. % (2016) |
|-----------------------------|----------------|----------------|----------------|
| <b>Total (GWh)</b>          | <b>581.227</b> | <b>578.898</b> | <b>100</b>     |
| Hidráulica (GWh)            | 359.743        | 380.911        | 65,8           |
| Gás Natural (GWh)           | 79.490         | 56.485         | 9,8            |
| Derivados de Petróleo (GWh) | 25.657         | 12.103         | 2,1            |
| Carvão (GWh)                | 18.856         | 17.001         | 2,9            |
| Nuclear (GWh)               | 14.734         | 15.864         | 2,7            |
| Biomassa (GWh)              | 47.394         | 49.236         | 8,5            |
| Eólica (GWh)                | 21.626         | 33.489         | 5,8            |
| Outras (GWh)                | 13.728         | 13.809         | 2,4            |

**Fonte: Adaptado de EPE (2017b).**

Um ponto importante a ser destacado é a produção de energia elétrica por fontes eólicas, responsável apenas por 5,8% da produção de energia elétrica total do país. Porém este valor corresponde a um aumento de 54,9% se comparado com o ano de 2015.

A maior parte da energia elétrica total produzida no país é consumida na região sudeste, responsável pelo consumo de 49,9% da energia do país no ano de 2016. A região sul se encontra na segunda posição, com um consumo de 17,8% do total.

Com estes dados e levando em consideração a relação entre o consumo de energia e a participação no PIB de um país, pode-se concluir que o Paraná tem participação significativa no PIB do Brasil, estando atrás apenas da região sudeste.

## 2.5 A PLACA ELETRÔNICA ESP32

Segundo Vagapov (2017) o ESP32 é uma plataforma de hardware livre ideal para a criação de dispositivos que permitam interação com o ambiente utilizando sensores como entrada, e recomendado para aplicações na internet das coisas, por possuir módulo *WiFi* integrado e tamanho reduzido. A plataforma utiliza-se de uma camada simples de software implementada na placa que utiliza a linguagem C/C++, a qual é open source também. Por ser uma plataforma de hardware livre existem

diversos fabricantes desta e os mesmos utilizam nomes diferentes para suas placas, porém todas elas possuem a mesma finalidade.

### 2.5.1 HARDWARE

A placa ESP32 a ser utilizada no projeto será a versão ESP-WROOM-32. Esta placa utiliza o microcontrolador ESP32-D0WDQ6 de 12 bits e dois núcleos (*dual core*), de acordo com a documentação (ESPRESSIF, 2018c) as especificações estão na Tabela 3 a seguir:

**Tabela 3: Especificações do ESP-WROOM-32.**

| ESP-WROOM-32                       |               |
|------------------------------------|---------------|
| Tensão de Operação                 | 2,7 V / 3,6 V |
| Tensão de Operação (Recomendada)   | 3,3 V         |
| Memória interna ROM                | 448 KB        |
| Memória interna SRAM               | 520 KB        |
| Frequência de <i>clock</i> interno | 40 Mhz        |
| Corrente de operação (média)       | 80 mA         |

**Fonte: Adaptado de ESPRESSIF (2018c).**

#### 2.5.1.1 PORTAS

A documentação do ESP32-WROOM-32 (ESPRESSIF, 2018c) informa que há no total 32 portas digitais, das quais 16 podem ser utilizadas como saída PWM de 12 bits. Cada porta digital opera em uma tensão de 0V para nível lógico baixo ou 3,3V para nível lógico alto, podendo fornecer uma corrente de 80mA em condições normais de operação.

A placa possui 18 entradas analógicas para conversão digital, as quais fornecem uma resolução de 12 bits na escada de 0 a 3,3V. Estas entradas serão utilizadas para a aquisição dos dados dos sensores que vem em um formato analógico, convertendo-os para uma escala digital entre 0 e 4095 (12 bits) para



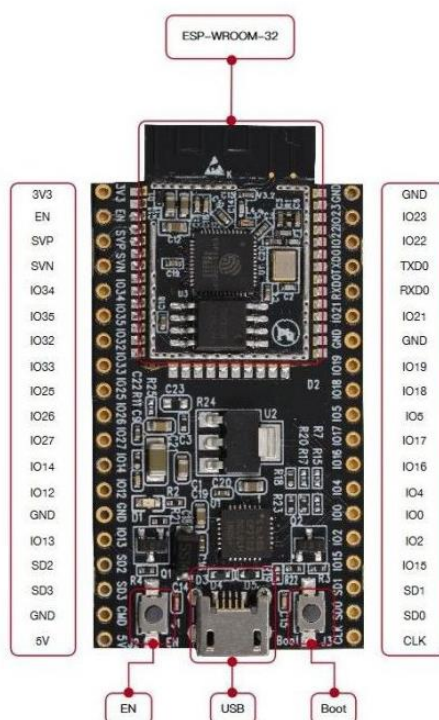
posteriormente serem manipulados no software (ESPRESSIF, 2018b).

De acordo com a documentação do ESP-WROOM-32 (ESPRESSIF, 2018c) os pinos de tensão existentes da placa são:

- 5V: Tensão de saída com 5V fornecida pela placa.
- 3,3V: Tensão de saída com 3,3V fornecida pela placa com corrente máxima de 80 mA.
- GND: Pino conhecido como terra, que realiza fechamento do circuito.

A placa ainda possui dez pinos GPIO com suporte a toque capacitivo, dois canais conversores digital-analógico (DAC) de 8 bits, duas portas com interface I<sup>2</sup>C, muito utilizadas para módulos externos a placa, duas portas com interface UART, e duas portas com interface I<sup>2</sup>S, utilizadas para conectar componentes de áudio digital (ESPRESSIF, 2018c).

A Figura 10 mostra uma visão esquematizada do ESP-WROOM-32:



**Figura 10: Portas ESP-WROOM-32.**

Fonte: KAMAMI (2017).

## 2.5.2 SENSORES

Os sensores são responsáveis pela obtenção de valores de entrada no ESP32, com a aquisição dos valores é possível convertê-los para um formato digital em uma resolução de 12 bits através das portas analógicas e posteriormente manipular os mesmos dentro do software.

### 2.5.2.1 CORRENTE

Segundo Shajahan e Anand (2013), existem 2 tipos de sensores de corrente, os invasivos e os não-invasivos. No método invasivo o sensor é conectado diretamente a alimentação. No método não-invasivo a conexão direta a alimentação é evitada.

O sensor de corrente que será utilizado no projeto é do tipo não invasivo, o qual se adapta melhor ao projeto devido a sua facilidade de instalação. Seu método de funcionamento é como um alicate amperímetro onde o núcleo envolve um fio que conduz a corrente elétrica. A Figura 11 ilustra o sensor YHDC SCT-013 a ser utilizado, o mesmo tem capacidade de monitorar correntes entre 0A e 100A.



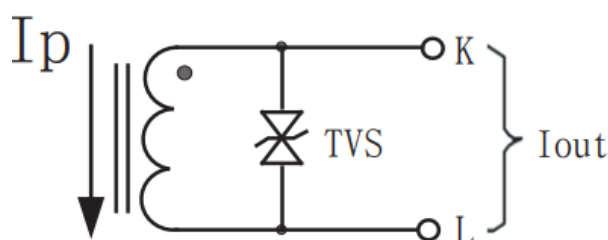
**Figura 11: Sensor de corrente não-invasivo.**

**Fonte: SHAJAHAN E ANAND (2013).**

Segundo Ramos e Andrade (2015), o sensor permite a leitura da corrente operando como um transformador de corrente, o qual o primário está conectado à rede elétrica e o secundário está conectado a um resistor que é utilizado para se obter a tensão com a corrente de saída.

Segatto e Coury (2006, p. 425) definem que a função do transformador de corrente (TC) é “transformar as correntes oriundas do sistema de potência em valores de baixa magnitude e fornecer isolamento galvânica entre a rede elétrica e os relés e outros instrumentos conectados ao enrolamento secundário do TC”.

Segundo o datasheet do componente (YHDC, 2015) a proposta do sensor de corrente é para medição, monitoramento e proteção de equipamentos. A Figura 12 ilustra o esquema de funcionamento do sensor definindo a polaridade de entrada por onde passa a corrente primária e os terminais de saída K e L que deverão ser ligados a um resistor para obter a tensão a ser ligada na porta analógica do ESP32.

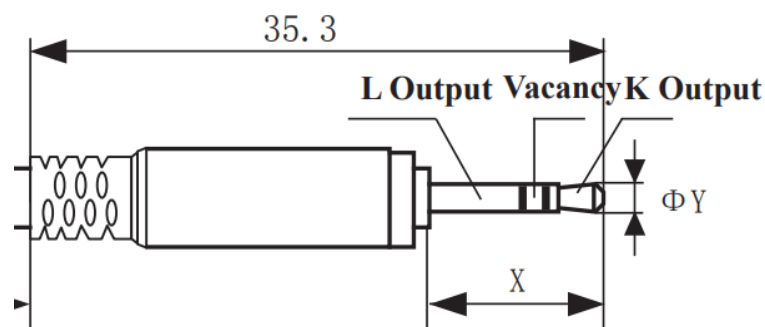


**Figura 12: Polaridade de entrada e terminais de saída do sensor.**

**Fonte: YHDC (2015).**

Há sensores da própria YHDC que utilizam resistor integrado e ao invés de ter uma corrente de saída, possuem tensão de saída.

A Figura 13 ilustra o conector de saída do sensor que é do tipo P2, amplamente utilizado em sistemas de áudio, onde possui os terminais de saída L e K identificados na imagem anterior.



**Figura 13: conector de saída do sensor.**

**Fonte: YHDC (2015).**

Algumas informações do fabricante estão listadas abaixo (YHDC, 2017):

- Corrente de entrada: 0 à 100A;
- Corrente de saída: 0 à 50mA;
- Tipo de saída: corrente ou tensão (resistência à amostra incorporada em forma de saída de tensão);
- Material do núcleo: ferrite;
- Força mecânica: os tempos de abertura e de fechamento do componente não podem ultrapassar 1000 vezes;
- Índice de segurança: rigidez dielétrica 3000V / 1min (entre o invólucro e a saída);
- Temperatura de trabalho: -25 ° C à + 70 ° C.

Este sensor de corrente não invasivo é para o projeto, pois não há contato direto com a rede elétrica evitando possíveis danos à placa eletrônica ESP32, é de fácil instalação e possui uma boa precisão.

### 2.5.2.2 TENSÃO

Para realizar a medição da tensão será utilizado um divisor de tensão com resistores, cuja tensão de saída seja próxima aos valores máximos de leitura que o ESP32 suporta que é de 3,3V.

O sensor de tensão do projeto tem como objetivo identificar se há tensão após o disjuntor ou não para determinar se houve acionamento do mesmo, também é responsável pela medição das fases de entrada do quadro.

Os valores de tensão serão calculados por um algoritmo que irá adquirir amostras de tensão e fazer a integração dos mesmos retornando o valor RMS.

## 2.6 SOFTWARE E LINGUAGEM DE PROGRAMAÇÃO

Esta seção irá explicar com mais detalhes como será feito o desenvolvimento do software e quais linguagens de programação serão utilizadas.

### 2.6.1 IDE ARDUINO

Para programar o ESP32, pode-se utilizar o IDE (Ambiente de Desenvolvimento Integrado) do Arduino, um software livre no qual se programa na linguagem C. O IDE é instalado em um PC com plataforma Windows ou Linux, ele permite que você escreva instruções para controlar as portas do ESP32, das quais você faz o *upload* para o ESP32. O ESP32 executará essas instruções, interagindo com o que estiver conectado a ele. O software do Arduino é de fonte aberta, o que significa que pode ser utilizado livremente por qualquer pessoa e com qualquer propósito (MCROBERTS, 2011).

O IDE do Arduino foi desenvolvido especificamente para trabalhar com placas eletrônicas Arduino. Para que o IDE possa ser utilizado para a programação do ESP32, são necessárias algumas mudanças. Primeiramente, deve-se fazer o download dos arquivos necessários, chamados de *Arduino Core* para o ESP32 (GITHUB, 2018). Em seguida, os arquivos devem ser incluídos na pasta *hardware*, no diretório do Arduino. Finalmente, o arquivo “*get.exe*” deve ser executado, habilitando a opção *ESP32 Dev Module* no IDE do Arduino, como mostrado na Figura 14, podendo este agora ser utilizado para a programação da placa ESP32.

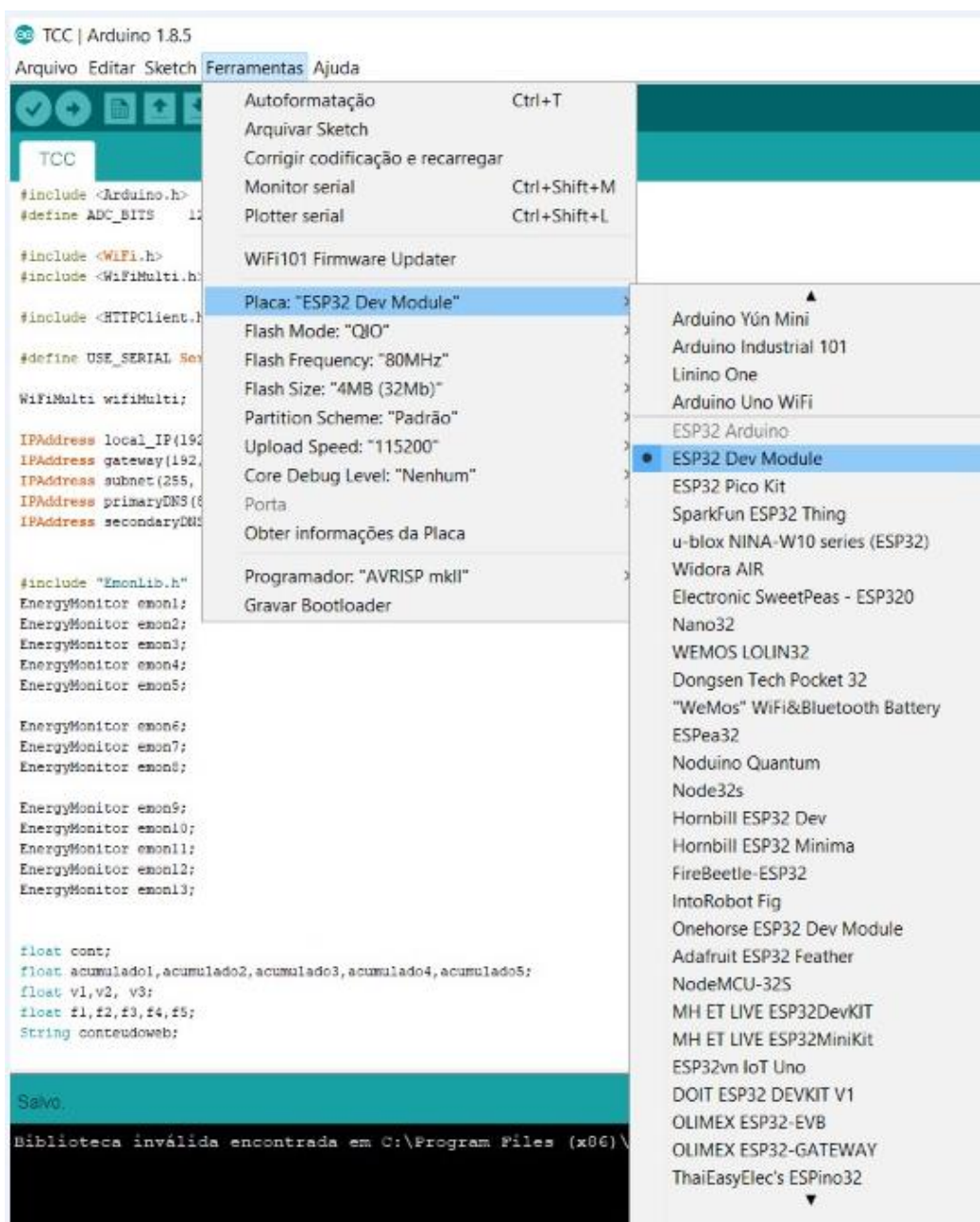
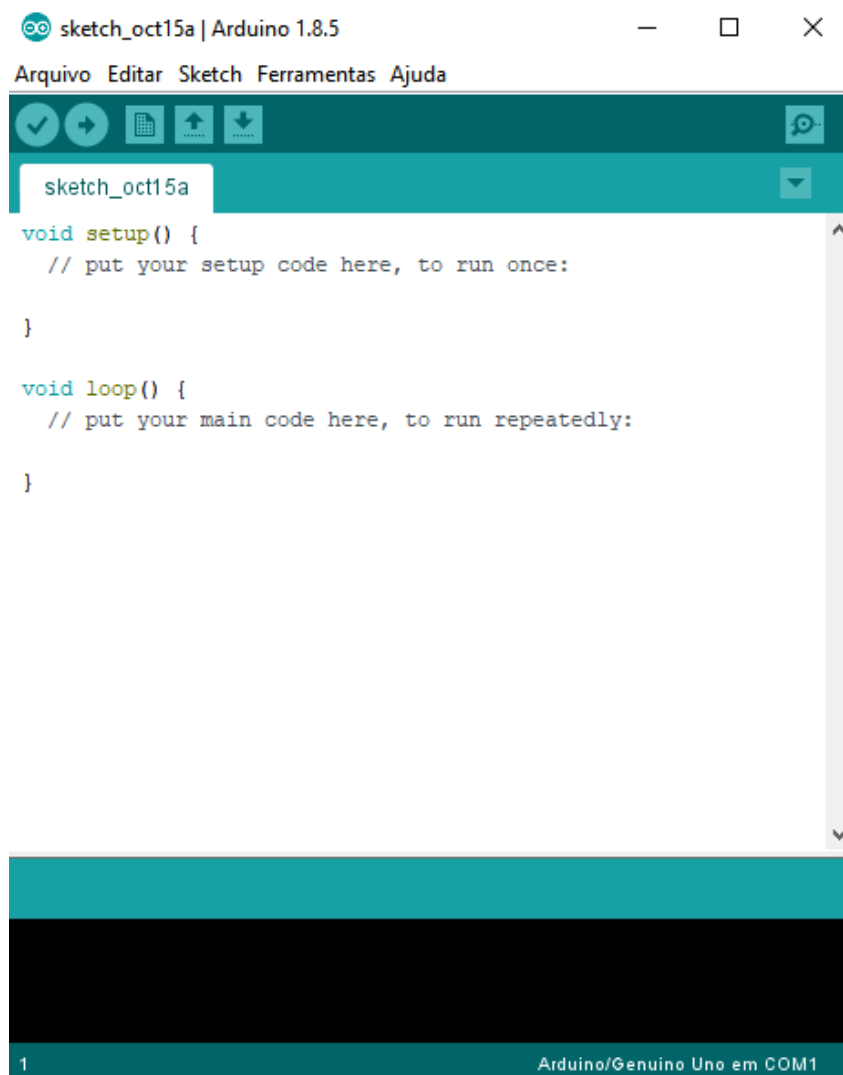


Figura 14: Seleção do ESP32 Dev Module.

Fonte: Autoria própria

O IDE do Arduino é útil para este trabalho em sua fase inicial, pois pode tanto programar o microcontrolador quanto verificar as informações registradas no mesmo de forma rápida e eficiente, facilitando o processo de verificação da correta execução do programa e correção de erros caso seja necessário. A Figura 15 mostra a janela principal da IDE do Arduino que é chamada de *sketch*.



**Figura 15: Sketch aberto na IDE do Arduino.**

**Fonte: Autoria própria.**

A Figura 15 possui uma janela de edição, a qual será incluída a programação. Basicamente nesta janela de edição há duas funções principais, a função *setup* e a função *loop*, que funcionam da seguinte forma:

*setup()* : É chamada quando o sketch inicia. Ela é utilizada para iniciar variáveis, modo de funcionamento dos pinos, incluir livrarias, etc (ARDUINO, 2017b). Esta função não possui um laço de repetição, então será executada somente uma vez quando a placa foi ligada ou reiniciada.

*loop()* : Realiza um laço de repetições permitindo o programa se adaptar as

modificações de seu ambiente, como por exemplo a variação de dados de um sensor, e responder a essas modificações (ARDUINO, 2017a).

O IDE do Arduino proporciona a utilização de bibliotecas, como é mostrado na Figura 16, fornecendo funcionalidades adicionais para o sketch (ARDUINO, 2017c). É necessário acessar o menu *Sketch*, depois o item Incluir Biblioteca e selecionar a biblioteca que esteja de acordo com as necessidades do projeto.

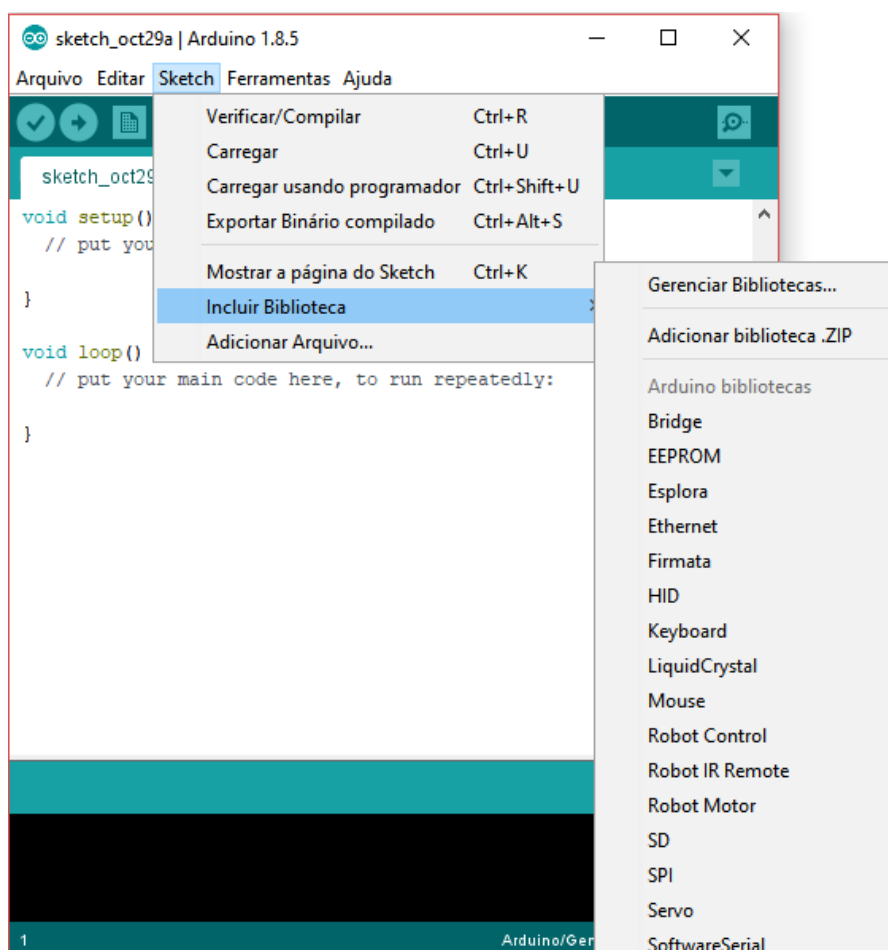


Figura 16: Exemplo de seleção de bibliotecas dentro do sketch do Arduino.

Fonte: Autoria própria.

A IDE já possui algumas bibliotecas, mas existe a possibilidade de criar uma própria ou utilizar de uma fonte externa.



## 2.6.2 HTML

O HTML (*Hyper Text Markup Language*) é a linguagem padrão para criação de páginas na web. Os elementos HTML são representados por *tags* as quais definem o "cabeçalho", "parágrafo", "tabela". O navegador não mostra estas tags, mas utiliza as mesmas para formatar o conteúdo da página web (W3SCHOOLS, 2017a).

Abaixo estão listadas as funcionalidades de algumas tags HTML:

- `<!DOCTYPE html>`: Define o tipo do document.
- `<html>`: É o elemento raiz da pagina HTML.
- `<head>`: É o elemento onde são definida as informações de cabeçalho.
- `<title>` : É o elemento onde é definido o título da página web.
- `<body>` : Dentro desta tag será definido o que estará visível na pagina web.
- `<h1>`: Elemento que define títulos e subtítulos dentro do conteúdo visível da página.
- `<p>`: Elemento que define um parágrafo dentro do conteúdo visível da página.

A Figura 17 contém um exemplo de como as tags são utilizadas; à esquerda temos o código em HTML e no lado direito temos a interpretação feita pelo navegador.



Figura 17: Interpretação do conteúdo HTML pelo navegador.

Fonte: W3SCHOOLS (2017a)

### 2.6.3 PHP

PHP é uma linguagem de programação de ampla utilização para desenvolvimento para a web e pode ser mesclada dentro do código HTML. O objetivo principal da linguagem é permitir a desenvolvedores escreverem páginas que serão geradas dinamicamente e com alta velocidade de resposta (PHP, 2017).

Uma das grandes vantagens do PHP é que ele é gratuito. Outra característica importante é que o PHP é um software com código-fonte aberto. O código-fonte do PHP e sua documentação detalhada também estão disponíveis no site oficial. (NIEDERAUER, 2004).

O ASP.NET, PHP e JSP são as principais linguagens utilizadas para desenvolvimento na web (YU;YI, 2010). A Tabela 4 mostra a comparação de algumas linguagens utilizadas na web.

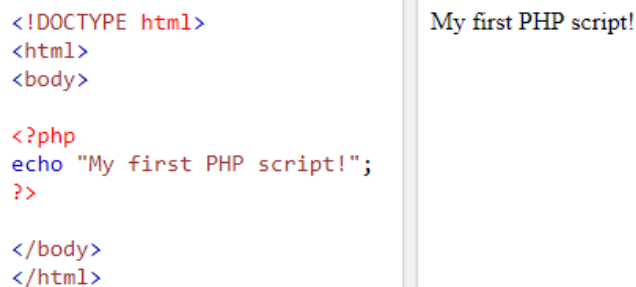
**Tabela 4. Comparação entre as linguagens de programação.**

| Item                             | ASP.NET  | PHP  | JSP                             |
|----------------------------------|--|--|---------------------------------|
| Segurança                        | Boa segurança, porém existem algumas vulnerabilidades. | PHP é reconhecida por sua performance de segurança | Possui a segurança mais elevada |
| Incompatibilidade de plataformas | Única plataforma                                       | Multiplataforma                                    | Multiplataforma                 |
| Eficiência de operação           | Alta   | Muito Alta   | Extremamente Alta               |
| Custo                            | Alto   | Livre  | Alto                            |

**Fonte: Adaptado de (YU;YI, 2010).**

De acordo com a Tabela 4 é possível concluir que o custo-benefício da utilização da linguagem de programação PHP é o melhor entre todas as linguagens comparadas

A Figura 18 retirada do site W3Schools mostra um exemplo do código à esquerda e como ele é apresentado no servidor web à direita.



```
<!DOCTYPE html>
<html>
<body>

<?php
echo "My first PHP script!";
?>

</body>
</html>
```

My first PHP script!

**Figura 18: Interpretação do conteúdo HTML com PHP pelo navegador.**

**Fonte: Adaptado de (W3SCHOOLS, 2017c).**

Diversos bancos de dados são suportados pelo PHP como MySQL, PostgreSQL, Sybase, Oracle, SQL Server e muitos outros (NIEDERAUER, 2004). O banco de dados escolhido para este projeto será o MySQL, seu funcionamento será detalhado no próximo item.

#### 2.6.4 MySQL

O SQL permite acessar e manipular bancos de dados, possui padrão ANSI (W3SCHOOLS, 2017b). Algumas funcionalidades do SQL são listadas abaixo:

- Executar consultas contra um banco de dados.
- Recuperar dados de um banco de dados.
- Inserir registros em um banco de dados.
- Atualizar registros em um banco de dados.
- Excluir registros de um banco de dados.
- Criar novos bancos de dados.
- Criar novas tabelas numa base de dados.
- Criar procedimentos armazenados em um banco de dados.
- Criar visualizações em um banco de dados.
- Redefinir permissões em tabelas, procedimentos e visualizações.

Para construir um site que mostra dados de um banco de dados é necessário

um programa de banco de dados RDBMS (*Relational Database Management System*) como o MySQL, que pode utilizar uma linguagem de programação cliente-servidor como o PHP (W3SCHOOLS, 2017b).

Os principais comandos utilizados no ambiente SQL são SELECT, UPDATE, DELETE, INSERT, WHERE (W3SCHOOLS, 2017b).

O MySQL é o banco de dados de código aberto mais conhecido no mundo. Com comprovado desempenho, confiabilidade e facilidade de uso, o MySQL tornou-se a principal opção de banco de dados para aplicativos baseados na Web, usado por propriedades da Web de alto perfil. (ORACLE, 2017)

Existem vários programas que auxiliam o acesso a base de dados MySQL. O que será utilizado neste projeto é o phpMyAdmin. O phpMyAdmin é um software livre escrito em PHP, ele realiza a administração do MySQL através de uma página web. A Figura 19 mostra a tela de acesso ao phpMyAdmin.



A imagem mostra a interface de login do phpMyAdmin. No topo, há o logotipo do phpMyAdmin, que consiste em um ícone de um veleiro e o texto 'phpMyAdmin'. Abaixo do logotipo, o texto 'Bem-vindo ao phpMyAdmin' é exibido. A interface contém um formulário com os seguintes elementos:

- Um menu suspenso rotulado 'Linguagem - Language' com a opção 'Português - Brazilian portuguese' selecionada.
- Um botão rotulado 'Entrar' com um ícone de olho.
- Dois campos de entrada rotulados 'Usuário:' e 'Senha:'.
- Um botão rotulado 'Executar' na parte inferior direita do formulário.

**Figura 19: Tela de acesso ao phpMyAdmin**

**Fonte: Autoria própria.**

A Figura 20 mostra o ambiente de administração do phpMyAdmin:

| #                          | Nome   | Tipo    | Colaço            | Atributos | Nulo | Padrão                     | Extra          |
|----------------------------|--|---------|-------------------|-----------|------|----------------------------|----------------|
| <input type="checkbox"/> 1 | id  | int(11) |                   |           | Não  | Nenhum wrap (padrão: none) | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | usuario  | text    | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |                |
| <input type="checkbox"/> 3 | senha  | text    | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |                |
| <input type="checkbox"/> 4 | habilitado   | int(11) |                   |           | Não  | Nenhum wrap (padrão: none) |                |
| <input type="checkbox"/> 5 | desativado   | int(11) |                   |           | Não  | Nenhum wrap (padrão: none) |                |

**Figura 20: Ambiente de administração do phpMyAdmin.**

**Fonte: Autoria própria.**

Cada linha representa um índice diferente na tabela, sendo que cada um deles podem possuir características diferenciadas como é mostrado abaixo.

**Nome:** É utilizado como índice da tabela. Estes índices representam o conteúdo que será inserido dentro de cada um deles e na programação PHP os mesmos serão utilizados como chaves para chamar o conteúdo.

**Tipo:** Define o tipo de conteúdo inserido, se ele é inteiro, texto, data, etc.

**Primaria:** Função que geralmente é atribuída ao índice ID, pois é uma função que evita duplicação de valores, ou seja, nunca existira dois ID com valores iguais.

**Extra:** Geralmente é atribuída a coluna ID, pois cada inserção nova no banco de dados gera um número automático em ordem progressiva, sem necessidade de determinar o valor da linha.

| id | usuario   | senha | habilitado | desativado |
|----|-----------|-------|------------|------------|
| 2  | guilherme | teste | 1          | 0          |

**Figura 21: Dado inserindo dentro da tabela do banco de dados.**

**Fonte: Autoria própria.**

A Figura 21 contém um exemplo de dados inseridos no banco, observa-se que a linha superior contém os índices definidos, e abaixo de cada um deles estão as informações pertinentes.

## 2.6.5 JAVASCRIPT

JavaScript é uma linguagem leve, interpretada e baseada em objetos com funções de primeira classe, mais conhecida como a linguagem de script para páginas Web. O JavaScript pode ser aplicado em um documento HTML e ser usado para criar interações dinâmicas em sites (MOZILLA, 2017a).

O JavaScript possui diversas ferramentas, entre elas as APIs (Application Programming Interfaces) disponibilizada em navegadores web, possuem várias funções como criar HTML e atribuir estilos de CSS, gerar gráficos, entre outros. (MOZILLA, 2017b).

A Figura 22 mostra como a linguagem javascript é inserida dentro do HTML.



**Figura 22: Demonstração da utilização da linguagem JavaScript.**

Fonte: W3SCHOOLS, 2017d

Após clicar no botão, o conteúdo HTML com a data do sistema será carregado dentro do item com “id=demo”, pois a linguagem javascript atribuiu através do comando “document.getElementById(‘demo’).innerHTML = Date()”. A Figura 23 mostra como o conteúdo foi carregado.

## My First JavaScript

Click me to display Date and Time.

Tue Oct 24 2017 11:41:51 GMT-0200 (Horário brasileiro de verão)

**Figura 23: Exemplo da dinâmica gerada pelo JavaScript.**

Fonte: W3SCHOOLS, 2017d

### 2.6.6 AJAX

O AJAX (Asynchronous JavaScript and XML) utiliza o objeto XMLHttpRequest para se comunicar com os scripts do lado do servidor. Ele pode enviar bem como receber informações em uma variedade de formatos. A característica intrínseca do AJAX é poder fazer tudo isso sem a necessidade de atualizar a página, permitindo atualizar partes de uma página com base em eventos do usuário. (MOZILLA, 2017c).

Essa possibilidade de não ser necessário redirecionar páginas para obter conteúdo proporciona maior fluidez ao ambiente do software, possibilitando um menor tempo de resposta do servidor. A Figura 24 mostra um exemplo de utilização do AJAX, diferente do exemplo do JavaScript, ele consegue carregar um arquivo para dentro de um campo com um “id” determinado por ele, no caso da Figura 24 o campo é “id=demo” que irá carregar o arquivo “ajax\_info.txt”.



**Figura 24: Exemplo de código AJAX.**

**Fonte: W3SCHOOLS (2017e).**

O resultado após clicar no botão é mostrado na Figura 25, todo o conteúdo que estava dentro da tag com “id=demo” foi substituído pelo conteúdo dentro do arquivo “ajax\_info.txt”, este arquivo poderia ser de uma extensão .php, .html, entre outras.

# AJAX

AJAX is not a programming language.

AJAX is a technique for accessing web servers from a web page.

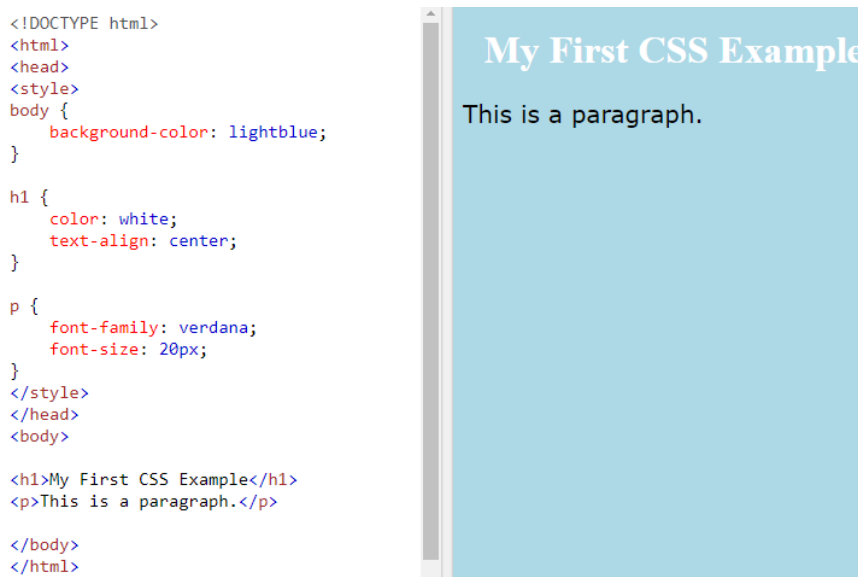
AJAX stands for Asynchronous JavaScript And XML.

**Figura 25: Exemplo de dinâmica do código AJAX.**

**Fonte: W3SCHOOLS (2017e).**

## 2.6.7 CSS

CSS (Cascading Style Sheets) é uma linguagem de estilo usada para descrever a apresentação de um documento escrito em HTML. O CSS descreve como elementos são mostrados na tela (MOZILLA, 2017d). A linguagem CSS é fundamental para a formatação visual do software, na Figura 26 é possível observar um simples exemplo de como o CSS é utilizado e como fica sua visualização.



**Figura 26: Exemplo de interpretação do código CSS pelo navegador.**

**Fonte: W3SCHOOLS, (2017f).**



O código CSS é incluído dentro da tag “style”, na Figura 26 foi definido a cor de fundo, tamanho da fonte e parágrafo. Com a utilização do CSS a visualização de tabelas, textos e efeito visuais ficam mais agradáveis ao usuário, fazendo com que a experiência de utilização do software seja melhorada.

## 2.7 COMUNICAÇÃO HARDWARE-SOFTWARE

A programação do ESP32 será feita dentro do IDE do Arduino. Além do código de programação usual, será necessária a utilização de bibliotecas externas para o sensor de corrente, tensão e conexão com o servidor PHP.

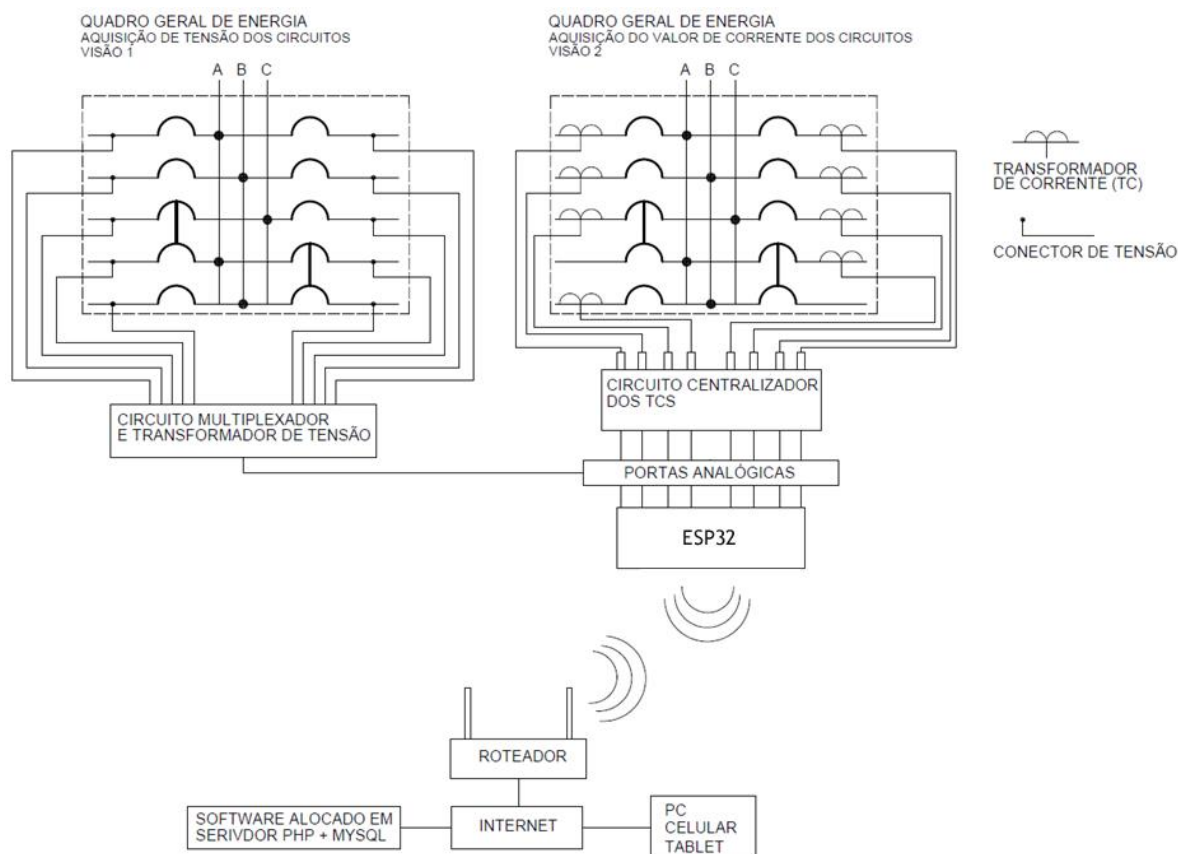
O ESP32 será responsável por adquirir os dados de corrente e tensão através de suas entradas analógicas convertendo os mesmos para uma escala de 12 bits. A programação dentro do ESP32 enviará os dados, através da conexão *WiFi*, para um servidor externo que registrará dentro do banco de dados MySQL que será implementado.

O software externo será responsável pela manipulação dos dados, sendo o mesmo hospedado em um servidor a parte isentando o ESP32 de tarefas que exijam muito de seu processador. O software utilizará os dados contidos dentro do banco de dados MySQL e em conjunto com um arquivo PHP fará a demonstração dos resultados.

A utilização das linguagens CSS, Javascript e AJAX serão necessárias para dar fluidez ao software e fornecer uma apresentação organizadas dos dados obtidos.

O software pode ser acessado por qualquer dispositivo que possua navegador e conexão com internet.

A Figura 27 mostra o esquema de comunicação entre *hardware* e *software*:



**Figura 27: Esquema do projeto a ser realizado.**

**Fonte: Autoria própria**

Os sinais de tensão serão multiplexados através de transistores para que utilizem somente uma porta ADC, estes sinais terão como finalidade identificar a possível atuação dos disjuntores e níveis de tensão. A multiplexação será feita através do acionamento da base dos transistores pelas portas digitais do ESP32 e a saída será conectada a uma porta analógica do mesmo.

Os sensores de corrente serão organizados dentro de uma placa com conectores entradas P2 e as saídas serão conectadas as entradas analógicas do ESP32. Estes valores serão convertidos para uma escala digital e serão utilizados para o cálculo de consumo de energia.

Os dados processados pelo ESP32 serão enviados a cada 20 segundos, via módulo *Wi-Fi*, para um banco de dados *MySQL*. Estes dados serão processados no *software* alocado no mesmo servidor e estarão disponíveis para acesso através de um computador, *Smartphone* ou *Tablet*.

### 3. DESENVOLVIMENTO

Neste capítulo serão abordados os procedimentos realizados para obter o protótipo final. Utilizando os sensores e componentes mostrados no capítulo anterior. Também são tratados aspectos importantes da programação em linguagem C utilizada para programar a placa eletrônica ESP32 e fazer com que esta envie os dados coletados para o banco de dados. Finalmente, como estes dados são tratados para que o usuário final tenha a informação da forma mais clara o possível.

#### 3.1 TESTES INICIAIS

Em um primeiro momento, os testes foram realizados com a utilização de um *protoboard*, apesar de não ser a plataforma adequada, por possuir conexões fracas e sofrer muito com a influência de ruídos, o *protoboard* é ideal para realizar os primeiros testes, pela facilidade de aplicar mudanças na topologia.

Este circuito foi utilizado para testar os transformadores de corrente e sua relação de transformação. O TC tem como finalidade transformar o valor da corrente real em um valor reduzido, que não danifique a placa eletrônica. Esta corrente reduzida passa por um resistor, onde gera uma tensão. A relação de transformação a ser encontrada é a relação entre a corrente real, neste projeto medida por um amperímetro alicate, e o valor de tensão medida no resistor. A Figura 28 mostra como foi realizado um destes testes para obtenção da relação de transformação. Observe-se que a corrente de 7,05A equivale a uma tensão de 0,367V no resistor.

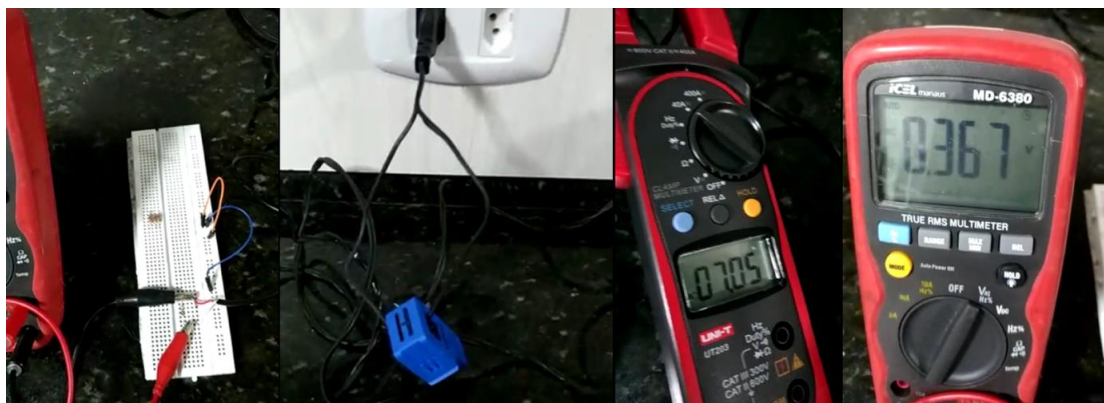
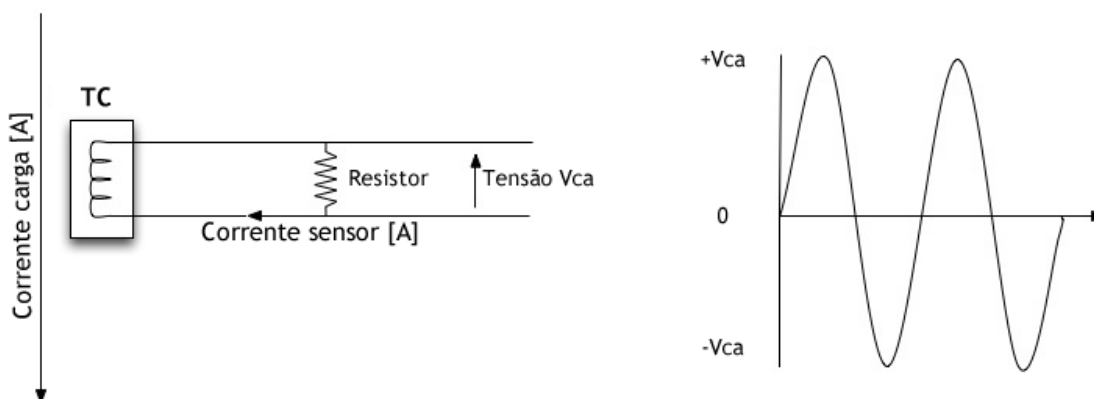


Figura 28: Obtenção da relação de transformação.

Fonte: Autoria própria

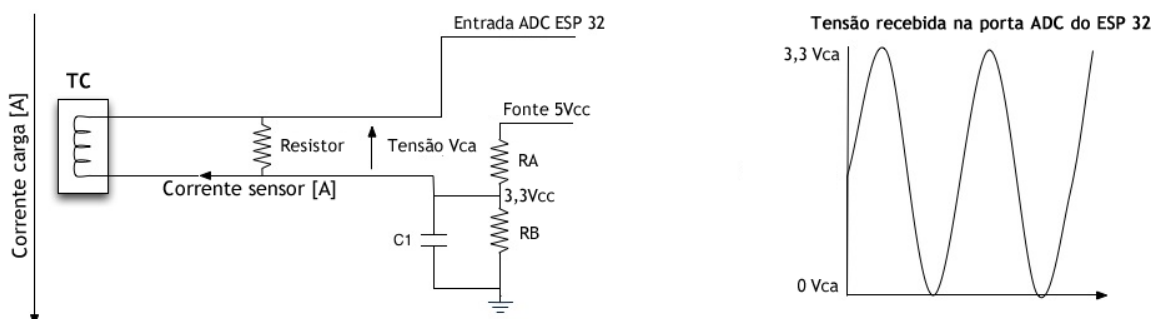
A Figura 29 mostra o circuito de medição de corrente em detalhes.



**Figura 29: Circuito de medição de corrente utilizado.**

Fonte: Adaptado de HOMAUTOMATION (2018).

A tensão medida é alternada e contém valores negativos, porém a entrada ADC do ESP32 não é capaz de fazer a leitura dos valores negativos. Para resolver este problema, se faz necessária a adição de uma tensão contínua  $V_{CC}$  de valor 1,65V, para que a parte negativa de  $V_{CA}$  se torne positiva. A Figura 30 mostra com detalhes como é feita esta modificação.

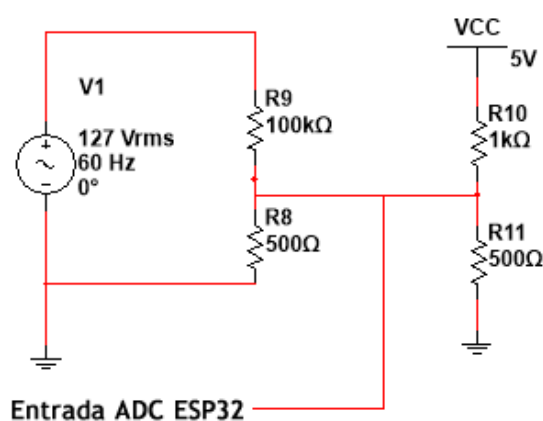


**Figura 30: Circuito de medição de corrente para entrada na porta ADC do ESP32.**

Fonte: Adaptado de HOMAUTOMATION (2018).

Com o circuito da Figura 30 foram feitos testes ajustando o resistor que está em paralelo com o TC para que a tensão de entrada no ESP32 não ultrapassasse 3,3Vp, o que poderia danificar o microcontrolador.

Outro teste realizado foi da aquisição de tensão através de divisor de tensão, dispensando a utilização de transformadores para a leitura da mesma. A metodologia de inserção de uma Vcc para que não haja leitura negativa de tensão segue a mesma lógica que a medição de corrente. A Figura 31 mostra como foi feito o circuito de leitura de tensão da rede.



**Figura 31: Circuito de medição de tensão para entrada na porta ADC do ESP32.**

**Fonte: Autoria própria.**

Devido a limitação de números de portas ADC, se faz necessário realizar testes com um sistema de multiplexação, onde as portas digitais acionam bases de transistores, fazendo com que passe somente um sinal por vez, dessa forma é necessário somente da utilização de 1 porta ADC, ficando somente restrito ao número de portas digitais disponíveis no ESP32 que irão acionar os transistores. Figura 32 explicita como foi feito multiplexação dos sinais.

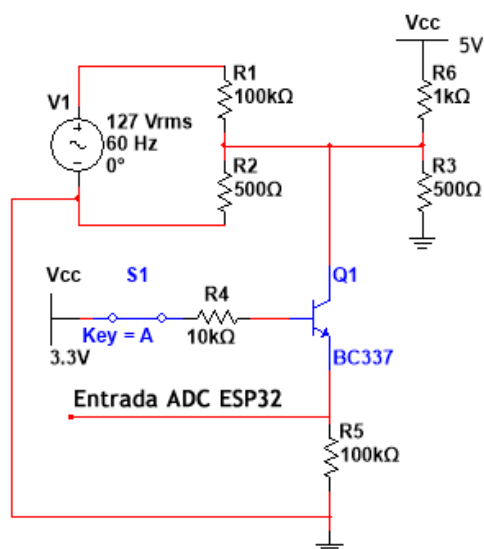


Figura 32: Circuito de multiplexação de sinal de tensão.

Fonte: Autoria própria.

### 3.2 MONTAGEM DO HARDWARE

Após realizados todos os testes foi confeccionado o circuito de medição completo. A Figura 33 mostra o conjunto de circuitos que medem as tensões  $V_a$ ,  $V_b$ ,  $V_c$  e mais 5 sinais de tensão referente ao acionamento de disjuntores.

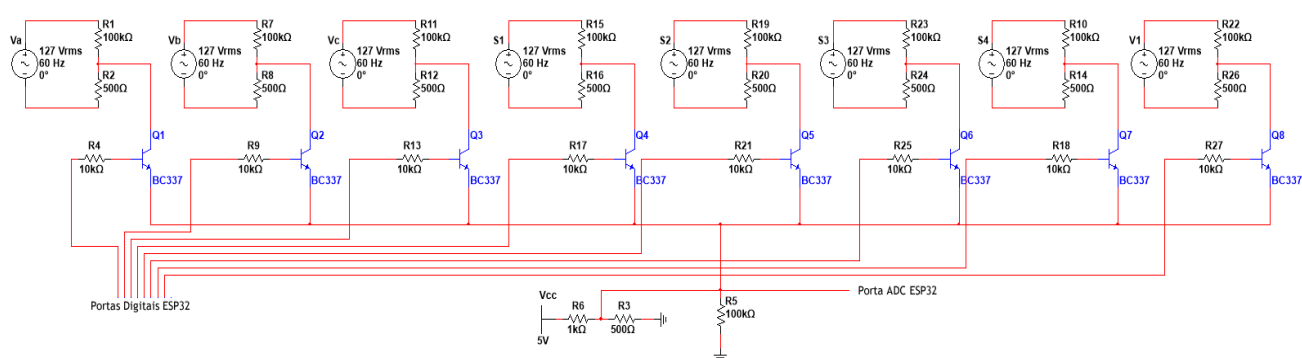


Figura 33: Esquemática do circuito de multiplexação de tensão.

Fonte: Autoria própria.

A Figura 34 mostra o circuito de medição de corrente, que ao total faz a medição de corrente de 5 circuitos.

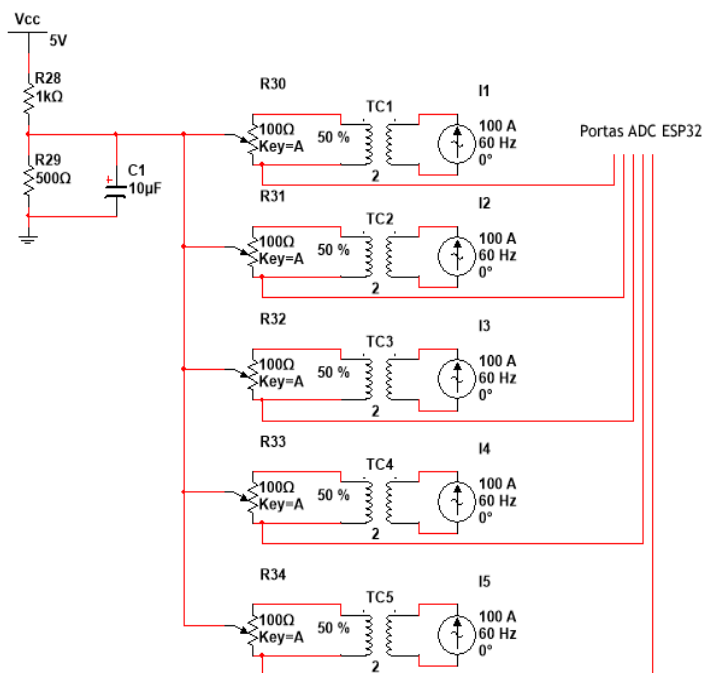


Figura 34: Esquemática de multiplexação de tensão.

Fonte: Autoria própria.

A Figura 35 mostra o protótipo finalizado, parte do circuito foi montado em uma placa ilhada onde são feitas as medições de corrente e também possui os divisores de tensão dos sinais adquiridos dos disjuntores e tensões da rede, outra parte foi confeccionada na *protoboard*, onde foi montada a multiplexação dos sinais de tensão de disjuntores e tensões de rede.

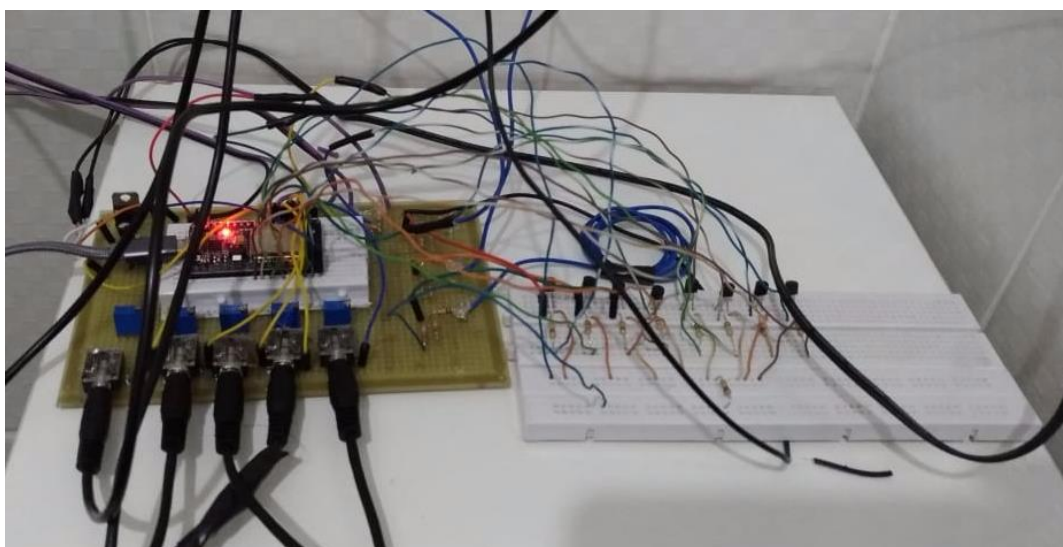


Figura 35: Circuito de medição.

Fonte: Autoria própria

### 3.3 PROGRAMAÇÃO DO MICROCONTROLADOR

Como mostrado anteriormente no item 2.6, o ambiente de programação escolhido para este projeto foi o Arduino IDE, pois este é compatível com a placa eletrônica ESP32 a ser utilizada, e possui bibliotecas como a *Electricity monitoring library* (Emonlib), facilitando assim a obtenção dos dados desejados.

#### 3.3.1 CÓDIGO DE CALIBRAÇÃO

A biblioteca de monitoramento de eletricidade (emonlib), é uma biblioteca do Arduino com comandos de cálculo e monitoramento de grandezas elétricas como tensão, corrente e potências, fornecida gratuitamente pela comunidade *OpenEnergyMonitor* (OPEN ENERGY MONITOR, 2018). Esta biblioteca possui aplicações estritamente relacionadas com o desenvolvimento deste projeto, sua utilização é de extrema importância e facilita grande parte da programação em C do microcontrolador.

Em um primeiro momento foi elaborado um código para calibração das medições como é mostrado no Apêndice “A”, página 84.

A biblioteca Emonlib utilizada no código do Apêndice “A” foi desenvolvida para as placas Arduino que possuem uma tensão de entrada nas portas ADC entre 0 e 5V com uma resolução de 10bits, portanto foi preciso realizar testes práticos para adaptar a biblioteca para a configuração do ESP32 que possui ADC entre 0 e 3,3V e resolução 12bits.

As funções de cálculo contidas dentro da biblioteca Emonlib utilizam constantes de entrada que podem ser alteradas para calibração, ou seja, esses valores deverão ser alterados de acordo com a especificação da medição a ser realizada.

As constantes adotadas foram práticas, ou seja, foram modificadas até os valores do monitor serial da IDE Arduino e dos instrumentos de medição entrarem em paridade.



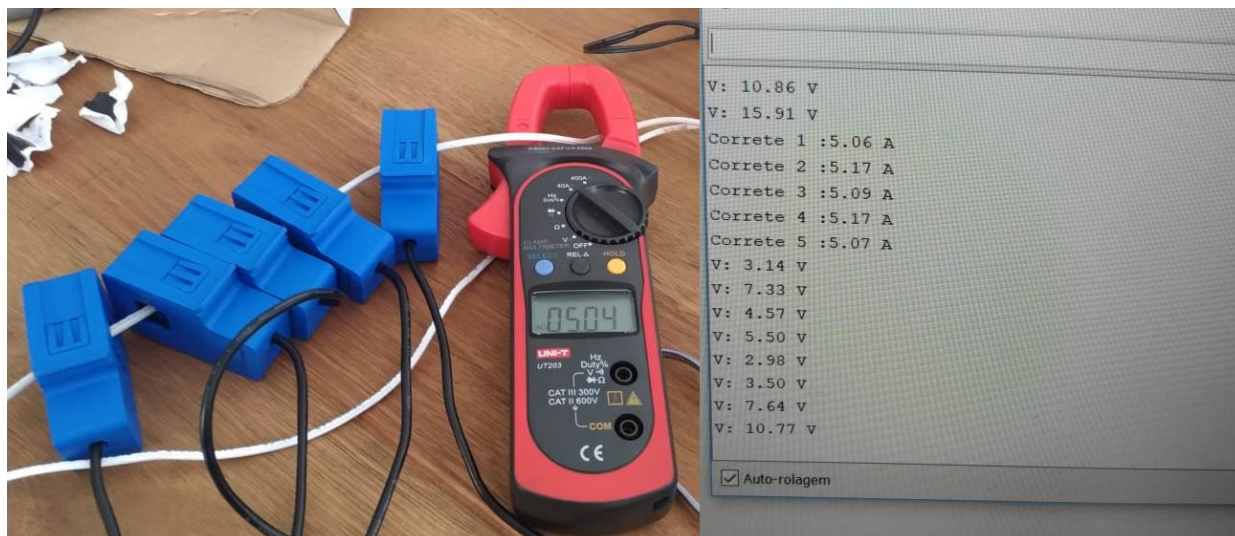


Figura 36: Calibração da corrente.

Fonte: Autoria própria.

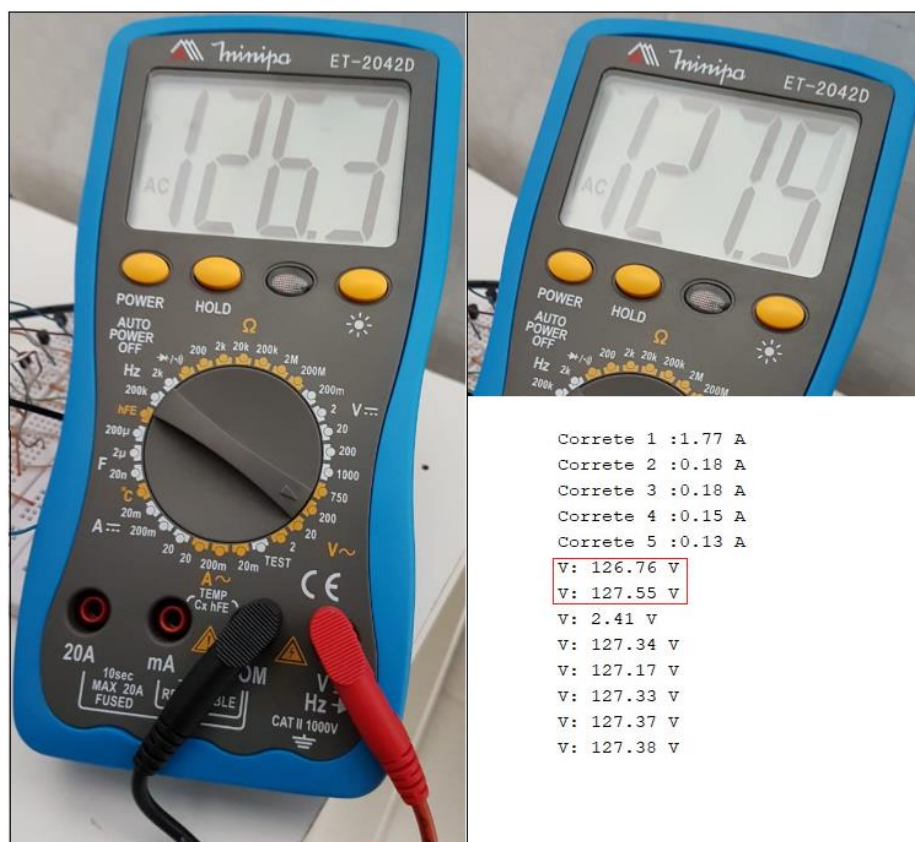


Figura 37: Calibração de tensão.

Fonte: Autoria própria.

Depois dos testes realizados, foi constatado que o valor de calibração utilizado no código neste projeto é de 7 para a corrente, e 1,7 para valores de tensão. Na Figura 36 observa-se o valor real, medido pelo amperímetro alicate, de 5,04A, em comparação com os valores medidos e calculados pela placa eletrônica, que variam de 5,06A a 5,17A. Na Figura 37 observa-se a tensão real, dada pelo multímetro, de 126,3V para a fase um, e de 127,5 para a fase dois. Em seguida, está marcada em vermelho a medição de tensão calculada através do programa, de 126,76V para a fase um e 127,55V para a fase dois.

Para calcular o erro absoluto foram utilizados os valores do amperímetro e do multímetro como reais, o erro do equipamento foi então calculado em comparação com estes valores. Para a corrente, temos o exemplo de uma corrente real de 5,04A, e o valor medido de 5,17A, que nos leva a um erro absoluto que pode ser calculado pela Equação (1).

$$E\% = \frac{|I_m - I_r|}{I_r} \times 100\% \quad (1)$$

Onde:

$I_m$ : Valor de corrente medido pelo programa.

$I_r$ : Valor de corrente real, medido pelo amperímetro alicate.

Logo, chegamos ao valor de erro absoluto igual a:

$$E\% = \frac{|5,17 - 5,04|}{5,04} \times 100\% = 2,58\%$$

O mesmo pode ser feito para calcular o erro absoluto da medição de tensão, desta vez comparando os valores encontrados pelo programa com estes medidos pelo multímetro, através da Equação (2).

$$E\% = \frac{|V_m - V_r|}{V_r} \times 100\% \quad (2)$$

Onde:

Vm: Valor de tensão medido pelo programa.

Vr: Valor de tensão real, medido pelo multímetro.

O que resulta no valor do erro absoluto de tensão igual a:

$$E\% = \frac{|126,76 - 126,3|}{126,3} \times 100\% = 0,36\%$$

### 3.3.2 CONEXÃO *WiFi*

O ESP32 possui módulo *WiFi* integrado, isto é, não é necessário nenhum componente adicional para obter conexão à internet, porém, ainda é necessário realizar a programação da placa para que esta tenha conexão à rede *WiFi* desejada.

Existem exemplos dentro da biblioteca do Emonlib que podem realizar a busca de redes *WiFi*, obter IP fixo e fazer requisição HTTP. Utilizando estes exemplos a configuração pode ser realizada facilmente, primeiramente é necessário testar se o ESP32 está conseguindo identificar a rede desejada, para isto utiliza-se o código do Apêndice “B” página 86;

Após executar o código de *WiFi Scan* pode-se observar através da janela serial a listagem das redes que o ESP32 está enxergando conforme pode ser observado na Figura 38.

```
COM4
scan start
scan done
2 networks found
1: Rede 1 (-39) *
2: Rede 2 (-39)
```

**Figura 38: Listagem de redes obtidas pelo ESP32.**

**Fonte: Autoria própria.**

Após concluir que o ESP32 está identificando da rede desejada utiliza-se o exemplo de requisição HTTP mesclado com o exemplo de IP estático. O exemplo de requisição HTTP será responsável por enviar os dados para uma página PHP que irá processar os dados através do método `$_GET` através de uma variável enviada dentro da URL desejada. O exemplo de IP estático se faz necessário para evitar quaisquer conflitos na rede interna, atribuindo um endereço único de rede. O Apêndice “C”, página 87, mostra como fica a mescla da requisição HTTP e do IP estático.

Após a execução do código do Apêndice “A”, página 84, foi verificado se na janela serial houve retorno do código 200, ou seja, este código confirma que houve resposta da página solicitada.

### 3.3.3 ARMAZENAMENTO DA VARIÁVEL NA URL

Após a confirmação de que existe resposta da página WEB, finaliza-se o código que será utilizado no ESP32 reunindo a informação em somente uma variável que será enviada para um endereço de uma página PHP conforme o Apêndice “D”. Esta página será responsável pela extração da variável da URL através do método `$_GET` registro de todas as informações no banco de dados MySQL conforme o Apêndice “F”.

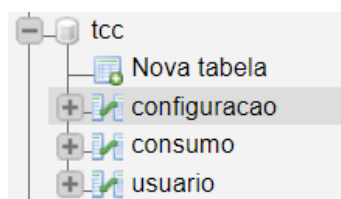
## 3.5 MANIPULAÇÃO DE DADOS

O projeto tem como objetivo facilitar o acesso das informações ao usuário, que poderá observar todas as informações de forma gráfica, em tempo real, através de qualquer equipamento com acesso à internet, seja este um celular, *tablet* ou computador, apenas entrando em um *website*. Para isso os dados coletados devem ser manipulados primeiramente.

### 3.5.1 ESTRUTURA DO BANCO DE DADOS MYSQL

O banco de dados é composto por diferentes tabelas, as quais reúnem informações de acordo com o que o desenvolvedor projetou para ser utilizado. Neste

projeto foi criado um banco de dados chamado “tcc” e as tabelas foram criadas de acordo com a Figura 39.



**Figura 39: Estrutura do banco de dados.**

**Fonte: Autoria própria.**

A tabela “configuração” é responsável por reunir toda informação referente a quais fases cada disjuntor utiliza, quanto de corrente suporta cada disjuntor, níveis de referência de tensão de rede, valores de tarifas. Sua estrutura pode ser verificada na Figura 40.

| #  | Nome          | Tipo    | Colação           | Atributos | Nulo | Padrão                     | Comentários | Extra          |
|----|---------------|---------|-------------------|-----------|------|----------------------------|-------------|----------------|
| 1  | <b>id</b> 🔑   | int(11) |                   |           | Não  | Nenhum wrap (padrão: none) |             | AUTO_INCREMENT |
| 2  | <b>c1</b>     | text    | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |             |                |
| 3  | <b>c2</b>     | text    | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |             |                |
| 4  | <b>c3</b>     | text    | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |             |                |
| 5  | <b>c4</b>     | text    | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |             |                |
| 6  | <b>c5</b>     | text    | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |             |                |
| 7  | <b>va</b>     | text    | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |             |                |
| 8  | <b>vb</b>     | text    | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |             |                |
| 9  | <b>vc</b>     | text    | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |             |                |
| 10 | <b>tarifa</b> | text    | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |             |                |
| 11 | <b>valor</b>  | text    | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |             |                |
| 12 | <b>email</b>  | text    | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |             |                |

**Figura 40: Estrutura da tabela “configuracao” no MySQL.**

**Fonte: Autoria própria.**

A tabela “consumo” reúne todas as informações referentes aos dados enviados do ESP32 para a página de inserção de dados. Sua estrutura pode ser verificada na Figura 41.

| # | Nome       | Tipo     | Colação           | Atributos | Nulo | Padrão                     | Comentários | Extra          |
|---|------------|----------|-------------------|-----------|------|----------------------------|-------------|----------------|
| 1 | id         | int(11)  |                   |           | Não  | Nenhum wrap (padrão: none) |             | AUTO_INCREMENT |
| 2 | hora       | datetime |                   |           | Não  | Nenhum wrap (padrão: none) |             |                |
| 3 | consumo    | text     | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |             |                |
| 4 | desativado | int(11)  |                   |           | Não  | Nenhum wrap (padrão: none) |             |                |

**Figura 41: Estrutura da tabela “consumo” no MySQL.**

**Fonte: Autoria própria.**

A tabela “usuario” possui os dados para acesso do sistema, a utilização desta tabela é de extrema importância para a segurança das informações que constam no sistema supervisorio. Sua estrutura pode ser verificada na Figura 42.

| # | Nome       | Tipo    | Colação           | Atributos | Nulo | Padrão                     | Comentários | Extra          |
|---|------------|---------|-------------------|-----------|------|----------------------------|-------------|----------------|
| 1 | id         | int(11) |                   |           | Não  | Nenhum wrap (padrão: none) |             | AUTO_INCREMENT |
| 2 | usuario    | text    | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |             |                |
| 3 | senha      | text    | latin1_swedish_ci |           | Não  | Nenhum wrap (padrão: none) |             |                |
| 4 | desativado | int(11) |                   |           | Não  | Nenhum wrap (padrão: none) |             |                |

**Figura 42: Estrutura da tabela “usuario” no MySQL.**

**Fonte: Autoria própria**

### 3.5.2 REGISTRO DE DADOS NO BANCO DE DADOS MYSQL

Para registrar os dados no banco de dados MySQL é necessário configurar a conexão primeiramente. Neste projeto toda a configuração de conexão foi reunida em somente um arquivo chamado “classes.php” conforme o Apêndice “E”, página 94. O objetivo de criar um arquivo único para conexões é a possibilidade de inseri-lo em qualquer outro arquivo PHP que necessite conexão com o banco de dados.

A página “bd.php” é responsável por retirar toda informação na URL e registrar no banco de dados. Na URL a informação registrada no ESP32 foi armazenada na variável “consumo”, a URL também possui outra variável chamada “hash”, esta outra é responsável por fornecer a chave de segurança que irá liberar o armazenamento da variável “consumo” no banco de dados como pode ser observado no Apêndice “F”.

A variável consumo que consta na URL é armazenada no formato “Corrente\_1,Corrente\_2,Corrente\_3,Corrente\_4,Corrente\_5/Va,Vb,Vc/Sinal\_D1,Sinal\_D2,Sinal\_D3, Sinal\_D4,Sinal\_D5” conforme pode ser visto no campo “consumo” da Figura 43.

| Coluna     | Tipo     | Função               | Nulo | Valor  |
|------------|----------|----------------------|------|--|
| id         | int(11)  | <input type="text"/> |      | 1  |
| hora       | datetime | <input type="text"/> |      | 2018-05-22 01:06:19  |
| consumo    | text     | <input type="text"/> |      | 3.77,2.68,2.55,2.56,2.55/127.17,127.08,4.46/127.34,127.38,127.62,127.76,126.74 |
| desativado | int(11)  | <input type="text"/> |      | 0  |

**Figura 43: Exemplo da dado inserido pelo ESP32 na tabela “consumo” dentro do MySQL.**

**Fonte: Autoria própria**

O armazenamento da variável neste formato simplifica e minimiza a criação de mais campos dentro do banco de dados, dando mais flexibilidade caso haja uma possível alteração na quantidade de circuitos a serem medidos.

Com a programação PHP é possível obter as informações pertencentes a cada circuito através da separação de caracteres como “,” e “/”. Outro ponto a ser observado é o horário da inserção, a página de inserção foi programada para que a cada dado inserido inclua no campo hora atual do servidor utilizando a função “CURRENT\_TIMESTAMP” do MySQL.

## 3.6 INTERFACE GRÁFICA

### 3.6.1 TELA DE LOGIN

Ao entrar no *website*, a primeira tela que o usuário encontrará será a de *login*, consistindo esta apenas de caixas de texto nas quais devem ser informados o usuário e a senha, como mostrado na Figura 44.

Usuário:

Senha:

**Figura 44: Tela de login.**

**Fonte: Autoria própria**

Os dados informados são comparados com os contidos no banco de dados, se estes forem corretos o usuário terá acesso ao *website*, caso contrário, a mensagem “Erro, usuário ou senha incorretos” aparecerá.

### 3.6.2 TELA INICIAL

Ao inserir os dados corretos, o usuário estará na página inicial do programa, onde se encontram as principais informações e opções, como mostrado na Figura 45.

[\[Sair\]](#)

HISTÓRICO ▾

- Alarmes
- Corrente
- Tensão
- Consumo
- Configurações

| Monitoramento de Circuitos do Quadro Elétrico |            |
|---|------------|
| Ultima aquisição: 10/06/2018 21:24:19         |            |
| Corrente                                      |            |
| Circuito 1:                                   | 1.61 [A]   |
| Circuito 2:                                   | 0.00 [A]   |
| Circuito 3:                                   | 0.00 [A]   |
| Circuito 4:                                   | 1.55 [A]   |
| Circuito 5:                                   | 3.10 [A]   |
| Tensão  |            |
| Va  | 127.38 [V] |
| Vb  | 129.03 [V] |
| Vc  | 0.54 [V]   |
| Estado de Disjuntor                           |            |
| Disjuntor 1:                                  | Fechado    |
| Disjuntor 2:                                  | Fechado    |
| Disjuntor 3:                                  | Fechado    |
| Disjuntor 4:                                  | Fechado    |
| Disjuntor 5:                                  | Fechado    |

**Figura 45: Informações da tela inicial.**

**Fonte: Autoria própria.**



As informações são atualizadas automaticamente em um período de aproximadamente 20 segundos, através do método AJAX como mostrado anteriormente. Através desta tela o usuário tem acesso às demais telas de alarmes, corrente, tensão, consumo e configurações.

### 3.6.3 TELA DE CONFIGURAÇÃO

A Figura 46 mostra a tela de configuração, que contém as informações que devem ser dadas pelo usuário para que o custo possa ser calculado da forma mais correta possível.

|  |   |
|--|---|
| <b>Circuito 1</b><br><b>Tensão</b><br>Va: <input checked="" type="checkbox"/><br>Vb: <input type="checkbox"/><br>Vc: <input type="checkbox"/><br>Disjuntor: 10 [A]<br>Gasto Max. Mensal: R\$ 10            | <b>Circuito 5</b><br><b>Tensão</b><br>Va: <input type="checkbox"/><br>Vb: <input checked="" type="checkbox"/><br>Vc: <input type="checkbox"/><br>Disjuntor: 50 [A]<br>Gasto Max. Mensal: R\$ 14 |
| <b>Circuito 2</b><br><b>Tensão</b><br>Va: <input checked="" type="checkbox"/><br>Vb: <input type="checkbox"/><br>Vc: <input type="checkbox"/><br>Disjuntor: 10 [A]<br>Gasto Max. Mensal: R\$ 11            | <b>Tensões</b><br>Va: 127 [V]<br>Vb: 127 [V]<br>Vc: 0 [V]   |
| <b>Circuito 3</b><br><b>Tensão</b><br>Va: <input checked="" type="checkbox"/><br>Vb: <input checked="" type="checkbox"/><br>Vc: <input type="checkbox"/><br>Disjuntor: 35 [A]<br>Gasto Max. Mensal: R\$ 12 | <b>Preço Tarifas</b><br>Convencional: 0,44056 [R\$]<br>Branca FP: 0,37245 [R\$]<br>Branca Interm.: 0,53154 [R\$]<br>Branca Ponta: 0,81674 [R\$]<br>PIS+COFINS+ICMS: 34 [%]                      |
| <b>Circuito 4</b><br><b>Tensão</b><br>Va: <input checked="" type="checkbox"/><br>Vb: <input type="checkbox"/><br>Vc: <input type="checkbox"/><br>Disjuntor: 50 [A]<br>Gasto Max. Mensal: R\$ 13            | Email: guilherme_183@hotmail.com  |

**Figura 46: Informações da tela de configuração.**

**Fonte: Autoria própria.**

Como mostrado na imagem, o cliente deve informar a qual fase cada circuito pertence,  $V_A$ ,  $V_B$  ou  $V_C$ , para que o programa possa calcular o consumo em cada uma das fases de maneira correta, bem como o valor de capacidade de corrente do disjuntor, para que os alarmes funcionem corretamente. Além disso, também deve ser informado o valor da tarifa convencional, bem como o valor de cada um dos horários, fora de ponta, intermediário e de ponta, da tarifa branca. O usuário também pode

informar o valor máximo mensal a ser gasto no circuito a ser medido. Finalmente, o usuário deve informar o seu *e-mail*, para o qual serão enviados os alertas.

### 3.6.4 TELA DE HISTÓRICO DE CORRENTE E TENSÃO

As telas de histórico de corrente e tensão, são duas telas separadas, contendo as informações de corrente e tensão coletadas pelo protótipo, traduzidas em forma de gráfico através do *Google Charts*, uma ferramenta interativa da empresa *Google*, que transforma os números coletados e especificados em uma combinação de *JavaScript* com *CSS* na forma de gráficos, como o exemplo mostrado nas Figura 51 e Figura 52, na página 69.

A Figura 47 mostra o campo “Data” que está situado na parte superior da tela de histórico de corrente e tensão, o qual pode ser modificado de acordo com a data que o usuário deseja visualizar o histórico.

[Voltar](#)

Data:

maio de 2018

| dom | seg | ter | qua | qui | sex | sáb |
|-----|-----|-----|-----|-----|-----|-----|
| 29  | 30  | 1   | 2   | 3   | 4   | 5   |
| 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 13  | 14  | 15  | 16  | 17  | 18  | 19  |
| 20  | 21  | 22  | 23  | 24  | 25  | 26  |
| 27  | 28  | 29  | 30  | 31  | 1   | 2   |

**Figura 47: Seleção de data a ser visualizada.**

**Fonte: autoria própria.**

### 3.6.5 TELA DE ALARMES

Os alarmes são acionados de acordo com as informações pré-definidas pelo usuário na tela de configurações. Estes alarmes, além de serem mostrados pelo supervisor também são enviados por *e-mail* para o usuário e são ativados em situações consideradas anômalas no sistema, como por exemplo, uma situação de sobrecorrente, atuação de disjuntores, variações de tensões acima do normal, queda de conexão com o ESP32. A Figura 48 mostra um exemplo da atuação destes alarmes.

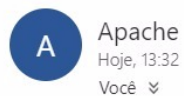
| Alarmes          |  |
|------------------|--|
| <b>Corrente</b>  |  |
| 25/05/2018 16:54 | Circuito 1 / Valor: 10.80[A]                               |
| 25/05/2018 20:08 | Circuito 1 / Valor: 10.72[A]                               |
| 27/05/2018 09:00 | Circuito 1 / Valor: 10.50[A]                               |
| 27/05/2018 19:05 | Circuito 1 / Valor: 11.09[A]                               |
| 28/05/2018 10:02 | Circuito 1 / Valor: 10.87[A]                               |
| 28/05/2018 10:03 | Circuito 1 / Valor: 10.96[A]                               |
| 28/05/2018 12:34 | Circuito 1 / Valor: 10.92[A]                               |
| 28/05/2018 15:32 | Circuito 1 / Valor: 10.39[A]                               |
| 28/05/2018 20:05 | Circuito 1 / Valor: 10.04[A]                               |
| 29/05/2018 09:13 | Circuito 1 / Valor: 10.58[A]                               |
| 29/05/2018 09:14 | Circuito 1 / Valor: 10.61[A]                               |
| 29/05/2018 19:45 | Circuito 1 / Valor: 10.90[A]                               |
| 29/05/2018 19:54 | Circuito 1 / Valor: 11.25[A]                               |
| 29/05/2018 23:37 | Circuito 1 / Valor: 10.81[A]                               |
| 29/05/2018 23:38 | Circuito 1 / Valor: 11.19[A]                               |
| 30/05/2018 10:11 | Circuito 1 / Valor: 10.82[A]                               |
| 30/05/2018 10:24 | Circuito 1 / Valor: 10.86[A]                               |
| <b>Tensão</b>    |  |
| 22/05/2018 12:15 | Tensão Vb / Valor de leitura: 132.91[V] / Variação: 4,65 % |
| 23/05/2018 11:21 | Tensão Va / Valor de leitura: 132.72[V] / Variação: 4,50 % |
| 24/05/2018 16:49 | Tensão Vb / Valor de leitura: 132.74[V] / Variação: 4,52 % |

**Figura 48: Tela de Alarmes.**

**Fonte: Autoria própria.**

Os alarmes vistos na Figura 48 são enviados para o e-mail adicionado na tela “configurações” através da função mail() no PHP. Para que haja envio automático de e-mails existe uma função dos servidores Apache chamada “cronjob” que possibilita a execução de páginas de acordo com o período definido. Neste projeto desenvolvido, o período de execução foi definido para cada 1 minuto. A Figura 49 mostra um exemplo de e-mail enviado quando houve alerta de sobrecorrente.

Alarme sistema supervisorio



29/05/2018 19:45 - Circuito 1 / Valor: 10.90[A]

**Figura 49: Exemplo de email enviado pelo sistema de alarmes.**

**Fonte: Autoria própria.**

## **4 RESULTADOS E DISCUSSÕES**

Neste capítulo serão expostos os resultados finais do projeto, depois de sua instalação em uma residência por um período de duas semanas, também são discutidos pontos importantes como as dificuldades encontradas durante a realização do projeto, conclusões finais e ideias que podem ser implementadas em trabalhos futuros.

### **4.1 TESTE REALIZADO**

Com o protótipo em mãos, os testes foram realizados com este instalado em uma residência de consumo mensal médio de 400 kWh/mês, sendo o monitoramento feito em cinco circuitos distintos, como mostrado na Figura 50.



**Figura 50. Protótipo instalado.**

**Fonte: Autoria própria.**

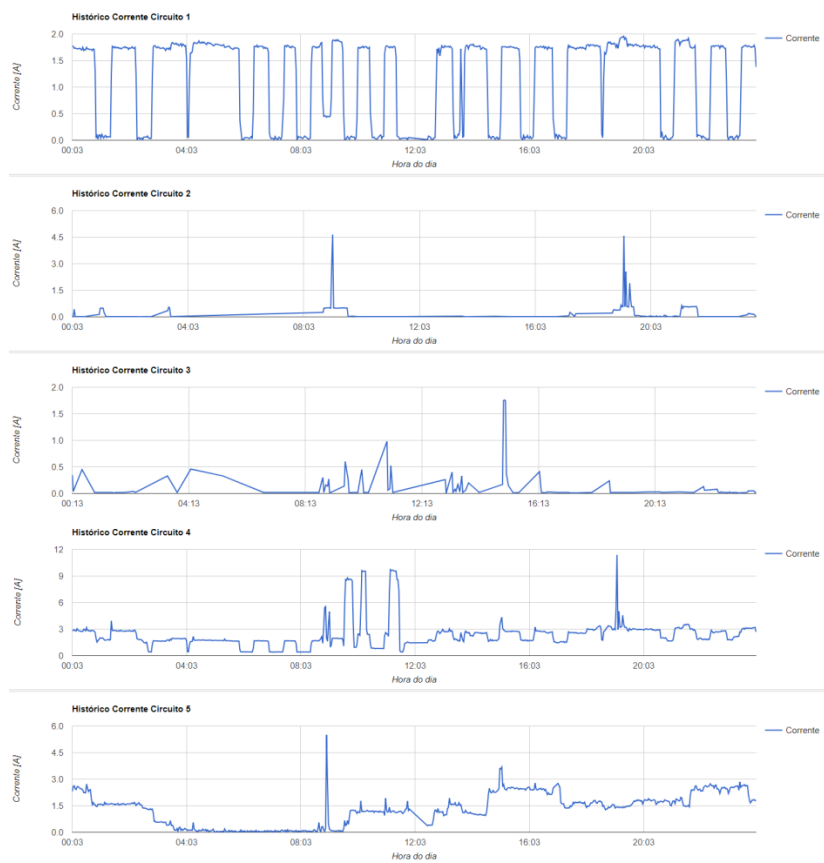
Como observado na imagem, cada transformador de corrente foi instalado em um circuito diferente no quadro, em um total de cinco circuitos, sendo eles:

- Circuito 1: Tomadas da cozinha;
- Circuito 2: Iluminação;
- Circuito 3: Chuveiro elétrico;
- Circuito 4: Fase de entrada 1;
- Circuito 5: Fase de entrada 2.

Além dos transformadores de corrente, cada circuito também teve a tensão mensurada, porém, apenas as tensões obtidas para as duas fases de entrada são utilizadas nos cálculos da potência ativa consumida, as tensões dos demais circuitos são utilizadas exclusivamente para informar ao usuário se o disjuntor foi aberto ou não.

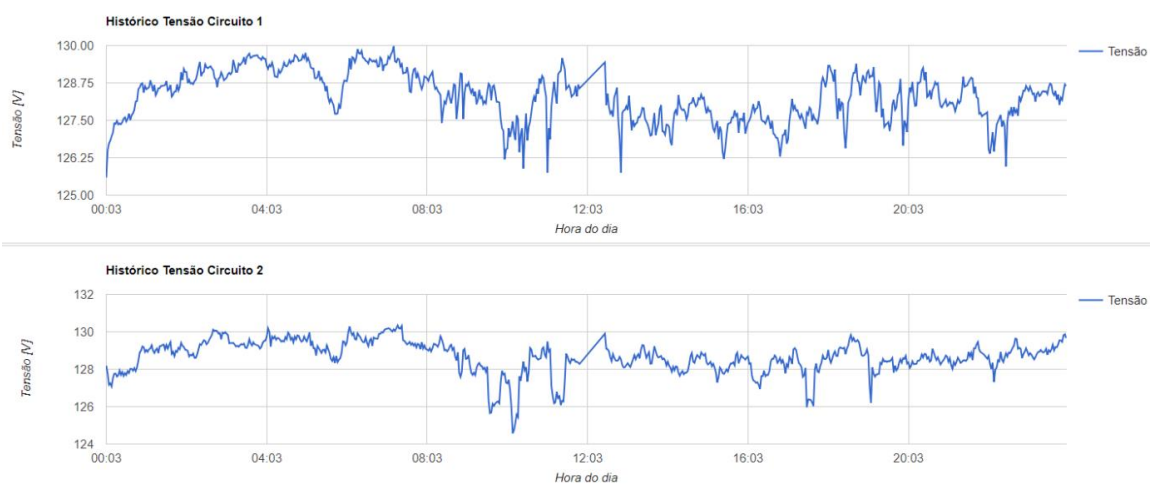
Com o protótipo em posição, a instalação elétrica foi monitorada durante um período de duas semanas, com a finalidade de obter o maior número de dados possível e detectar possíveis anomalias ou erros no projeto. Não foram detectadas anomalias e o protótipo trabalhou sem interrupções durante todo o período. Observa-

se na Figura 51 e Figura 52 as medições de corrente e tensão, respectivamente, nos cinco circuitos testados para a corrente e duas fases de entrada para a tensão, durante o dia 27 de maio de 2018.



**Figura 51. Medições de corrente.**

**Fonte: Autoria própria.**



**Figura 52. Medições de tensão.**

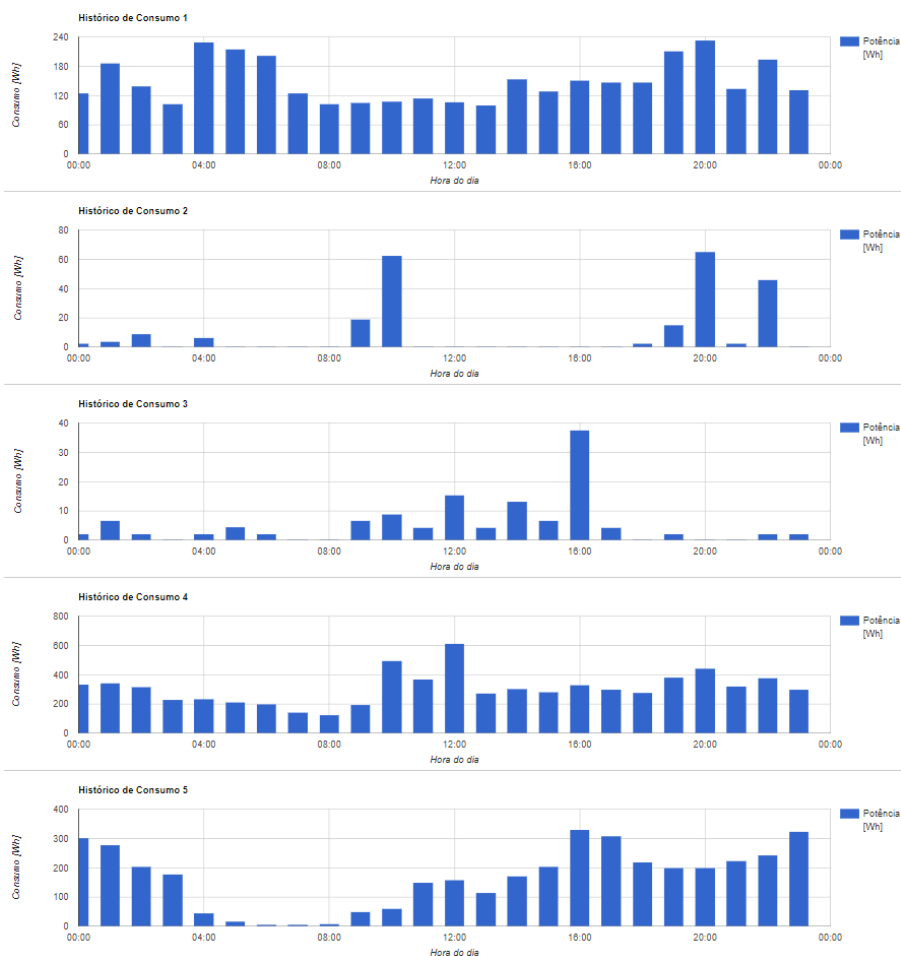
**Fonte: Autoria própria.**

O usuário poderá ter acesso a todas essas informações, podendo selecionar qualquer dia no mês para ver os gráficos de consumo naquele dia, através da opção “Consumo” no *website*, além disso, também terá acesso ao custo através da opção “Consumo”, discutido no item 4.2 a seguir.

## 4.2 CONSUMO E CUSTO

Com o intuito de facilitar a visualização do cliente, que não está interessado em questões de corrente e tensão, também existe uma tela contendo gráficos de consumo, em watts hora (Wh), bem como o valor do custo aproximado em reais, calculado de acordo com o item 4.2.1.

Observam-se na Figura 53 os gráficos de consumo obtidos no teste, no dia 27 de maio de 2018.



**Figura 53. Histórico de consumo em Wh.**

**Fonte: Autoria própria.**

Além dos gráficos contendo o consumo durante todos os períodos do dia, a página também oferece informações importantes como o consumo total em cada um dos circuitos. A partir destes dados são calculados os valores de custo para a tarifa convencional e branca, como mostrado no item a 4.2.1.

#### 4.2.1 COMPARAÇÃO DA TARIFA BRANCA COM A CONVENCIONAL

Para o cálculo dos custos em ambas as tarifas, não foram utilizadas as bandeiras tarifárias, pois estas adicionam um valor baixo no final do mês, e não são importantes para a comparação entre tarifa convencional e branca, já que o valor adicionado é o mesmo para as duas.

Primeiramente, o cálculo foi feito através de equações obtidas no site da Copel. Os valores utilizados, em real/KWh, podem ser encontrados na Tabela 5.

**Tabela 5. Valores da tarifa branca e convencional.**

| TARIFA BRANCA |               |         | CONVENCIONAL | DIFERENÇA |
|---------------|---------------|---------|--------------|-----------|
| RESIDENCIAL   | Ponta         | 0,81674 | 0,44056      | 85%       |
|               | Intermediária | 0,53154 |              | 21%       |
|               | Fora ponta    | 0,37245 |              | -15%      |

**Fonte: Autoria própria.**

Estes valores não estão considerando os valores de impostos a serem pagos, são eles o estadual (ICSM) e federal (PIS/PASEP e COFINS). De acordo com a Copel, os valores dos impostos no mês em questão são de 29% para o ICSM e 5% para os impostos federais (COPEL, 2018a), e o valor final a ser pago pelo consumidor deve ser calculado através da Equação (3).



$$V_F = A \cdot T \cdot \left( 1 + \left( \frac{x}{100 - x} \right) \right) \quad (3)$$

Onde:

$V_F$ : Valor final a ser pago pelo consumidor.

A: Consumo.

T: Tarifa sem o valor de impostos.

X: Imposto a ser pago, neste caso 34%.

A partir dos valores de tarifação e impostos, que devem ser informados pelo usuário na página de configuração, é construída uma tabela com os custos tanto para a tarifa convencional, como para a tarifa branca, esta tabela se encontra na página de consumo do supervisor. Os valores encontrados no teste realizado para o dia 27 de maio de 2018 se encontram na Tabela 6.

**Tabela 6. Comparação de custos entre as tarifas.**

|             | [kWh] | Custo Convencional [R\$] | Branca Total [R\$] |
|-------------|-------|--------------------------|--------------------|
| Circuito 1: | 6,55  | 2,88                     | 3,01               |
| Circuito 2: | 0,45  | 0,30                     | 0,37               |
| Circuito 3: | 2,14  | 1,43                     | 1,28               |
| Circuito 4: | 9,15  | 6,11                     | 6,38               |
| Circuito 5: | 4,64  | 3,10                     | 3,21               |

**Fonte: Autoria própria.**

O *website* informa os valores de custo para ambas as tarifas em todos os circuitos mensurados. Neste teste destacam-se os circuitos 4 e 5, pois estes são os circuitos das duas fases de entrada, logo a soma dos dois circuitos resulta no custo total diário nesta instalação.

Portanto pode-se chegar à conclusão de que neste dia em específico, o dono da residência teria gasto um total de R\$9,21 se utilizada a tarifa convencional, e R\$9,59 se utilizada a tarifa branca, sendo mais favorável a adesão da tarifa convencional. É claro que esta conclusão é apenas para o dia em questão, 27 de maio de 2018, para saber se realmente a adesão da tarifa branca é benéfica para o proprietário, é necessário fazer um estudo durante o mês inteiro, se possível até mesmo em vários meses.

#### 4.3 DIFICULDADES ENCONTRADAS

Este tópico apresenta dificuldades encontradas durante o desenvolvimento do projeto, que não estavam previstas, bem como suas soluções.

Primeiramente, o maior desafio a ser superado foi o número de pinos de conversão analógica para digital, e o número de medições a serem realizadas. A placa eletrônica ESP-WROOM-32 possui um total de 18 portas de conversão de valores analógicos para digitais, tecnicamente seriam portas o suficiente para desenvolver todas as medições do projeto, que são cinco de corrente, e sete de tensão, somando um total de 12 pinos. Porém estas portas ADC estão divididas em dois canais, chamados de *ADC1* e *ADC2*, infelizmente os 10 pinos pertencentes ao canal *ADC2* não podem ser utilizados em conjunto com o *Wi-Fi* da placa, gerando conflito entre eles. Como o *Wi-Fi* é um componente essencial neste projeto, este problema se mostrou um grande obstáculo a ser superado.

Para solucionar este problema, foi desenvolvido um circuito de multiplexação, mostrado anteriormente no item 3.2, este circuito é responsável por dividir as 8 tensões medidas, para que estas sejam medidas uma a uma, em um mesmo pino de conversão analógica para digital. Com este circuito, foi possível economizar pinos da placa eletrônica, tendo espaço suficiente para o objetivo final.

#### 4.4 ANÁLISE DE CUSTO DO SISTEMA

O custo total do sistema desenvolvido pode ser encontrado na Tabela 7.

**Tabela 7. Relação de custos do protótipo.**

| <b>Nome do componente</b>           | <b>Custo</b>     |
|-------------------------------------|------------------|
| 1 placa ilhada 10x15                | R\$10,00         |
| 5 conectores jack P2 fêmea          | R\$10,00         |
| Resistores e capacitores            | R\$1,00          |
| 10 metros de fio 1,5mm <sup>2</sup> | R\$6,50          |
| 5 trimpot de 100 ohm                | R\$7,50          |
| 8 transistores BC337                | R\$1,36          |
| 1 fonte 5V/1A                       | R10,00           |
| 5 TC YHDC SCT-013 100A              | R\$100,00        |
| 1 microcontrolador ESP-WROOM-32     | R\$35,00         |
| <b>TOTAL</b>                        | <b>R\$171,36</b> |

**Fonte: Autoria própria.**

É pertinente observar que a maior parte do custo do sistema vem dos sensores de corrente YHDC de 100A, e neste projeto desenvolvido, foram utilizados sensores para 5 circuitos diferentes apenas para fins de teste. Na instalação deste sistema em uma residência real, podem existir variações no número de sensores, para 2 sensores se o cliente quiser analisar apenas as fases de entrada, barateando o custo total do sistema, ou até mesmo para mais de 5 sensores, se o cliente quiser monitorar um número maior de circuitos. A adição de novos sensores pode ser feita, requerendo apenas algumas modificações no circuito e na programação.

## **5 CONCLUSÕES**

Este trabalho teve como objetivo o desenvolvimento de um sistema supervisor residencial, que oferece informações como potência consumida e custo em reais, da maneira mais visual e simplificada possível, através de qualquer aparelho que tenha conexão à internet. Além disso, oferecer ao usuário comparações entre a tarifa branca e a convencional, auxiliando a modalidade a ser adotada pelo seu consumidor.

Com a instalação deste sistema em uma residência, o usuário passa a ter maior controle sobre a energia elétrica consumida, tendo uma visão ampla de onde e quando esta energia é utilizada, podendo assim tomar decisões que levem a diminuição do consumo de energia. Todavia, o real valor deste projeto está na comparação de custo da tarifa convencional e branca, auxiliando na decisão de sua escolha e podendo

desta forma trazer grandes retornos financeiros ao cliente.

Vale ressaltar que o custo de implementação do protótipo é adaptável ao número de circuitos que o usuário queira medir, quanto maior o número de circuitos a serem medidos, mais caro será. No protótipo desenvolvido foram utilizados 5 circuitos, cujo valor foi considerado ideal pela equipe por fornecer informações suficientes para tirar conclusões sobre a eficiência do projeto.

Existe a preocupação com a segurança do usuário, o sistema necessita de um maior desenvolvimento nesta área, para que sua instalação seja segura, e não comprometa a estética do quadro elétrico.

Finalmente, os resultados obtidos com a instalação do sistema foram positivos, atingindo os objetivos iniciais de medição de potência ativa, com um erro de aproximadamente 2,5%. Comprovando o grande potencial existente para desenvolvimento deste tipo de produto, que agora passa a ter real valor financeiro ao usuário.

## 5.1 TRABALHOS FUTUROS

Para o desenvolvimento de trabalhos futuros, destacam-se as seguintes sugestões:

- Medição de fator de potência.
- Cálculo de potência realizado ponto a ponto.
- Garantia de sincronismo dos valores de tensão e corrente.
- Utilização de um multímetro com certificado de calibração rastreado.
- Possível intervenção nos circuitos: ligar e desligar cargas através da interface web.
- Redução de ruído de medição.
- Ampliação da possibilidade de escolha de diversos períodos de gráficos.
- Identificação da frequência de rede.
- Utilização da conexão *bluetooth* do ESP32 para configuração da rede *WiFi*.
- Aplicação de um visor LCD.

## 5.2 CONSIDERAÇÕES GERAIS

O código fonte do projeto encontra-se disponível na plataforma Github através do *link*:

[https://github.com/guilhermenap/sistema\\_supervisorio\\_energia\\_eletrica\\_residencial](https://github.com/guilhermenap/sistema_supervisorio_energia_eletrica_residencial)

## REFERÊNCIAS

ANEEL - AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA. **BIG - Banco de Informações de Geração**. 2017a. Disponível em: <<http://www2.aneel.gov.br/aplicacoes/capacidadebrasil/OperacaoCapacidadeBrasil.cfm>>. Acesso em: 11 de Novembro de 2017.

ANEEL. **Tarifas para consumidores de baixa e alta tensão**. 2017b. Disponível em: <<http://www.aneel.gov.br/tarifas-consumidores>>. Acesso em 13 de novembro de 2017.

ARDUINO. **loop()**. 2017a. Disponível em: <<https://www.arduino.cc/en/Reference/Loop>>. Acesso em: 15 de outubro de 2017.

ARDUINO. **setup()**. 2017b. Disponível em: <<https://www.arduino.cc/en/Reference/Setup>>. Acesso em: 28 de outubro de 2017.

ARDUINO. **Libraries**. 2017c. Disponível em: <<https://www.arduino.cc/en/Reference/Libraries>>. Acesso em: 28 de outubro de 2017.

BETINI, R. C. ESTRATÉGIAS DE PESQUISA E DESENVOLVIMENTO PARA A REDUÇÃO DO USO DE COMBUSTÍVEIS FÓSSEIS. **Energias renováveis: Políticas Públicas e Planejamento Energético**, Edição Digital, p 100-112, 2014. Disponível em:<

[http://repositorio.utfpr.edu.br/jspui/bitstream/1/942/1/Energias%20Renov%C3%A1veis\\_Pereira%2C%20Thulio%20C%C3%ADcero%20Guimar%C3%A3es\\_2014.pdf](http://repositorio.utfpr.edu.br/jspui/bitstream/1/942/1/Energias%20Renov%C3%A1veis_Pereira%2C%20Thulio%20C%C3%ADcero%20Guimar%C3%A3es_2014.pdf)>.

Acesso em: 16 de novembro de 2017.

COPEL. **ETC 4.11 – Especificação técnica para medidor eletrônico de energia**. 2015. Disponível em: <http://www.copel.com/hpcopel/root/nivel2.jsp?endereco=%2Fhpcopel%2Froot%2Fpagcopel2.nsf%2Fdocs%2F17857CE54578ABC4032573F300700740>. Acesso em 13 de novembro de 2017.

COPEL. **Tributos**. 2018a. Disponível em: <<http://www.copel.com/hpcopel/root/nivel2.jsp?endereco=%2Fhpcopel%2Froot%2Fpagcopel2.nsf%2Fdocs%2F3EE1D015FA3F433203257EE6003E128E>>. Acesso em 30

de maio de 2018.

COPEL. **Tarifa Branca**. 2018b. Disponível em: <<http://www.copel.com/hpcopel/root/nivel2.jsp?endereco=%2Fhpcopel%2Frural%2Fpagcopel2.nsf%2Fdocs%2FB0CA4C8DF4B62F98832581F00058CCF9>>. Acesso em 25 de maio de 2018.

ELIPSE. **Elipse E3**. 2018. Disponível em: <<https://www.elipse.com.br/produto/elipse-e3/>>. Acesso em 10 de junho de 2018.

EPE - EMPRESA DE PESQUISA ENERGÉTICA. **Projeção da demanda de energia elétrica**. 2017a. Disponível em: <[http://www.epe.gov.br/mercado/Documents/DEA%20001\\_2017%20-%20Proje%C3%A7%C3%B5es%20da%20Demanda%20de%20Energia%20El%C3%A9trica%202017-2026\\_VF.pdf](http://www.epe.gov.br/mercado/Documents/DEA%20001_2017%20-%20Proje%C3%A7%C3%B5es%20da%20Demanda%20de%20Energia%20El%C3%A9trica%202017-2026_VF.pdf)>. Acesso em 11 de novembro de 2017.

EPE - EMPRESA DE PESQUISA ENERGÉTICA, **Anuário estatístico de energia elétrica 2017, ano base 2016**. 2017b. Disponível em: <<http://www.epe.gov.br/AnuarioEstatisticodeEnergiaEletrica/Forms/Anurio.aspx>>. Acesso em 13 de novembro de 2017.

ESPRESSIF. **Placas eletrônicas de desenvolvimento** 2018a. Disponível em: <<https://www.espressif.com/en/products/hardware/development-boards>>. Acesso em 14 de maio de 2018.

ESPRESSIF. **ESP32 Datasheet** 2018b. Disponível em: <[https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)>. Acesso em 14 de maio de 2018.

ESPRESSIF. **ESP-WROOM-32 Datasheet** 2018c. Disponível em: <[https://www.espressif.com/sites/default/files/documentation/esp-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp-wroom-32_datasheet_en.pdf)>. Acesso em 14 de maio de 2018.

GALVÃO, J.; BERMANN, C. Crise hídrica e energia: conflitos no uso múltiplo das águas. **Estudos avançados**, v. 29, n. 84, p. 43-68, 2015. Disponível em: <<http://www.scielo.br/pdf/ea/v29n84/0103-4014-ea-29-84-00043.pdf>>. Acesso em: 15 de Maio de 2017.

GITHUB. **Arduino core for the ESP32**. 2018. Disponível em <<https://github.com/espressif/arduino-esp32>>. Acesso em 14 de maio de 2018.

GOLDEMBERG, J.; LUCON, O. Energia e meio ambiente no Brasil. **Estudos avançados**, v. 21, n. 59, p. 7-20, 2007. Disponível em: <<http://www.scielo.br/pdf/ea/v21n59/a02v2159>>. Acesso em: 29 de Abril 2017.

HOMAUTOMATION. **Current monitoring with non-invasive sensor and arduino**. 2018. Disponível em: <<http://www.homautomation.org/2013/09/17/current-monitoring-with-non-invasive-sensor-and-arduino/>>. Acesso em: 28 de maio de 2018.

LAWSON, A.; PEREIRA, G. **Termelétricas e seu papel na matriz energética Brasileira**. Fundação Getulio Vargas, 2017. Disponível em: <[http://fgvenergia.fgv.br/sites/fgvenergia.fgv.br/files/coluna\\_opiniao\\_fevereiro\\_-\\_termeletricas\\_-\\_andre\\_e\\_guilherme.pdf](http://fgvenergia.fgv.br/sites/fgvenergia.fgv.br/files/coluna_opiniao_fevereiro_-_termeletricas_-_andre_e_guilherme.pdf)>. Acesso em: 11 de junho de 2018.

JURIZATO, L. A.; PEREIRA, P. S. R. **Sistemas supervisórios orientados a objetos**. Nova odessa, network Technologies, 2003. Disponível em: <<http://centralmat.com.br/Artigos/Mais/sistemasSupervisorios.pdf>>. Acesso em: 10 de outubro de 2017.

KAMAMI. **Modelo da placa ESP-WROOM-32**. 2017. Disponível em: <<https://kamami.pl/esp32/563392-esp32-devkitc-plytka-rozwojowa-iot-z-modulem-wi-fi-esp-wroom-32-firmy-espressif.html>>. Acesso em: 19 de maio de 2018.

Landis+Gyr. **Empresa de medidores de energia eletrônicos**. 2017. Disponível em <<https://www.landisgyr.com/>>. Acesso em 13 de novembro de 2017.

MCROBERTS, M. **Arduino básico**. 1 ed. São Paulo: Novatec, 2011. 443 p.

ML - MERCADO LIVRE. **Medidor Consumo Energia Wifi Pelo Celular Pow Potência**. 2017. Disponível em: <[https://produto.mercadolivre.com.br/MLB-923216084-medidor-consumo-energia-wifi-pelo-celular-pow-potncia-\\_JM?source=gps](https://produto.mercadolivre.com.br/MLB-923216084-medidor-consumo-energia-wifi-pelo-celular-pow-potncia-_JM?source=gps)>. Acesso em: 13 de novembro de 2017.

MOZILLA. **Javascript**. 2017a. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 15 de outubro de 2017.



MOZILLA. **Javascript básico.** 2017b. Disponível em: <[https://developer.mozilla.org/pt-BR/docs/Aprender/Getting\\_started\\_with\\_the\\_web/JavaScript\\_basico](https://developer.mozilla.org/pt-BR/docs/Aprender/Getting_started_with_the_web/JavaScript_basico)>. Acesso em: 15 de outubro de 2017.

MOZILLA. **AJAX: Primeiros passos.** 2017c. Disponível em: <[https://developer.mozilla.org/pt-BR/docs/AJAX/Getting\\_Started](https://developer.mozilla.org/pt-BR/docs/AJAX/Getting_Started)>. Acesso em: 15 de outubro de 2017.

MOZILLA. **CSS.** 2017d. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acesso em: 15 de outubro de 2017.

MURATORI, J. R.; BÓ, P. H. D. **Automação residencial: Histórico, definições e conceitos.** São Paulo – FATEC-SP, 2011. Disponível em: <[http://static2.voltimum.com/sites/www.voltimum.com.br/files/pdflibrary/04\\_automacao\\_residencial1.pdf](http://static2.voltimum.com/sites/www.voltimum.com.br/files/pdflibrary/04_automacao_residencial1.pdf)>. Acesso em: 10 de outubro de 2017.

NBR 8377. **Medidor de energia ativa – especificação.** 1995. Disponível em: <<https://www.target.com.br/produtos/normas-tecnicas/29614/nbr8377-medidor-de-energia-ativa>>. Acesso em 13 de novembro de 2017.

NEGREIROS, L. R.; DIAS, E. W. **Automação de arquivos no Brasil: Os discursos e seus momentos.** Universidade Federal de Minas Gerais, 2007. Disponível em: <[http://www.brapci.ufpr.br/brapci/\\_repositorio/2010/05/pdf\\_b9c69b711d\\_0010723.pdf](http://www.brapci.ufpr.br/brapci/_repositorio/2010/05/pdf_b9c69b711d_0010723.pdf)>. Acesso em: 10 de outubro de 2017.

NIEDERAUER, Juliano. **Desenvolvendo websites com PHP.** São Paulo: Novatec, 2004.

ONS - OPERADOR NACIONAL DO SISTEMA ELÉTRICO. **Volume Útil dos Principais Reservatórios.** 2015.

OPEN ENERGY MONITOR. **Electricity monitoring,** 2018. Disponível em: <<https://learn.openenergymonitor.org/electricity-monitoring/ac-power-theory/introduction>>. Acesso em: 25 de maio de 2018.

ORACLE. **MySQL.** 2017. Disponível em: <<https://www.oracle.com/br/mysql/index.html>>. Acesso em: 15 de outubro de 2017.

PHP. **Manual do PHP**. 2017. Disponível em: <[http://php.net/manual/pt\\_BR/preface.php](http://php.net/manual/pt_BR/preface.php)>. Acesso em: 15 de outubro de 2017.

RAMOS, M. C.; ANDRADE, Valcimar Silva. Desenvolvimento, construção e calibração de uma central de monitoramento de consumo de energia elétrica e de água utilizando o microcontrolador Arduino. **Anais do ENEDS**, 2015. Disponível em: <[http://www3.ifmg.edu.br/site\\_campi/v/images/arquivos\\_governador\\_valadares/Artigo\\_Eneds\\_2015.pdf](http://www3.ifmg.edu.br/site_campi/v/images/arquivos_governador_valadares/Artigo_Eneds_2015.pdf)>. Acesso em: 15 de outubro de 2017.

SEGATTO, E. C.; COURY, D. V. Redes neurais artificiais recorrentes aplicadas na correção de sinais distorcidos pela saturação de transformadores de corrente. **Sba: Controle & Automação Sociedade Brasileira de Automatica**, v. 17, n. 4, p. 424-436, 2006. Disponível em: <<http://www.scielo.br/pdf/ca/v17n4/a04v17n4.pdf>>. Acesso em: 15 de outubro de 2017.

SHAJAHAN, Altaf Hamed; ANAND, A. Data acquisition and control using arduino-android platform: Smart plug. In: **Energy Efficient Technologies for Sustainability (ICEETS)**, 2013 International Conference on. IEEE, 2013. p. 241-244. Disponível em: <<http://ieeexplore.ieee.org/document/6533389/>>. Acesso em: 15 de outubro de 2017.

TANCREDI, M.; ABBUD, O. A. **Por que o Brasil está trocando as hidrelétricas e seus reservatórios por energia mais cara e poluente?** 2013. Disponível em <[http://www2.senado.leg.br/bdsf/bitstream/handle/id/496199/TD128-MarcioTancredi\\_OmarAbbud.pdf?sequence=1](http://www2.senado.leg.br/bdsf/bitstream/handle/id/496199/TD128-MarcioTancredi_OmarAbbud.pdf?sequence=1)>. Acesso em: 29 de abril de 2017.

TANCREDI, M.; ABBUD, O. A. **Transformações recentes da matriz brasileira de geração de energia elétrica: causas e impactos principais**. 2010. Disponível em: <<http://www2.senado.leg.br/bdsf/bitstream/handle/id/182500/Texto69-OmareTancredi.pdf?sequence=4>>. Acesso em: 10 de maio de 2017.

TEZA, VANDERLEI RABELO. **Alguns aspectos sobre a automação residencial – domótica**. Dissertação para mestrado de ciência da computação. Universidade Federal de Santa Catarina, maio de 2002. Disponível em: <<https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/83015/212312.pdf?sequence=1&isAllowed=y>>. Acesso em: 10 de outubro de 2017.

TOLMASQUIM, M. T.; GUERREIRO, A.; GORINI, R. Matriz energética brasileira: uma prospectiva. **Novos estudos-CEBRAP**, n. 79, p. 47-69, 2007. Disponível em <<http://www.scielo.br/pdf/nec/n79/03.pdf>>. Acesso em: 10 de maio de 2017.

VAGAPOV, Yuriy. **Comparative Analysis and Practical Implementation of the ESP32 Microcontroller Module for the Internet of Things**. 2017. Disponível em: <[https://www.researchgate.net/publication/320273388\\_Comparative\\_Analysis\\_and\\_Practical\\_Implementation\\_of\\_the\\_ESP32\\_Microcontroller\\_Module\\_for\\_the\\_Internet\\_of\\_Things](https://www.researchgate.net/publication/320273388_Comparative_Analysis_and_Practical_Implementation_of_the_ESP32_Microcontroller_Module_for_the_Internet_of_Things)>. Acesso em: 14 de maio de 2018.

VELASCO, Loana Nunes. **Análise experimental de erros de medição de energia elétrica ativa em medidores eletromagnéticos tipo indução, sujeitos a distorções harmônicas de correntes e tensões, em sistemas equilibrados e desequilibrados**. São Paulo, 2007. 187 f. . Disponível em: <<https://repositorio.unesp.br/handle/11449/87277>>. Acesso em: 10 de outubro de 2017.

W3SCHOOLS. **HTML Introdução**. 2017a. Disponível em: <[https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)>. Acesso em: 15 de outubro de 2017.

W3SCHOOLS. **Introdução ao SQL**. 2017b. Disponível em: <[https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp)>. Acesso em: 15 de outubro de 2017.

W3SCHOOLS. **PHP 5 Tutorial**. 2017c. Disponível em: <<https://www.w3schools.com/php/>>. Acesso em: 15 de outubro de 2017.

W3SCHOOLS. **JavaScript Tutorial: Tente você mesmo**. 2017d. Disponível em: <[https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_myfirst](https://www.w3schools.com/js/tryit.asp?filename=tryjs_myfirst)>. Acesso em: 15 de outubro de 2017.

W3SCHOOLS. **AJAX Introdução: Tente você mesmo**. 2017e. Disponível em: <[https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_ajax\\_first](https://www.w3schools.com/js/tryit.asp?filename=tryjs_ajax_first)>. Acesso em: 15 de outubro de 2017.

W3SCHOOLS. **CSS Tutorial: Tente você mesmo**. 2017f. Disponível em: <[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_default](https://www.w3schools.com/css/tryit.asp?filename=trycss_default)>. Acesso em: 28 de

outubro de 2017.

YHDC. **Datasheet: Current transformer**. Modelo SCT013-000V. Publicação Eletrônica, 2015. Disponível em: <[http://www.yhdc.us/ENpdf/SCT013-000V-0-100A-0-1V\\_en.pdf](http://www.yhdc.us/ENpdf/SCT013-000V-0-100A-0-1V_en.pdf)>. Acesso em: 15 de outubro de 2017.

YHDC. **13mm hole Split core current transformer**. 2017. Disponível em: <[http://yhdc.en.alibaba.com/product/519312318-218696501/13mm\\_hole\\_Split\\_core\\_current\\_transformer.html](http://yhdc.en.alibaba.com/product/519312318-218696501/13mm_hole_Split_core_current_transformer.html)>. Acesso em: 15 de novembro de 2017.

YU, X.; YI, C. Design and Implementation of the Website Based on PHP & MYSQL. In: **E-Product E-Service and E-Entertainment (ICEEE), 2010 International Conference on**. IEEE, 2010. p. 1-4. Disponível em: <<http://ieeexplore.ieee.org/abstract/document/5661595/>>. Acesso em: 15 de outubro de 2017.

## APÊNDICE A – CÓDIGO CALIBRAÇÃO ESP32

```

#include "EmonLib.h" //Inclui biblioteca Emonlib
#define ADC_BITS 12 //Define a leitura como 12 Bits
EnergyMonitor emon[13]; //Criação de 14 instâncias dentro do vetor emon

int pin[]={23,22,1,3,21,19,18,5}; //Pinos digitais utilizados para multiplexação

//----- Função de multiplexação-----
int multiplex(int ativar)
{
  for(int i=0;i<8;i++)
  {
    if(ativar == i)
    {
      digitalWrite(pin[i], HIGH);
    }
    else
    {
      digitalWrite(pin[i], LOW);
    }
  }

  emon[ativar+6].calcVI(200,200); // calcVI(x,y) onde x é a quantidade de meia onda a ser
  mensurada e y é o atraso para a proxima medição

  Serial.print("V: ");
  Serial.print(emon[ativar+6].Vrms);
  Serial.print(" V");
  Serial.print("\n");
}

//----- Função de medição de corrente-----
int corrente(int ativar)
{
  Serial.print("Correte "+String(ativar)+" :");
  Serial.print(emon[ativar].calcIrms(1480)); //Numero de amostras que contém em 1 período de
  60Hz
  Serial.print(" A");
  Serial.print("\n");
}

//-----

void setup()
{
  Serial.begin(115200); //Inicializa a janela serial em 115200 baud

  //-----Configuração de corrente-----
  emon[1].current(36, 7); //current(x,y) onde x é o pino ADC a ser mensurado e y é a constante de
  calibração
  emon[2].current(39, 7);
  emon[3].current(34, 7);
  emon[4].current(35, 7);
  emon[5].current(32, 7);

  //-----Configuração de tensão-----
  for(int i=6;i<14;i++)
  {
    emon[i].voltage(33, 70, 1.7); //voltage(x,y,z) onde x é o pino ADC a ser mensurado, y é a
    constante de calibração e z é a constante de avanço de fase.
  }
}

```

```
//-----  
for(int i=0;i<8;i++)  
{  
  pinMode(pin[i], OUTPUT); //define os pinos do vetor pin como saída.  
}  
}  
  
void loop()  
{  
  for(int i=1;i<6;i++)  
  {  
    corrente(i); //looping de medição de corrente;  
  }  
  
  for(int i=0;i<8;i++)  
  {  
    multiplex(i); //looping de medição de tensão  
  }  
  
  delay(2000);  
}
```

## APÊNDICE B – CÓDIGO DE PROCURA DE REDES *WiFi* ESP32

```

#include "WiFi.h"

void setup()
{
  Serial.begin(115200);

  // Set WiFi to station mode and disconnect from an AP if it was previously connected
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);

  Serial.println("Setup done");
}

void loop()
{
  Serial.println("scan start");

  // WiFi.scanNetworks will return the number of networks found
  int n = WiFi.scanNetworks();
  Serial.println("scan done");
  if (n == 0) {
    Serial.println("no networks found");
  } else {
    Serial.print(n);
    Serial.println(" networks found");
    for (int i = 0; i < n; ++i) {
      // Print SSID and RSSI for each network found
      Serial.print(i + 1);
      Serial.print(": ");
      Serial.print(WiFi.SSID(i));
      Serial.print(" (");
      Serial.print(WiFi.RSSI(i));
      Serial.print(")");
      Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN)? " ":"*");
      delay(10);
    }
  }
  Serial.println("");

  // Wait a bit before scanning again
  delay(5000);
}

```

## APÊNDICE C – CÓDIGO DE REQUISIÇÃO HTTP E IP ESTÁTICO ESP32

```

#include <Arduino.h>
#include <WiFi.h>
//----- Bibliotecas necessárias-----
#include <WiFiMulti.h>
#include <HTTPClient.h>
#define USE_SERIAL Serial

//-----Atribui um IP estático -----

IPAddress local_IP(192, 168, 1, 115);//IP estático escolhido
IPAddress gateway(192, 168, 1, 1);//Gateway padrão da rede
IPAddress subnet(255, 255, 255, 0);//Máscara de Sub-Rede
IPAddress primaryDNS(8, 8, 8, 8); //Servidor DNS primário opcional
IPAddress secondaryDNS(8, 8, 4, 4); // Servidor DNS secundário opcional
//-----

WiFiMulti wifiMulti;//Criação de uma nova instância

void setup() {
  //--- Parâmetros de inicialização da conexão Wifi-----
  USE_SERIAL.begin(115200);
  for(uint8_t t = 4; t > 0; t--) {
    USE_SERIAL.printf("[SETUP] WAIT %d...\n", t);
    USE_SERIAL.flush();
    delay(1000);
  }

  //-----Executa IP fixo-----
  if (!WiFi.config(local_IP, gateway, subnet, primaryDNS, secondaryDNS)) {
    Serial.println("STA Failed to configure");
  }

  //-----Nome e senha da rede desejada-----
  wifiMulti.addAP("SSID", "PASSWORD");//nome e senha da rede desejada
  //-----
}

void loop() {
  // wait for WiFi connection
  if((wifiMulti.run() == WL_CONNECTED)) {
    HTTPClient http;
    http.begin("http://paginatcc/bd.php?consumo=variavel"); //URL a ser enviada onde a
    variável será manipulada dentro do arquivo php.
    USE_SERIAL.print("[HTTP] GET...\n");
    int httpCode = http.GET();
    if(httpCode > 0) {
      USE_SERIAL.printf("[HTTP] GET... code: %d\n", httpCode);// Imprime o código de
      requisição na janela serial, se for 200 é porque foi concluído.
      if(httpCode == HTTP_CODE_OK) {
        String payload = http.getString();
        USE_SERIAL.println(payload);
      }
    }
  }
}

```



```
    }  
  } else {  
    USE_SERIAL.printf("[HTTP] GET... failed, error: %s\n",  
http.errorToString(httpCode).c_str());  
  }  
  http.end();  
}  
delay(5000);  
}
```

## APÊNDICE D – CÓDIGO FONTE ESP32

```

//-----Inclusão de bibliotecas-----
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiMulti.h>
#include <HTTPClient.h>
#include "EmonLib.h"
//-----

Serial #define USE_SERIAL Serial // Declaração do termo USE_SERIAL com o mesmo valor de

#define ADC_BITS 12 //Leitura 12 bits

WiFiMulti wifiMulti; //instância do WiFi

IPAddress local_IP(192, 168, 0,81);
IPAddress gateway(192, 168, 0, 1);
IPAddress subnet(255, 255, 0, 0);
IPAddress primaryDNS(8, 8, 8, 8);
IPAddress secondaryDNS(8, 8, 4, 4);

//-----Criação das instâncias-----
EnergyMonitor emon1;
EnergyMonitor emon2;
EnergyMonitor emon3;
EnergyMonitor emon4;
EnergyMonitor emon5;

EnergyMonitor emon6;
EnergyMonitor emon7;
EnergyMonitor emon8;

EnergyMonitor emon9;
EnergyMonitor emon10;
EnergyMonitor emon11;
EnergyMonitor emon12;
EnergyMonitor emon13;

//-----Definição de variáveis-----
float cont;
float acumulado1,acumulado2,acumulado3,acumulado4,acumulado5;
float v1,v2, v3;
float f1,f2,f3,f4,f5;
String conteudoweb;
//-----

int pino[]={23,22,1,3,21,19,18,5};//vetor que contem os pinos digitais que serão acionados

//-----Função de mutiplexação-----
int multiplex(int ativar)
{
  for(int a=0;a<8;a++)
  {
    if(ativar == a)
    {
      digitalWrite(pino[a], HIGH);
    }
    else
    {
      digitalWrite(pino[a], LOW);
    }
  }
}

```

```

    }
  }
//-----

void setup()
{
  USE_SERIAL.begin(115200);//inicia a janela serial 115200 baud para USE_SERIAL

//----- Inicialização do USE_SERIAL-----
  for(uint8_t t = 4; t > 0; t--) {
    USE_SERIAL.printf("[SETUP] WAIT %d...\n", t);
    USE_SERIAL.flush();
    delay(1000);
  }
//-----

  wifiMulti.addAP("rede", "senha");//Nome da rede e senha

  Serial.begin(115200);//inicia a janela normal serial 115200 baud

//----- Função de ativação do IP estático -----
  if (!WiFi.config(local_IP, gateway, subnet, primaryDNS, secondaryDNS)) {
    Serial.println("STA Failed to configure");
  }

//----- Definições de pinos e constantes de calibração-----
  emon1.current(36, 7);
  emon2.current(39, 7);
  emon3.current(34, 7);
  emon4.current(35, 7);
  emon5.current(32, 7);

  emon6.voltage(33, 70, 1.7);
  emon7.voltage(33, 70, 1.7);
  emon8.voltage(33, 70, 1.7);

  emon9.voltage(33, 70, 1.7);
  emon10.voltage(33, 70, 1.7);
  emon11.voltage(33, 70, 1.7);
  emon12.voltage(33, 70, 1.7);
  emon13.voltage(33, 70, 1.7);
//-----Definição dos pinos do vetor como saídas digitais-----

  for(int a=0;a<8;a++)
  {
    pinMode(pino[a], OUTPUT);
  }
//-----

}

void loop()
{
//----- Multiplexação, definição do tamanho da amostra e obtenção do valor de tensão
  multiplex(0);//ativa somente o pino digital da posição 0 para multiplexação
  emon6.calcVI(200,200);//calcVI(x,y) onde x é a quantidade de meia onda a ser
  mensurada e y é o atraso para a proxima medição
  v1 = emon6.Vrms; //Pega valor da tensão RMS

  multiplex(1);
  emon7.calcVI(200,2000);//Vb
  v2 = emon7.Vrms;

  multiplex(2);

```

```

emon8.calcVI(200,200);//Vc
v3 = emon8.Vrms;

multiplex(3);
emon9.calcVI(200,200);//falha1
f1 = emon9.Vrms;

multiplex(4);
emon10.calcVI(200,200);//falha2
f2 = emon10.Vrms;

multiplex(5);
emon11.calcVI(200,200);//falha3
f3 = emon11.Vrms;

multiplex(6);
emon12.calcVI(200,200);//falha4
f4 = emon12.Vrms;

multiplex(7);
emon13.calcVI(200,200);//falha5
f5 = emon13.Vrms;
//-----Looping que faz medição de cada corrente 3 vezes

if(cont<3)
{
  emon1.calcIrms(1480);//Realiza 1480 amostras por ciclo
  emon2.calcIrms(1480);
  emon3.calcIrms(1480);
  emon4.calcIrms(1480);
  emon5.calcIrms(1480);

  float i1=emon1.Irms; //Pega valores da corrente RMS
  float i2=emon2.Irms;
  float i3=emon3.Irms;
  float i4=emon4.Irms;
  float i5=emon5.Irms;

  // O valor abaixo de 0.3 A é considerado como nulo, ou seja, um erro a ser considerado
  para medição de corrente não invasiva quando o valor é muito baixo gera uma tensão flutuante não
  significativa.
  if (i1>0.3)
  {
    acumulado1=acumulado1+i1;
  }
  if(i2>0.3)
  {
    acumulado2=acumulado2+i2;
  }
  if(i3>0.3)
  {
    acumulado3=acumulado3+i3;
  }
  if(i4>0.3)
  {
    acumulado4=acumulado4+i4;
  }
  if(i5>0.3)
  {
    acumulado5=acumulado5+i5;
  }
}

```

```

    //delay(100);
  }
  //-----Faz a média entre as 3 medições de corrente
  else
  {
    acumulado1=acumulado1/cont;
    acumulado2=acumulado2/cont;
    acumulado3=acumulado3/cont;
    acumulado4=acumulado4/cont;
    acumulado5=acumulado5/cont;
  }
  //-----
  cont=0;//zera o contador

  //----- Inicia a conexão WiFi
  Serial.print(".....");
  if((wifiMulti.run() == WL_CONNECTED)) {
    HTTPClient http;
    USE_SERIAL.print("[HTTP] begin...\n");
    conteudoweb=
String(acumulado1)+","+String(acumulado2)+","+String(acumulado3)+","+String(acumulado4)+
","+String(acumulado5)+","+String(v1)+","+String(v2)+","+String(v3)+","+String(f1)+","+String(
f2)+","+String(f3)+","+String(f4)+","+String(f5); //Reune toda a informação obtida em uma
variável.

    http.begin("https://paginatcc/bd.php?consumo="+conteudoweb+"&hash=3452345246456");
    //Envio de variavel para a pagina PHP com o hash de segurança (será explicado posteriormente).
    USE_SERIAL.print("[HTTP] GET...\n");
    int httpCode = http.GET();
    if(httpCode > 0) {
      USE_SERIAL.printf("[HTTP] GET... code: %d\n", httpCode);//Retorna o código 200
      execute corretamente.
      if(httpCode == HTTP_CODE_OK) {
        String payload = http.getString();
        USE_SERIAL.println(payload);
      }
    } else {
      USE_SERIAL.printf("[HTTP] GET... failed, error: %s\n",
      http.errorToString(httpCode).c_str());//Retorna o código de errado caso não seja executado
      corretamente.
    }

    http.end();
  }
}
cont++;
}

```

## APÊNDICE E – ARQUIVO DE CONEXÃO CLASSES.PHP

```
<?php
$hostname_servidor = "ip_do_servidor";
$database_servidor = "tcc";
$username_servidor = "usuario";
$password_servidor = "senha";
$servidor=mysqli_connect($hostname_servidor,$username_servidor,$password_servidor) or
trigger_error(mysqli_error(),E_USER_ERROR);
?>
```

**APÊNDICE F – REGISTRO DE DADOS MYSQL BD.PHP**

```
<?php require_once('classes.php');
$consumo=addslashes($_GET['consumo']);
$hash=addslashes($_GET['hash']);//
if($consumo and $hash=='3452345246456')
{
mysqli_select_db($servidor, $database_servidor);//conecta com o banco de dados
mysqli_query($servidor,"INSERT INTO `consumo` (`id`, `hora`, `consumo`,
`desativado`) VALUES (NULL, CURRENT_TIMESTAMP, '$consumo', "");");//registra os
dados
}
?>
```