

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
ENGENHARIA ELÉTRICA

NOHARTI AFONSO DE OLIVEIRA

**MEDIÇÃO DE ENERGIA EM CIRCUITOS SEGMENTADOS E APRESENTAÇÃO  
VIA WEB**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO  
2019

NOHARTI AFONSO DE OLIVEIRA

**MEDIÇÃO DE ENERGIA EM CIRCUITOS SEGMENTADOS E APRESENTAÇÃO  
VIA WEB**

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina Trabalho de Conclusão do curso de Engenharia Elétrica da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Prof. Dr. Luís Fernando Caparroz Duarte



**Universidade Tecnológica Federal do Paraná**  
**Campus Cornélio Procópio**  
**Departamento Acadêmico de Elétrica**  
**Curso de Engenharia Elétrica**



## **FOLHA DE APROVAÇÃO**

**Noharti Afonso de Oliveira**

### **Medição de consumo de energia segmentado por circuito e apresentação via web**

Trabalho de conclusão de curso apresentado às 10:20hs do dia 07/06/2019 como requisito parcial para a obtenção do título de Engenheiro Eletricista no programa de Graduação em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Avaliadora composta pelos professores abaixo assinados. Após deliberação, a Banca Avaliadora considerou o trabalho aprovado.

---

Prof(a). Dr(a). Luis Fernando Caparroz Duarte - Presidente (Orientador)

---

Prof(a). Dr(a). Pedro Carvalhaes Dias - (Membro)

---

Prof(a). Dr(a). André Sanches Fonseca Sobrinho - (Membro)

## RESUMO

DE OLIVEIRA, Noharti Afonso. Medição de energia em circuitos segmentados e apresentação via *web*. Trabalho de Conclusão de Curso – Engenharia Elétrica. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2019.

Neste trabalho é desenvolvido um medidor eletrônico do consumo de energia elétrica residencial, que oferece aos usuários dados sobre o consumo dos circuitos elétricos de sua residência mostrados remotamente em uma página da *web*.

Para tal, utiliza-se um chip medidor de energia, que recebe os sinais de corrente e de tensão provenientes dos disjuntores do quadro de distribuição da residência. Em seguida, esses sinais são processados para se obter os valores de potência ativa, reativa e de energia consumida. Os resultados obtidos são enviados para um microcontrolador e, então, transmitidos para uma interface *web* para a visualização do usuário, através de um módulo *Wi-Fi*.

**Palavras-chave:** Medidor de energia elétrica, Microcontrolador, Interface *web*, *Wi-Fi*.

## ABSTRACT

DE OLIVEIRA, Noharti Afonso. Measurement of energy in segmented circuits and web presentation. Trabalho de Conclusão de Curso – Engenharia Elétrica. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2019.

In this project is developed an electronic meter of the consumption of residential electrical energy, which provides to the user data about the consumption of the electrical circuits in their home shown on a web page.

For this, is used an energy meter chip, which receives the current and voltage signals coming from the circuit breakers of the residence's distribution board. Then, these signals are processed to obtain the values of active power, reactive power and consumed energy. The results obtained are sent to a microcontroller, and then transmitted to a web interface for user's viewing, by a Wi-Fi module.

**Keywords:** Electrical energy meter, Microcontroller, Web interface, Wi-Fi.

## LISTA DE FIGURAS

Figura 1 - Medidor de Samuel Gardiner.....	18
Figura 2 - Medidor de Thomas Edison.....	18
Figura 3 - Medidor de Thomas Duncan.....	19
Figura 4 - Esquema do medidor de energia eletromecânico.....	20
Figura 5 - Diagrama de blocos de um medidor eletrônico.....	21
Figura 6 - Medidor eletrônico.....	21
Figura 7 - Triângulo das potências.....	23
Figura 8 - Composição da tarifa energética.....	28
Figura 9 - Estrutura básica de um transformador ideal.....	31
Figura 10 - Esquema de um TC.....	32
Figura 11 - Esquema de ligação para o protocolo SPI.....	34
Figura 12 - Esquema de ligação para comunicação UART.....	35
Figura 13 - Arquitetura do TCP-IP.....	37
Figura 14 - Diagrama de blocos do hardware.....	39
Figura 15 - Sensor de corrente SCT 013-050.....	40
Figura 16 - Esquemático da saída do sensor.....	40
Figura 17 - Circuito divisor de tensão.....	41
Figura 18 - Transformador 110/220V para 9V.....	42
Figura 19 - Circuito divisor de tensão.....	43
Figura 20 - Chip medidor ADE7753.....	43
Figura 21 - Pinagem do chip medidor de energia.....	44
Figura 22 - Diagrama de blocos medidor monofásico ADE7753.....	44
Figura 23 - PIC utilizado no projeto.....	45
Figura 24 - Pinagem do microcontrolador PIC18F4520.....	46
Figura 25 - Módulo <i>Wi-Fi</i> ESP8266-12E.....	46
Figura 26 - Pinagem do módulo ESP8266-12E.....	47
Figura 27 - Circuito da fonte de alimentação.....	48
Figura 28 - Interface do Eagle.....	49
Figura 29 - Interface do MPLAB X IDE.....	50
Figura 30 - Interface do Visual Studio.....	51
Figura 31 - Interface do MySQL.....	52
Figura 32 - Circuito da placa de comunicação.....	54
Figura 33 - Layout da placa de comunicação.....	55

Figura 34 - Placa de comunicação finalizada. ....	55
Figura 35 - Circuito da placa de medição. ....	56
Figura 36 - Layout da placa de medição. ....	57
Figura 37 - Placa de medição finalizada (TOP). ....	57
Figura 38 - Placa de medição finalizada (BOTTOM). ....	58
Figura 39 - Fluxograma do Firmware. ....	59
Figura 40 - Diagrama do banco de dados MySQL. ....	61
Figura 41 - Tela de <i>login</i> do site. ....	61
Figura 42 - Tela de cadastro do site. ....	62
Figura 43 - Tela de inicial do site. ....	62
Figura 44 - Tela de consumo diário. ....	63
Figura 45 - Tela de consumo por período. ....	63
Figura 46 - Tela de medidas. ....	64
Figura 47 - Tela do usuário. ....	64
Figura 48 - Escrita de dados no padrão da comunicação SPI do CI ADE7753. ....	65
Figura 49 - Leitura de dados no padrão da comunicação SPI do CI ADE7753. ....	66
Figura 50 - Comunicação SPI. ....	66
Figura 51 - Montagem do protótipo para testes. ....	67
Figura 52 - Teste de medição da corrente da fonte do notebook. ....	68
Figura 53 - Gráfico linearidade das medições de corrente. ....	69
Figura 54 - Medição do consumo diário do ferro de passar. ....	69
Figura 55 - Valor gasto no consumo diário do ferro de passar. ....	70
Figura 56 - Medição da tensão ferro de passar. ....	70
Figura 57 - Medição da tensão ferro de passar. ....	71
Figura 58 - Medição do fator de potência ferro de passar. ....	71
Figura 59 - Medição do consumo por período do secador de cabelo. ....	72
Figura 60 - Valor gasto no consumo do período do secador de cabelo. ....	72
Figura 61 - Medição consumo por período do secador de cabelo na tela do celular. ....	73
Figura 62 - Medição da potência reativa do secador de cabelo. ....	73

## LISTA DE TABELAS

Tabela 1 - Categoria de consumidores. ....	29
Tabela 2 - Pinos da comunicação SPI. ....	33
Tabela 3 - Versões do padrão IEEE 802.11.....	36
Tabela 4 - Reguladores de tensão utilizados no projeto. ....	48
Tabela 5 - Comparativo entre medições de corrente com o ADE7753. ....	68
Tabela 6 - Custos do projeto. ....	74



## LISTA DE ABREVIATURAS E SIGLAS

ANEEL	Agência Nacional de Energia Elétrica
AP	<i>Access Point</i>
ADC	<i>Analog to Digital Converter</i>
CAD	<i>Computer Aided Design</i>
CI	Circuito Integrado
CSS	<i>Cascading Style Sheets</i>
COFINS	Contribuição para o financiamento de seguridade social
EEPROM	<i>Electrically-Erasable Programmable Read-Only Memory</i>
EPE	Empresa de Pesquisa Energética
FTP	<i>File Transfer Protocol</i>
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
ICMS	Imposto Sobre Circulação de Mercadoria e Serviços
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
MSB	<i>Most Significant Bit</i>
ONS	Operador Nacional do Sistema Elétrico
PCI	Placa de Circuito Impresso
PEE	Programa de Eficiência Energética
PIC	<i>Peripheral Interface Controller</i>
PIS	Programa de Interação Social
PROCEL	Programa Nacional de Conservação de Energia
PRODIST	Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional
RMS	<i>Root Mean Square</i>
SI	Sistema Internacional de Medidas
SMTP	<i>Simple Mail Transfer Protocol</i>
SOC	<i>System on Chip</i>
SPI	<i>Serial Peripheral Interface</i>
SQL	<i>Structured Query Language</i>
TC	Transformador de Corrente
TCP	<i>Transmission Control Protocol</i>

TI	Tecnologia da Informação
TP	Transformador de Potência
UDP	<i>User Datagram Protocol</i>
UFRJ	Universidade Federal do Rio de Janeiro
URL	<i>Uniform Resource Locator</i>
USART	<i>Universal Synchronous Asynchronous Receiver Transmitter</i>
WLAN	<i>Wireless Local Area Network</i>

## SUMÁRIO

<b>1. INTRODUÇÃO</b>	<b>14</b>
<b>1.1 Delimitação do tema</b>	<b>16</b>
<b>1.2 Motivação</b>	<b>16</b>
<b>1.3 Objetivos</b>	<b>16</b>
1.3.1 Objetivo Geral	16
1.3.2 Objetivos Específicos	17
<b>1.4 Organização do trabalho</b>	<b>17</b>
<b>2. FUNDAMENTAÇÃO TEÓRICA</b>	<b>17</b>
<b>2.1 Medidores de energia elétrica</b>	<b>17</b>
2.1.1 Medidores eletromecânicos	19
2.1.2 Medidores eletrônicos	20
<b>2.2 Potência</b>	<b>22</b>
2.2.1 Potência ativa	23
2.2.2 Potência reativa	24
2.2.3 Potência aparente	25
2.2.4 Fator de potência	26
<b>2.3 Estrutura tarifária</b>	<b>27</b>
2.3.1 Grupo de consumidores	28
2.3.2 Bandeiras tarifárias	29
<b>2.4 Transformadores</b>	<b>30</b>
2.4.1 Transformadores de potência	32
2.4.2 Transformadores de corrente	32
<b>2.5 Comunicação SPI</b>	<b>33</b>
<b>2.6 Comunicação USART</b>	<b>35</b>
<b>2.7 Wi-Fi</b>	<b>35</b>
<b>2.8 Protocolo TCP-IP</b>	<b>37</b>

2.9	Internet.....	38
3.	<b>MATERIAIS E MÉTODOS.....</b>	<b>38</b>
3.1	<b>Hardware.....</b>	<b>39</b>
3.1.1	Sensor de corrente SCT 013-050 .....	39
3.1.2	Sensor de tensão.....	42
3.1.3	Módulo medidor de energia .....	43
3.1.4	Microcontrolador .....	45
3.1.5	Módulo Wi-Fi.....	46
3.1.6	Fonte de alimentação.....	47
3.2	<b>Softwares .....</b>	<b>48</b>
3.2.1	Eagle.....	49
3.2.2	MPLAB X IDE .....	50
3.2.3	Visual Studio .....	50
3.2.4	MySQL.....	51
3.2.5	Google Charts.....	52
4.	<b>DESENVOLVIMENTO DO PROJETO .....</b>	<b>53</b>
4.1	<b>Placas de circuito impresso.....</b>	<b>53</b>
4.1.1	Placa de comunicação .....	53
4.1.2	Placa de medição .....	56
4.2	<b>Programação .....</b>	<b>58</b>
4.2.1	Programa embarcado no microcontrolador.....	58
4.2.2	Desenvolvimento do site .....	60
5.	<b>RESULTADOS .....</b>	<b>65</b>
5.1	<b>Custos do projeto.....</b>	<b>74</b>
6.	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>75</b>
6.1	<b>Sugestão para trabalhos futuros .....</b>	<b>745</b>
	<b>REFERÊNCIAS.....</b>	<b>77</b>

<b>APÊNDICE A – Código do PIC18F4520.....</b>	<b>82</b>
<b>APÊNDICE B – Código do ESP8266. ....</b>	<b>87</b>
<b>APÊNDICE C – Código do ADE7753.....</b>	<b>94</b>

## 1. INTRODUÇÃO

O uso de energia elétrica é hoje uma das questões que tem um grande destaque no mundo. Com o crescimento contínuo da população, de produção de alimentos, indústrias e comércio há, também, o aumento do número de cargas inseridas no sistema. Para conseguir suprir essa nova demanda, faz-se necessária a construção de novas usinas geradoras de energia elétrica, porém causando um impacto direto na natureza cada vez maior.

No Brasil, atualmente estão instalados 4.675 empreendimentos de geração de energia, com uma potência total de 151.532.427 kW, sendo a maioria desses, cerca de 61,41%, provenientes de fontes hídricas (ANEEL, 2017). Ainda há em construção 35 empreendimentos hidrelétricos e mais 177 com a construção não iniciada (ANEEL, 2017). Logo em seguida, com 16,83% do total de empreendimentos, estão as usinas termelétricas. Essas são, na atualidade, as fontes mais usadas no país.

Segundo Luiz Pinguelli Rosa, membro da Academia Brasileira de Ciências e diretor da Coordenação dos Programas de Pós-Graduação em Engenharia da Universidade Federal do Rio de Janeiro (UFRJ), o país deve investir mais nas hidrelétricas (com menos impacto ambiental e proveniente de uma fonte mais limpa) do que nas termelétricas (que faz o uso de combustíveis fósseis). Ainda, ele prevê o fim gradativo das termelétricas, as quais vê como ineficazes e poluentes.

Apesar de ter um potencial hídrico muito alto e se beneficiar com a construção de novas usinas hidrelétricas, o Brasil tem que se preocupar com as questões ambientais e sociais. Ainda que vistas como uma fonte de “energia limpa”, as hidrelétricas não podem ser ditas como a melhor solução, pois muitos são os impactos causados pela sua construção, tais como: perda de biodiversidade, alteração no clima, erosão marginal, pequenos tremores de terra com a acomodação de placas, elevação de carbono na atmosfera, realocação de grupos indígenas (AYA e EDGAR, 2005 apud LEITE, 2005).

A redução dos níveis de precipitação, ocorrida nos anos de 2014 e 2015, teve como consequência a redução das aflúncias naturais nos reservatórios, a diminuição da geração de energia, além de restringir o uso de água em várias cidades brasileiras (GALVÃO e BERMANN, 2015). Para ajudar a suprir a demanda de energia, foi preciso acionar as usinas termelétricas, cujo custo de geração é muito mais caro. A partir disso, foi criado pela ANEEL (Agência Nacional de Energia Elétrica), no ano

de 2015, o sistema de bandeiras tarifárias, que repassa aos consumidores um novo preço da energia elétrica a partir da utilização das termelétricas (COPEL, 2015).

As alterações climáticas que vêm acontecendo e o aumento contínuo do consumo trarão um desafio no futuro para suprir a demanda de energia elétrica. No Brasil, existe a lei 10.295/2001, também conhecida como a Lei de Eficiência Energética, na qual se estabelece níveis máximos de consumo e mínimos de eficiência energética (PROCEL, 2014). Além disso, o PROCEL (Programa Nacional de Conservação de Energia) junto com a ANEEL e a EPE (Empresa de Pesquisa Energética), promovem programas e incentivos para o uso eficiente de energia e combate ao desperdício. A ANEEL lançou o PEE (Programa de Eficiência Energética) para ajudar as distribuidoras de energia a desenvolver diferentes formas de uso racional da energia, o qual fixa que as empresas concessionárias de distribuição de energia elétrica devem aplicar pelo menos 0,5% de sua receita operacional líquida em projetos de eficiência energética (PROCEL, 2014).

Além dos fatos citados, o desperdício de energia no Brasil é relativamente grande. Segundo um levantamento feito pela Abesco (Associação Brasileira de Empresas de Serviço de Conservação de Energia), nos últimos três anos houve um desperdício de 143,6 mil GWh no país, esse volume é 1,4 vezes maior que toda a produção de energia da hidrelétrica de Itaipu no ano de 2016, esse número ainda representaria uma economia de R\$ 61,7 bilhões na conta dos consumidores (ABESCO, 2017).

Por isso é preciso criar um hábito responsável de consumo, para que no futuro ainda existam recursos disponíveis para a geração de energia, com o mínimo de impacto o possível. Porém, mesmo sendo conscientizado, o consumidor não sabe o quanto está gastando no momento em que acende uma luz, assiste televisão ou deixa algum equipamento elétrico no modo *stand by*.

Já existem alguns sistemas de monitoramento de energia em tempo real no mercado, eles são conhecidos como *Home Energy Monitors*, mas ainda não são tão populares por não apresentar preços acessíveis para a maioria dos brasileiros.

O trabalho desenvolvido visa a construção de um protótipo, focando na redução de custos de implementação deste sistema em relação aos disponíveis no mercado. Além disso, o mesmo é adaptado aos tipos de tarifas em funcionamento no Brasil, além de possuir uma instalação simples e uma interface amigável e de fácil entendimento aos consumidores.

## 1.1 Delimitação do tema

O presente trabalho apresenta o desenvolvimento um sistema embarcado para o monitoramento do consumo de energia elétrica no setor residencial que mostra em tempo real o valor de energia consumido, utilizando os valores do sistema tarifário nacional.

## 1.2 Motivação

A questão energética do Brasil na atualidade é um tema que vem chamando bastante atenção. Nos períodos em que a incidência de chuva é menor, além do aumento das tarifas energéticas pelo uso das termelétricas para suprir o consumo de energia, há também um aumento dos impactos ambientais causados. Por isso, cada vez mais se faz necessária uma melhor gestão da energia utilizada pela população.

Com a evolução dos dispositivos eletrônicos, ficou cada vez mais fácil o desenvolvimento de equipamentos para automação residencial, isso inclui sistemas de monitoramento de energia. Esse último é de grande interesse pois fornece aos consumidores a possibilidade de utilizar energia de um modo consciente e eficiente, além de diminuir o desperdício de energia.

## 1.3 Objetivos

### 1.3.1 Objetivo Geral

O objetivo deste trabalho é o desenvolvimento de um medidor de energia que monitora o consumo, por circuito, de uma residência e mede as seguintes grandezas elétricas: tensão, corrente, potência reativa, fator de potência e energia consumida. Tal medidor monitora o perfil de consumo elétrico e apresenta ao consumidor as informações em tempo real e via *web*, com o intuito de informar o hábito de consumo do cliente.



### 1.3.2 Objetivos Específicos

Os objetivos específicos para este projeto são:

- Levantar bibliografias acerca de eficiência energética;
- Pesquisar formas de medição de energia;
- Aprender sobre módulo *Wi-Fi* e apresentação via *web*;
- Criar o protótipo do *hardware* do medidor de consumo de energia;
- Visar um baixo custo no desenvolvimento e na escolha dos materiais;

### 1.4 Organização do trabalho

O presente trabalho tem a estrutura dividida nos seguintes capítulos:

Capítulo 2 - Tipos de medidores de energia elétrica e modelos atuais.

Capítulo 3 - Apresenta todo referencial técnico e componentes usados em sua construção.

Capítulo 4 - Apresenta o desenvolvimento do protótipo.

Capítulo 5 - Aplicação prática, resultados obtidos e custo.

Capítulo 6 - Conclusão e sugestão de trabalho futuro.

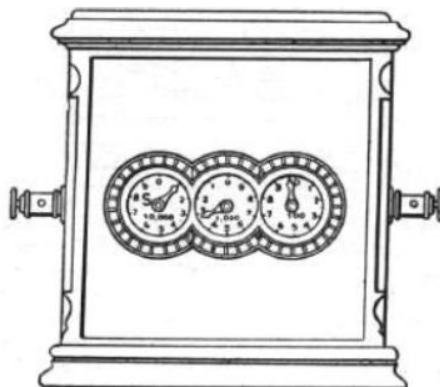
## 2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é efetuada a apresentação dos conceitos teóricos necessários para a elaboração do projeto proposto, uma vez que são abordados aspectos inerentes ao seu desenvolvimento.

### 2.1 Medidores de energia elétrica

O medidor de energia elétrica é um dispositivo eletromecânico, ou eletrônico, que faz a medida do consumo de energia de uma residência. Presente na maioria das residências do mundo, o primeiro medidor foi criado em 1872, por Samuel Gardiner, operava em corrente contínua e o resultado da medição era expresso em lâmpada-hora (unidade de medida que se baseava no tempo em que uma lâmpada permanecia acesa) (DAHLE, 2010).

**Figura 1 - Medidor de Samuel Gardiner.**

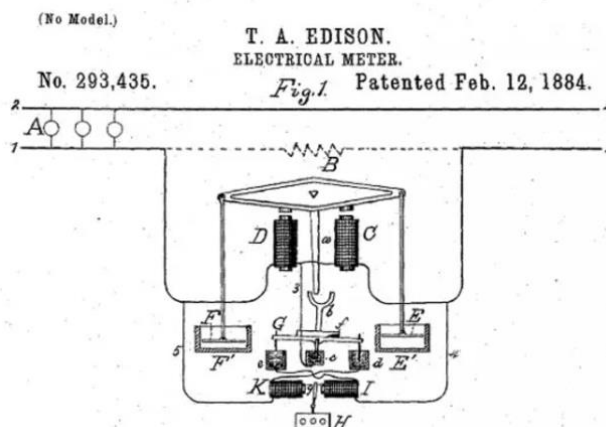


Fonte: Dahle, 2011.

Já o primeiro aparelho que operava em corrente alternada foi criado em 1878 por J. B. Fuller. O seu funcionamento era dado por um sistema de bobinas que vibravam de acordo com a frequência da onda da tensão da rede, assim acionando um conjunto de engrenagens que giravam um disco. Porém, esse medidor só era eficiente para cargas que não tinham variação de potência durante o período de medição.

Em 1884, Thomas Edson desenvolveu um medidor diferente dos inventados anteriormente. Seu medidor incorporava duas placas de zinco em um recipiente com sulfato de zinco ( $ZnSO_4$ ). Nesse processo, o zinco vai se depositando na placa do catodo de acordo com a passagem da corrente pelo circuito, enquanto que a massa da placa do anodo vai sendo reduzida. Uma equipe coletava mensalmente as placas do catodo e as pesava, era calculada então a tarifa do mês de acordo com a diferença do peso medido e o peso anterior.

**Figura 2 - Medidor de Thomas Edison.**

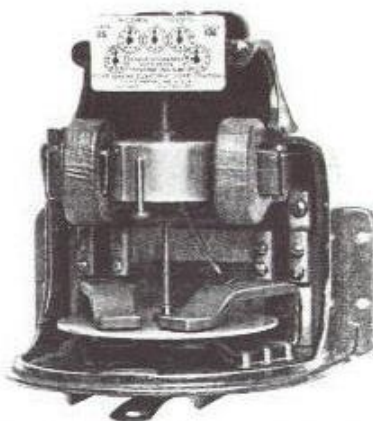


Fonte: Google Patents.

Na mesma década, mais precisamente em 1889, Elihu Thomsom, em conjunto com Thomas Duncan, apresentou um medidor que operava tanto em corrente alternada quanto em contínua. Esse foi considerado o primeiro instrumento medidor watt-hora, e usava o princípio da indução eletromagnética.

No ano de 1892, Duncan criou um medidor watt-hora eletromecânico que trabalhava com os mesmos princípios dos medidores atuais. Inicialmente foi projetado para medição de circuitos monofásicos, mas logo foram adaptados para a medição trifásica. A figura abaixo mostra o modelo do medidor de Duncan.

**Figura 3 - Medidor de Thomas Duncan.**



**Fonte: Dahle, 2011.**

Atualmente, os medidores de energia subdividem-se em duas categorias: os eletromecânicos, que utilizam os princípios dos medidores de Duncan, e os eletrônicos, que funcionam através de circuitos integrados. Os dois tipos serão abordados a seguir.

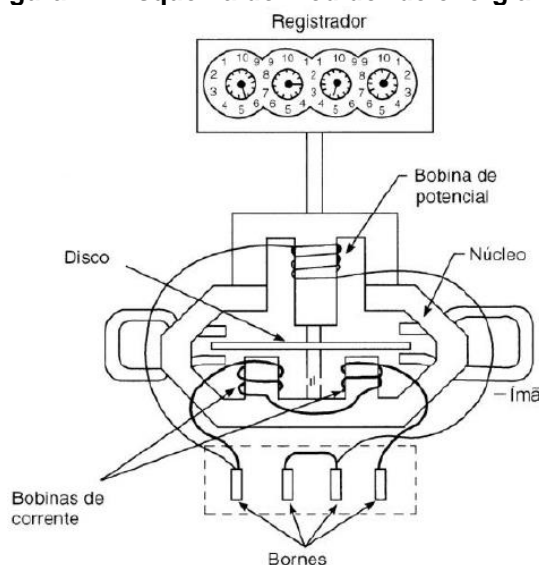
### 2.1.1 Medidores eletromecânicos

Mesmo com aperfeiçoamentos ocorridos ao longo do tempo, os medidores de energia eletromecânicos atuais utilizam o mesmo princípio de funcionamento desenvolvido por Duncan. Eles ainda são muito utilizados pela sua ampla disponibilidade no mercado.

O instrumento trabalha basicamente da mesma forma que um motor de indução. Para a medição são utilizadas bobinas de corrente, que conduz a corrente da linha, e bobina de potencial, que conduz uma corrente proporcional à tensão na

carga. Entre as duas bobinas está um disco de alumínio, onde uma corrente parasita criada pelos campos das bobinas faz com que ele gire em torno do seu próprio eixo. Ao girar, o disco tem uma velocidade angular proporcional à potência ativa que está sendo consumida (DAHLE, 2010). A Figura 4 representa o esquema de um medidor de energia eletromecânico.

**Figura 4 – Esquema do medidor de energia eletromecânico.**



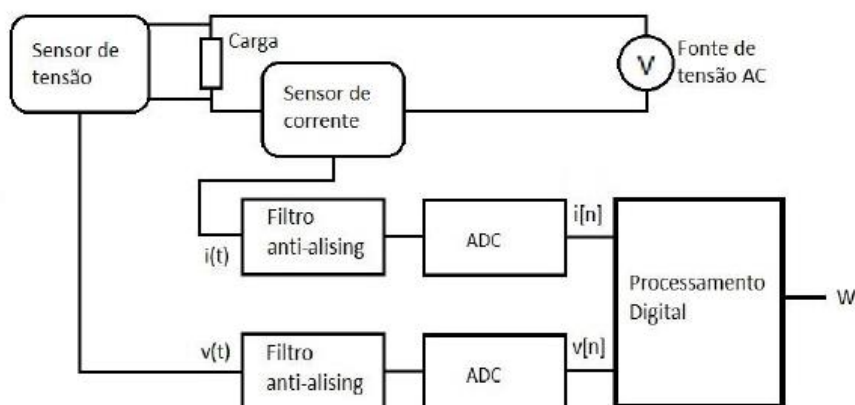
**Fonte: Dahle, 2011.**

### 2.1.2 Medidores eletrônicos

Atualmente, a medição de energia por medidores eletrônicos oferece melhor exatidão em relação aos medidores eletromecânicos, além do custo de aquisição estar cada vez mais acessível. Esse tipo de medidor tem funcionalidades que vão além de só medir o consumo de energia, como por exemplo a troca de informações em tempo real entre a concessionária e o consumidor e o monitoramento de qualidade de energia.

Com os medidores eletrônicos, o cliente pode ter acesso a várias informações tais como potência ativa, potência reativa, potência aparente, fator de potência, valores de tensão e corrente, além de poder acessar os registros de consumo, com informações de data e hora. A Figura 5 mostra o diagrama de blocos de um medidor eletrônico.

**Figura 5 - Diagrama de blocos de um medidor eletrônico.**



Fonte: Moro, 2013.

Conforme pode ser visto no diagrama de blocos, os medidores eletrônicos possuem sensores de corrente e tensão que são responsáveis por ler os sinais na carga. Eles então fornecem valores de tensão e corrente proporcionais aos da carga, que passam primeiro pelos filtros *anti-aliasing* e então seguem para os conversores analógico/digital (ADC). Os conversores amostram os sinais recebidos e os manda para a etapa de processamento digital, onde é calculada a potência ativa. Por fim, o valor da potência ativa é integrado no tempo para se obter a energia consumida pela carga. A Figura 6 ilustra um medidor de energia eletrônico.

**Figura 6 - Medidor eletrônico.**



Fonte: Energisa, 2013.

## 2.2 Potência

A potência é dita como a grandeza mais importante em sistemas de energia elétrica, pois compreende a transmissão de energia de um local a outro. Ela é uma grandeza física que é definida como a quantidade de energia gasta ou absorvida por unidade de tempo, podendo ser obtida através do produto entre a energia necessária para deslocar uma carga unitária através de um elemento, conhecida como tensão elétrica ou diferença de potencial, e o fluxo de carga por unidade de tempo, conhecido como corrente elétrica. A Equação 1 é a equação para cálculo da potência instantânea de um equipamento ligado à rede elétrica (ALEXANDER; SADIKU, 2013).

$$P = V * I \quad (1)$$

Onde:

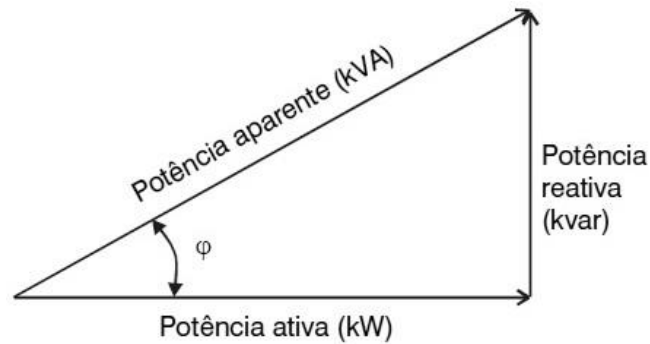
- $P$  é a potência elétrica instantânea dada em Watts (W);
- $V$  é a tensão elétrica dada em Volts (V);
- $I$  é a corrente elétrica dada em Ampères (A).

O watt é uma unidade de medida padrão do Sistema Internacional de Unidades (SI) e é equivalente à energia de um joule em cada segundo. As companhias de distribuição de energia elétrica medem a energia em watt-hora, correspondente a 3.600 Joules.

Quando um dispositivo tem impedância puramente resistiva, toda a energia consumida pelo mesmo é transformada em trabalho útil, essa é conhecida como potência ativa. Porém, existem aparelhos que possuem cargas com componentes reativas, que podem ser capacitivas ou indutivas. O surgimento dessas componentes provoca uma defasagem entre os ângulos da corrente e da tensão, fazendo com que em determinados momentos o sentido da corrente fique contrário à queda de tensão na carga, nessas ocasiões a carga fornece energia reativa à rede, conhecida como potência reativa.

Ao relacionar a potência ativa com a potência reativa, é possível montar um triângulo de potências, como visto na Figura 7, onde tem o aparecimento de uma terceira potência, conhecida como potência aparente. Além disso, é possível obter o fator de potência da razão entre a potência ativa pela potência aparente.

**Figura 7 – Triângulo das potências.**



Fonte: ENGELÉTRICA, 2011.

### 2.2.1 Potência ativa

Também conhecida como potência média ou real, a potência ativa corresponde à parcela da potência que é convertida efetivamente em trabalho, gerando calor, luz ou movimento e é expressa em Watts (W) (ALEXANDER; SADIKU, 2013). Assim, é possível definir a potência ativa com a Equação 2:

$$P = \frac{1}{T} \int_0^T P_{inst}(t) dt \quad (2)$$

Onde:

- $P$  é a potência ativa dada em Watts (W);
- $P_{inst}$  é a potência instantânea em Watts (W);
- $T$  é o período de tempo dado em segundos (s).

De forma alternativa, a potência aparente pode ser definida como o produto entre a corrente e a tensão, levando em consideração suas respectivas fases:

$$P = V * I * \cos(\theta_v - \theta_i) \quad (3)$$

Onde:

- $P$  é a potência ativa dada em Watts (W);
- $V$  é a tensão elétrica dada em Volts (V);
- $I$  é a corrente elétrica dada em Ampères (A);

- $\theta_v$  é o ângulo de fase da tensão elétrica dado em graus;
- $\theta_i$  é o ângulo de fase da corrente elétrica dado em graus.

Ainda, com base no triângulo das potências, pode-se escrever a potência ativa como o produto entre o módulo da potência aparente e o fator de potência:

$$P = |S| * \cos(\varphi) \quad (4)$$

Onde:

- $P$  é a potência ativa dada em Watts (W);
- $S$  é a potência aparente dada em (VA);
- $\cos(\varphi)$  é o fator de potência.

### 2.2.2 Potência reativa

Diz-se da potência reativa como a potência que não realiza trabalho útil, ela circula no sistema entre o gerador e a carga, e seu sentido muda dependendo do tipo de carga. Ainda, a potência reativa pode ser armazenada no campo magnético de um indutor ou no campo elétrico de um capacitor. Além disso, ela tem o seu valor médio nulo, trocando apenas potência instantânea com a fonte geradora, o que não é interessante. O seu valor pode ser calculado através do triângulo das potências:

$$Q = |S| * \sin(\varphi) \quad (5)$$

Onde:

- $Q$  é a potência reativa dada em Volt-Ampère Reativo (VAr);
- $S$  é a potência aparente dada em (VA);
- $\varphi$  é o ângulo do fator de potência dado em graus.

Assim como pode ser obtido através do produto entre a tensão e corrente elétrica, levando em consideração as suas respectivas fases:

$$Q = V * I * \sin(\theta_v - \theta_i) \quad (6)$$



Onde:

- $Q$  é a potência reativa dada Volt-Ampère Reativo (VAr);
- $V$  é a tensão elétrica dada em Volts (V);
- $I$  é a corrente elétrica dada em Ampères (A);
- $\theta_v$  é o ângulo de fase da tensão elétrica dado em graus;
- $\theta_i$  é o ângulo de fase da corrente elétrica dado em graus.

### 2.2.3 Potência aparente

A potência aparente é o vetor resultante no triângulo das potências, sendo composta pelas potências ativa e reativa. Seu valor é aquele que está efetivamente disponibilizado na rede, sendo assim utilizado no dimensionamento de alguns equipamentos elétricos, cabos e sistema de proteção de instalações elétricas. O módulo da potência aparente pode ser obtido através do produto entre os valores eficazes de tensão e de corrente elétrica.

$$|S| = V_{ef} * I_{ef} \quad (7)$$

Onde:

- $S$  é a potência aparente dada Volt-Ampère (VA);
- $V_{ef}$  é a tensão elétrica eficaz dada em Volts (V);
- $I_{ef}$  é a corrente elétrica eficaz dada em Ampères (A).

Assim como as outras potências, é possível obter o valor da potência aparente através das relações do triângulo das potências:

$$|S| = \sqrt{P^2 + Q^2} \quad (8)$$

Onde:

- $S$  é a potência aparente dada Volt-Ampère (VA);
- $P$  é a potência ativa dada em Watts (W);
- $Q$  é a potência reativa dada em Volt-Ampère Reativo (VAr).

#### 2.2.4 Fator de potência

Em certos equipamentos, tais como transformadores, motores elétricos, dentre outros, há a necessidade de uma certa quantidade de energia reativa para o seu funcionamento. Alguns desses aparelhos consumidores, normalmente os dotados de bobinas, apresentam um consumo de energia reativa indutiva para a formação de campos magnéticos, causando uma alteração no fator de potência. Análogo a esses, existem outros aparelhos que fornecem energia reativa capacitiva para a formação de campos elétricos, o que também interfere no fator de potência (ELEKTRO, 2014).

Por esses motivos, o fator de potência é utilizado como índice de qualidade de energia e é de extrema importância para a área de eficiência energética. No Brasil, o FP (fator de potência) mínimo permitido é de 0,92, tendo leis que multam consumidores caso o mesmo esteja abaixo desse valor (ANEEL, 2012).

O fator de potência tem como definição a razão entre a potência ativa e aparente, sendo o resultado um valor adimensional que varia entre 0 e 1. Ele é determinado pelo tipo de carga, sendo que quando o FP é 1, toda a energia fornecida pela fonte geradora é consumida pelo equipamento, isso acontece em aparelhos puramente resistivos, tais como ferro de passar roupas, chuveiros, etc. Quanto mais próximo de 1, melhor é a qualidade da energia. Por outro lado, aparelhos que possuem motores de indução, lâmpadas fluorescentes e computadores, entre outros, fazem com que o FP decaia, se aproximando de 0. Como consequência disso, são exigidos maiores condutores na transmissão e transformadores de capacidade mais elevada, além de provocar quedas de tensão no sistema de distribuição. A Equação 9 serve para o cálculo do FP:

$$FP = \frac{P}{S} \quad (9)$$

Onde:

- FP é o fator de potência;
- $P$  é a potência ativa dada em Watts (W);
- $S$  é a potência aparente dada Volt-Ampère (VA).

### 2.3 Estrutura tarifária

Ao mesmo tempo que novas usinas são construídas, também se faz necessária a construção de novas linhas de transmissão, assim como a melhoria contínua e uma boa manutenção nas linhas já existentes. Com isso, as tarifas visam assegurar às empresas prestadoras dos serviços receita suficiente para cobrir custos operacionais e investimentos necessários para expandir a capacidade e garantir o atendimento com qualidade. Os custos e investimentos repassados às tarifas são calculados pelo órgão regulador (ANEEL, 2015).

De maneira a manter e progredir com a continuidade de serviço de fornecimento de energia elétrica com a devida qualidade, a distribuidora possui custos que devem ser repassados aos seus consumidores através de tarifas de utilização. A composição da tarifa é definida através de três categorias distintas, sendo elas: geração de energia, custos de transporte da energia nos sistemas de transmissão e distribuição e encargos setoriais.

Além dos custos da companhia de distribuição são adicionados impostos Federais, Estaduais e Municipais, sendo, respectivamente, PIS (1,65%), COFINS (7,6%), ICMS (25%) e a contribuição para iluminação pública.

A partir de 2004, a compra de energia por parte das companhias distribuidoras passou a ser determinante através de leilões públicos, visando a compra com os menores preços.

Apesar de existirem várias empresas responsáveis pelo transporte da energia, a competição nesse segmento não traz ganhos econômicos. Portanto, a ANEEL atua na fiscalização de modo que as tarifas sejam compostas por custos eficientes, relacionados aos serviços prestados. O setor de transporte é dividido em dois segmentos, transmissão e distribuição, respectivamente, transporte da geração até a distribuição e a destruição à unidade consumidora final.

Os encargos setoriais e os tributos associados a tarifa de energia são instituídos por leis, sendo alguns incidentes sobre o custo da distribuição, enquanto outros presentes nos custos de geração e de transmissão.

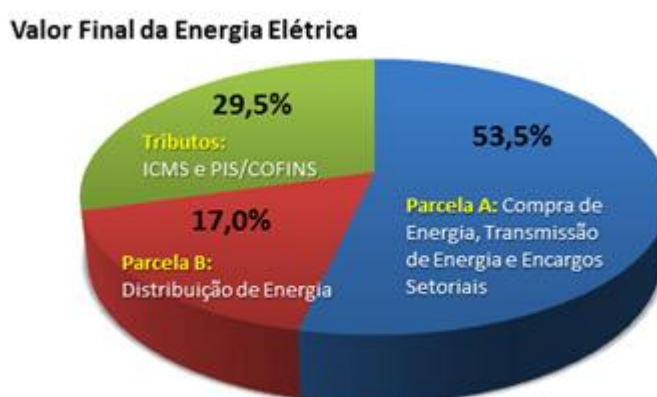
Ao consumidor final o pagamento da fatura final engloba a compra da energia, custos do gerador, a transmissão, custos da transmissora, e pela distribuição, custos dos serviços prestados pela distribuidora, com a adição dos encargos setoriais e tributos.

Para determinação do cálculo tarifário, os custos da distribuidora são classificados em dois tipos:

- Parcela A: Compra de energia, transmissão e encargos setoriais;
- Parcela B: Distribuição de energia.

Na Figura 8 é visível a distribuição percentual dos gastos na tarifa de energia elétrica do consumidor, os custos da energia representam a maior parcela (53,5%), seguido dos custos com tributos (29,5%) e os custos com a distribuição de (17%).

**Figura 8 – Composição da tarifa energética.**



Fonte: ANEEL, 2015.

### 2.3.1 Grupo de consumidores

A ANEEL divide os consumidores, de acordo com a demanda requerida, em dois grandes grupos, Grupo A e Grupo B, cada um contendo subdivisões como pode ser visto na Tabela 1.

**Tabela 1 - Categorias de consumidores**

<b>Subgrupo</b>	<b>Tensão de fornecimento</b>
A1	≥ 230kV
A2	88kV a 138kV
A3	69kV
A3a	30kV a 44kV
A4	2,3kV a 25kV
AS (sistemas subterrâneos de distribuição)	≤ 2,3kV
B1 (residencial)	≤ 2,3kV
B2 (rural)	≤ 2,3kV
B3 (demais classes)	≤ 2,3kV
B4 (iluminação pública)	≤ 2,3kV

Fonte: ANEEL, 2012.

O grupo que terá destaque para o desenvolvimento desse trabalho é o grupo B e subgrupo B1, que compreende a classe residencial.

### 2.3.2 Bandeiras tarifárias

O sistema de bandeiras tarifárias foi aplicado, a partir de 2015, nas contas de energia dos consumidores. O principal motivo da criação dessa tarifação foi o aumento de encargo setorial em função do custo de produção de energia termoelétrica. Todo mês as condições de operação do sistema são reavaliadas pelo Operador Nacional do Sistema Elétrico (ONS), e dependendo da geração térmica mais cara em operação, é definida a bandeira a ser aplicada no próximo mês. O sistema complementa o atual modelo de cobrança de consumo e é composto por três bandeiras que indicam se a energia custa mais ou menos, em função das condições de geração de eletricidade. As bandeiras são (ELETROBRAS, 2017):

- 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\$ 0,010 para cada quilowatt-hora (kWh) consumidos;
- Bandeira vermelha - Patamar 1: condições mais custosas de geração. A tarifa sofre acréscimo de R\$ 0,030 para cada quilowatt-hora kWh consumido.

- Bandeira vermelha - Patamar 2: condições ainda mais custosas de geração. A tarifa sofre acréscimo de R\$ 0,050 para cada quilowatt-hora kWh consumido.

Além dessas bandeiras tarifárias, a ANEEL criou a tarifa branca, uma nova modalidade para consumidores de baixa tensão (127, 220, 380 ou 440 Volts), que fazem parte do grupo B1. Trata-se de uma tarifa que sinaliza aos consumidores a variação do valor da energia conforme o dia e o horário do consumo (ANEEL, 2016).

Com essa nova modalidade, os consumidores passam a ter a possibilidade de pagar diferentes valores de energia em função da hora e do dia da semana em que consomem. Se na residência são adotados hábitos que priorizem o uso da energia elétrica nos períodos fora dos horários de pico (manhã, início da tarde e madrugada, por exemplo), a opção pela tarifa branca oferece a oportunidade de reduzir os gastos com energia. Nos dias úteis, a tarifa branca tem três valores: ponta, intermediário e fora de ponta. Esses períodos são estabelecidos pela ANEEL e são diferentes para cada distribuidora (ANEEL, 2017).

A tarifa branca incentiva alguns consumidores a migrarem o uso dos equipamentos elétricos nos períodos de ponta para aqueles em que a rede de distribuição de energia elétrica tem capacidade ociosa. Isso faz com que seja reduzida a necessidade de expandir a rede elétrica, diminuindo como consequência alguns impactos causados pela construção da mesma.

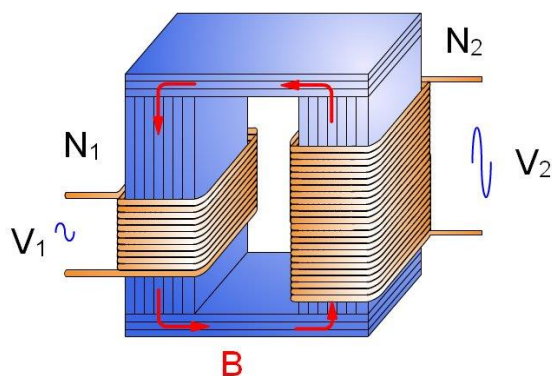
## **2.4 Transformadores**

O transformador é um dispositivo elétrico que possui a propriedade de converter, pela ação de um campo magnético, energia elétrica CA de um nível de tensão em energia elétrica CA de outro nível de tensão, ambas com a mesma frequência. O seu funcionamento é baseado no princípio da indução mútua, o qual diz que quando dois ou mais indutores estiverem próximos uns dos outros, o fluxo magnético provocado pela circulação de corrente elétrica em uma das bobinas se associa com as outras bobinas, induzindo tensão nessas últimas (ALEXANDER; SADIKU, 2013).

A estrutura física de um transformador é formada basicamente por duas bobinas de fios de cobre revestidos de esmalte, enroladas em torno de um núcleo

ferromagnético (existem alguns casos especiais em que o núcleo possui mais de duas bobinas enroladas), sendo que não há conexão física entre as bobinas, a única ligação entre elas é o fluxo magnético comum presente dentro do núcleo (CHAPMAN, 2013). A Figura 9 mostra a estrutura básica de um transformador ideal.

**Figura 9 - Estrutura básica de um transformador ideal.**



Fonte: CEBALLOS, 2013.

A enrolamento do lado esquerdo da figura acima é chamada de primário, nele é ligada a fonte de energia elétrica. Já o enrolamento do lado direito é denominado secundário, onde se conecta a carga. Os transformadores podem ser abaixadores ou elevadores de tensão, no caso da imagem apresentada a tensão está sendo elevada, sendo determinado pela relação de transformação. Como é visível, existe uma relação direta entre o número de espiras da bobina com a tensão, já a relação das correntes é inversamente proporcional. A Equação 10 apresenta as relações de transformação:

$$a = \frac{N1}{N2} = \frac{V1}{V2} = \frac{I2}{I1} \quad (10)$$

Onde:

- $a$  é relação de transformação;
- $N1$  é o número de espiras no enrolamento primário;
- $N2$  é o número de espiras no enrolamento secundário;
- $V1$  é a tensão elétrica no primário;
- $V2$  é a tensão elétrica no secundário;
- $I1$  é a corrente elétrica no primário;
- $I2$  é a corrente elétrica no secundário.

### 2.4.1 Transformadores de potência

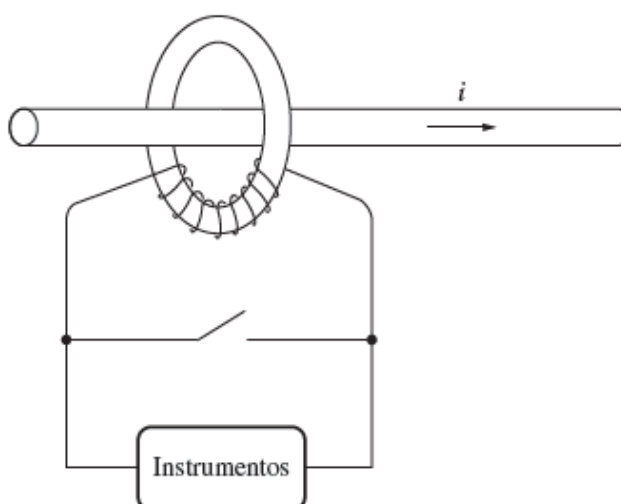
Como anteriormente citado, a principal finalidade de um transformador é a conversão do nível de tensão elétrica CA, sempre mantendo a mesma frequência. Porém, os transformadores também são utilizados para outros propósitos, como por exemplo, amostragem de tensão, amostragem de corrente e transformação de impedância (CHAPMAN, 2013).

O transformador de potência (TP) ou de potencial, possui bobinas especialmente enroladas para que o seu primário seja o lado de alta tensão com um secundário de baixa tensão. Sua potência nominal é baixa, tendo como única finalidade amostrar a tensão do sistema de potência para os instrumentos que irão monitorá-lo. Para isso, ele deve ser muito exato a fim de não distorcer os valores de tensão.

### 2.4.2 Transformadores de corrente

Um TC (transformador de corrente) é um equipamento que faz a redução do nível da corrente que flui em um fio, fazendo que esse seja seguro o suficiente para a realização de medidas. O secundário desse tipo de transformador consiste em uma bobina enrolada em torno de um material ferromagnético, em forma de anel, sendo o primário o fio que passa pela parte central do anel (CHAPMAN, 2013). É possível ver na Figura 10 o esquema de um transformador de corrente.

**Figura 10 – Esquema de um TC.**



**Fonte: CHAPMAN, 2013.**



Uma parte do fluxo magnético proveniente do primário (o fio) é concentrada no material ferromagnético (o anel). Esse fluxo então induz corrente e tensão na bobina do secundário.

## 2.5 Comunicação SPI

O SPI (*Serial Peripheral Interface*) é um protocolo de comunicação utilizado para conectar dispositivos a fim de que troquem informações entre si. Esse tipo de protocolo é amplamente utilizado na comunicação entre chips ou CI (circuitos integrados), sendo uma importante ferramenta de transmissão de dados em sistemas digitais que possuem conversores ADC, memórias EEPROM e *flash*, sensores com interface de comunicação, cartões de memória, entre outros (SACCO, 2014).

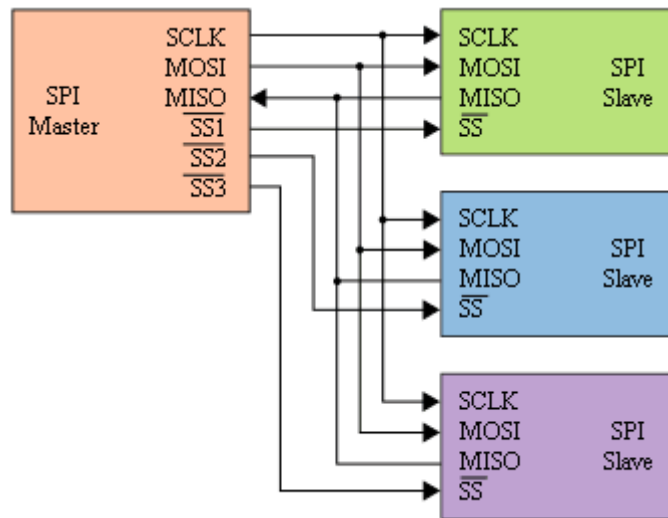
Consiste em um barramento de comunicação série síncrona que permite a troca de dados em modo *full-duplex*, isto é, cada componente executando o protocolo pode receber e transmitir dados ao mesmo tempo. Os dispositivos se comunicam segundo um modelo de cooperação do tipo mestre-escravo, sendo que a comunicação é sempre iniciada pelo mestre. Os pinos básicos de comunicação entre dispositivos SPI e o esquema padrão de ligação são dados conforme abaixo:

**Tabela 2 – Pinos da comunicação SPI.**

Pino	Nome Padrão	Descrição	Nomes Alternativos
Mestre	MOSI	Dados enviados pelo mestre	SDO, DO, SO
Escravo	MISO	Dados recebidos pelo mestre	SDI, DI, DI
Clock	SCLK	Sinal de relógio gerado pelo mestre	SCK, CLK
Chip select	SS	Seleção do escravo que vai comunicar	CS, nSS, nCS

Fonte: SACCO, 2014.

**Figura 11 – Esquema de ligação para o protocolo SPI.**



Fonte: SACCO, 2014.

Assim como pode ser visto na Figura 11, a comunicação através do protocolo SPI requer a utilização de 4 fios. O mestre seleciona o escravo com qual irá se comunicar através do pino SS. Cada escravo possui uma ligação única com o mestre, sendo que os escravos não se comunicam entre si. Através do controle do *clock*, o mestre decide quando o dado é enviado ou recebido por ele.

É possível citar como vantagens e desvantagens desse tipo de comunicação:

Vantagens:

- É mais rápida que uma comunicação serial assíncrona;
- O mestre pode suportar vários escravos;
- Permite comunicação *full-duplex*.

Desvantagens:

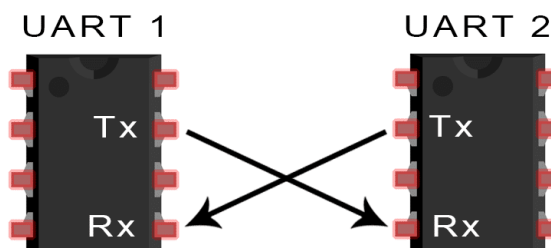
- Exige mais conexões que outros métodos de comunicação;
- Toda comunicação deve partir do mestre, e os escravos não podem trocar dados entre si;
- Se acontecer alguma falha no mestre, essa irá comprometer todo o sistema de comunicação.

## 2.6 Comunicação USART

*Universal Synchronous Asynchronous Receiver Transmitter*, (USART) ou Transmissor/Receptor Universal Síncrono e Assíncrono, é um tipo de comunicação de dados serial que se pode trabalhar de duas maneiras: síncrona ou assíncrona. Na forma síncrona, uma ponta é mestre e a outra escravo. Um fio é utilizado para a transmissão de dados, em regime *half-duplex*, ou seja, nos dois sentidos, mas um sentido de cada vez. O outro fio é usado para pulsos de *clock* emitidos pelo dispositivo mestre (CIRCUIT BASICS, 2016).

Já na forma assíncrona, dois fios são usados para a transmissão de dados (RX/TX), um em cada direção, em regime *full-duplex*, ou seja, bidirecional. Para isso, cada dispositivo deve ter seu *clock*, e as velocidades devem ser iguais. O esquema de ligação para esse tipo de comunicação pode ser visto na Figura 12.

Figura 12 – Esquema de ligação para comunicação UART.



Fonte: CIRCUIT BASICS, 2016.

Um pacote de *bits* é enviado do pino de transmissão (Tx) ao pino receptor (Rx). Cada pacote enviado contém 1 *start bit* que indica o início da mensagem, 1 ou 2 *stop bits* para indicar o final da mensagem, 5 a 9 *bits* de informação e 1 *bit* de paridade para evitar a recepção de erros.

## 2.7 Wi-Fi

Conhecido como *Wi-Fi*, o padrão IEEE 802.11 é um conjunto de especificações que descreve as características de redes locais sem fio WLAN (*Wireless Local Area Network*). O nome é a abreviatura de “*Wireless Fidelity*”, certificação emitida pela *Wi-Fi Alliance*, entidade responsável pelo licenciamento de produtos que possuem essa tecnologia (MUXFELDT, 2017).

Na prática, o *Wi-Fi* possibilita a implementação de redes que conectam computadores, smartphones, tablets, videogames e outros dispositivos compatíveis com esse tipo de tecnologia. A transmissão destas redes é efetuada através de ondas de radiofrequência, o que oferece vantagens como: permitir que o usuário utilize a rede em qualquer ponto da área de alcance da transmissão e evitar a adaptação ou avaria de estruturas prediais para a passagem de fios.

O padrão IEEE 802.11, também conhecido como 802.11 *legacy*, foi criado originalmente em 1997. Esse padrão opera numa faixa de frequência entre 2,4 GHz e 2,4832 GHz e sua taxa de transmissão de dados é de 1 Mb/s ou 2Mb/s. Com o aprimoramento da tecnologia, outras versões foram criadas. As versões existentes podem ser vistas na Tabela 3.

**Tabela 3 – Versões do padrão IEEE 802.11.**

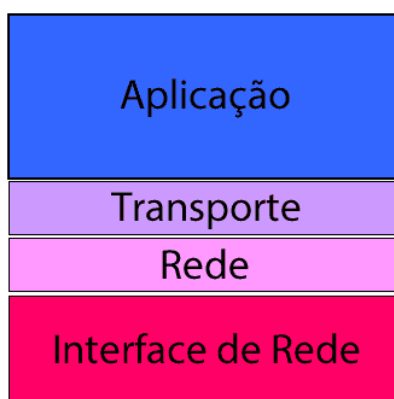
Nome da norma	Nome	Descrição
802.11a	Wi-Fi5	A norma 802.11a (chamada Wi-Fi 5) especifica 8 canais de rádio na banda de frequência dos 5 GHz.
802.11b	Wi-Fi	A norma 802.11b é, atualmente, a mais usada. Possui um alcance que pode ir até 300 metros num espaço aberto. A frequência utilizada é a banda dos 2.4 GHz, com 3 canais de rádio disponíveis.
802.11c	Uma ponte 802.11 para 802.1d	A norma 802.11c não interessa para o público em geral. Ela é apenas uma modificação da norma 802.1d a fim de poder estabelecer uma ponte com as tramas 802.11 (nível de ligação de dados).
802.11d	Internacionalização	A norma 802.11d é um complemento à norma 802.11, cujo objetivo é permitir uma utilização internacional das redes locais 802.11.
802.11e	Melhorias na qualidade	A norma 802.11e tem como objetivo definir as necessidades dos diferentes pacotes em termos de banda larga e prazo de transmissão.
802.11f	Roaming	A norma 802.11f é uma recomendação para vendedores de ponto de acesso para uma melhor interoperabilidade dos produtos. Ela propõe o protocolo <i>Inter-Access point roaming protocol</i> que permite a um usuário itinerante mudar de ponto de acesso de maneira transparente durante uma viagem. Esta possibilidade é chamada de <i>Roaming</i>
802.11g		A norma 802.11g oferece um débito elevado (54 Mbps teórico, 30 Mbps real) na banda de frequência dos 2.4 GHz.
802.11h		A norma 802.11h visa aproximar a norma 802.11 do padrão europeu (HiperLAN 2) e ficar em conformidade com o regulamento europeu.
802.11i		A norma 802.11i tem como objetivo melhorar a segurança das transmissões (gestão e distribuição das chaves, codificação e autenticação).
802.11r		A norma 802.11r foi elaborada de modo a utilizar sinais infravermelhos.
802.11j		A norma 802.11j visa aproximar a norma 802.11 para o regulamento japonês.

Fonte: MUXFELDT, 2017.

## 2.8 Protocolo TCP-IP

O TCP-IP é um conjunto de protocolos de comunicação que tem como objetivo padronizar todas as comunicações da rede de *internet*. O nome se refere a dois protocolos: TCP (*Transmission Control Protocol*) e o IP (*Internet Protocol*). Além disso, existem outros protocolos relacionados ao TCP-IP, tais como: FTP (*File Transfer Protocol*), HTTP (*Hypertext Transfer Protocol*), UDP (*User Datagram Protocol*) e SMTP (*Simple Mail Transfer Protocol*) (INFONOVA, 2018).

Figura 13 – Arquitetura do TCP-IP.



TCP-IP

Fonte: INFONOVA, 2018.

Como é possível observar na Figura 13, o conjunto de protocolo TCP-IP possui quatro camadas. Na camada de aplicação encontram-se os protocolos como o SMTP (para e-mail), FTP (para transferência de arquivos) e HTTP (para navegar na internet), que comunicam diretamente com o software para a identificação do tipo de requisição realizada. Após processar a requisição, a informação é repassada para a camada de transporte, usualmente o TCP. Esta camada tem como função dividir em pacotes os dados recebidos da camada de aplicação, a fim de enviá-los para a próxima camada. Além disso, nessa camada é verificado se o pacote alcançou o seu destino e se os dados chegaram de maneira íntegra. Na camada de rede, o IP adiciona um endereço virtual (o endereço do computador que está enviando) nos pacotes recebidos da camada de transporte, chamado de endereço IP. Em seguida, o pacote é então enviado para a próxima camada. A interface de rede é responsável por pegar os pacotes recebidos da camada de rede e enviar os mesmos através da

rede. O tipo de rede mais utilizado para comunicação entre computadores é a Ethernet (que é encontrada em diferentes faixas de velocidade) e pode ser cabeada ou *Wi-Fi*.

## 2.9 Internet

Originalmente chamada ARPANET, a *internet* era uma rede militar americana criada para resistir aos ataques caso um ponto da rede fosse destruído, assim as informações continuariam a circular. Este sistema é utilizado até hoje: quando uma informação é enviada ou recebida pela *internet*, seus pacotes de dados passam por centenas de computadores e seguem caminhos diferentes, como uma teia de aranha. Esta "teia de aranha" deu origem à palavra "web" e ao "*World Wide Web*" (termo utilizado para denominar páginas HTML) (NOURIA, 2016).

A *internet* interliga todos os computadores do mundo. São os protocolos TCP/IP que a fazem tão poderosa. Todos os dispositivos que entendem essa linguagem são capazes de trocar informações entre si.

A aplicação mais conhecida da *internet* é o HTTP (são as páginas *web* vistas no *browser*). O protocolo HTTP utiliza-se dela para transportar páginas HTML, imagens, músicas, vídeos, entre outros tipos de arquivos.

Contudo, a *internet* não se limita somente às páginas *web*. Existem inúmeros protocolos que possibilitam a criação de variados tipos de aplicações. Alguns projetos em desenvolvimento que têm como objetivo conectar todos os aparelhos pela *internet*, tais como: geladeira, fogão, sistema de aquecimento, sistema de alarme e monitoramento, registro de energia.

## 3. MATERIAIS E MÉTODOS

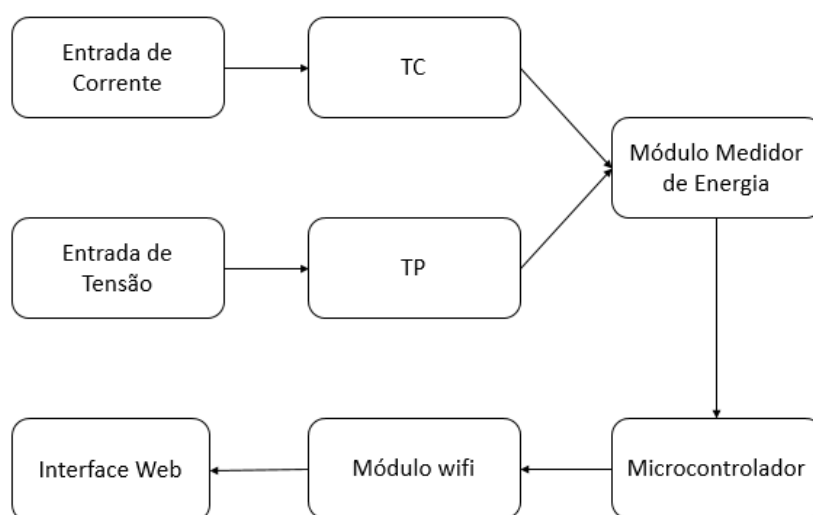
Neste capítulo serão apresentados os materiais utilizados para a estruturação e desenvolvimento do projeto, abordando as suas especificações técnicas, seus funcionamentos e as razões pelas quais eles foram escolhidos.

O mesmo foi dividido em dois tópicos: *hardware* e *software*, para melhor organização e entendimento.

### 3.1 Hardware

Para a montagem do protótipo, é necessária a utilização de equipamentos de *hardware* como, por exemplo, transformadores de corrente e potência, módulo medidor de energia, microcontrolador e um módulo para o envio de informações via rede *Wi-Fi*. Foram então analisados alguns dispositivos que poderiam ser utilizados nesse projeto. Os equipamentos que foram escolhidos, por melhor se adequarem ao projeto, serão apresentados a seguir. Na Figura 14 é possível ver o diagrama de blocos desenhado para a construção desse sistema.

**Figura 14 – Diagrama de blocos do hardware.**



**Fonte: Autoria própria.**

#### 3.1.1 Sensor de corrente SCT 013-050

Com o propósito de adequar a intensidade de corrente proveniente da entrada de energia para o medidor, foi escolhido o sensor de corrente SCT013-050 da Speed Technology Co. Esses são sensores conhecidos como transformadores de corrente e são usados para a medição de corrente alternada. É possível ver na Figura 15 um sensor desse tipo.

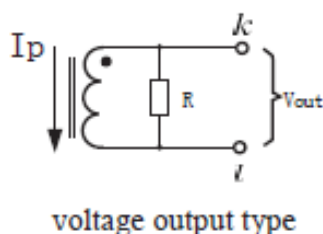
**Figura 15 – Sensor de corrente SCT 013-050.**



**Fonte: YHDC, 2015.**

Seu funcionamento é similar a um alicate amperímetro, sendo essa uma característica levada em consideração na hora da escolha, pois esse isola eletricamente os circuitos a serem medidos do circuito do protótipo, dando uma certa segurança caso ocorram variações bruscas de corrente, e também pelo fato de fazer leituras não invasivas. Esse sensor de corrente comporta até 50A de corrente no primário, com uma saída de 1V para o valor de fundo de escala, graças a um resistor *built-in* na saída, como pode ser visto na Figura 16, sendo uma característica essencial para a próxima etapa de condicionamento, pois o medidor de energia faz leituras de sinais de tensão. Além das características acima citadas, o sensor suporta temperaturas que variam entre  $-25^{\circ}\text{C}$  até  $70^{\circ}\text{C}$ , possui uma precisão de  $\pm 1\%$  com uma linearidade  $\leq 0.2\%$ .

**Figura 16 – Esquemático da saída do sensor.**



**Fonte: YHDC, 2015.**

Porém, para que não ocorra sobretensão na entrada do medidor de energia e uma possível queima do mesmo, foi necessário colocar um circuito divisor de tensão entre a saída do sensor e a entrada do módulo medidor de energia, pois este comporta



no máximo uma tensão de pico de +0,5V. Foi então utilizada a Equação 11 para calcular valor de pico da tensão de saída do sensor de corrente, sabendo que o seu valor RMS é de 1V:

$$V_{pIN} = V_{RMS} * \sqrt{2} = 1 * \sqrt{2} = 1,41V \quad (11)$$

Com esse valor em mãos, foi possível calcular o valor das resistências do circuito divisor de tensão, com o uso da Equação 12:

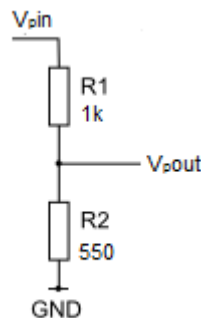
$$V_{pOUT} = \frac{R2}{R1+R2} * V_{pIN} \quad (12)$$

Depois de algumas manipulações e substituindo os valores de  $V_{pin}$  e  $V_{pout}$ :

$$R2 = 0,55 * R1 \quad (13)$$

A partir do resultado, foi escolhido um valor de  $R1 = 1k\Omega$  e conseqüentemente um valor de  $R2 = 550\Omega$ . Na Figura 17 é possível ver o circuito montado com os valores calculados:

**Figura 17 – Circuito divisor de tensão.**



**Fonte: Autoria própria.**

### 3.1.2 Sensor de tensão

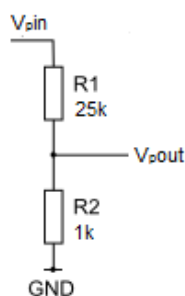
Seguindo a mesma linha de pensamento do sensor de corrente, para a tomada das amostras de tensão foi montado um circuito com o objetivo de condicionar o sinal para que este seja processado pelo medidor de energia. Primeiramente, na entrada do circuito foi colocado um transformador de 220/110V AC (primário) que abaixa a tensão para 9V (secundário). O transformador escolhido pode ser visto na Figura 18.

**Figura 18 – Transformador 110/220V para 9V.**



**Fonte: Autoria própria.**

Em seguida, foi calculado um divisor de tensão com o intuito de manter na saída deste uma tensão menor do que +0,5V de pico. Sabendo que a tensão do secundário do transformador é de 9V RMS, foi então utilizada a Equação 11 para calcular o seu valor de pico, obtendo-se um valor de  $V_{pin} = 12,7V$ . Em seguida, foi utilizada a Equação 12 para o cálculo das resistências do circuito, sendo encontrado que R1 tem que ser 25 vezes maior que R2. A partir do resultado, foi escolhido um valor de  $R2 = 1k\Omega$  e um valor de  $R1 = 25k\Omega$ . A Figura 19 mostra o circuito obtido:

**Figura 19 – Circuito divisor de tensão.**

Fonte: Autoria própria.

### 3.1.3 Módulo medidor de energia

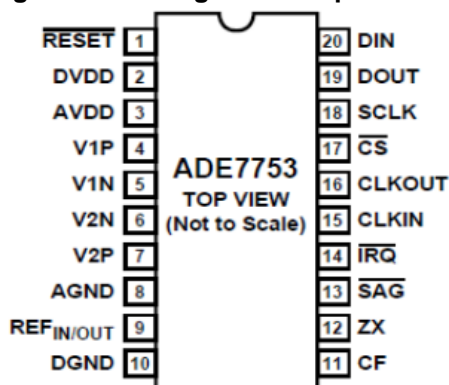
Com a evolução dos circuitos integrados (CI) que efetuam a medição de energia elétrica, estes componentes tornaram-se mais acessíveis e seu uso mais simplificado. Um exemplo destes dispositivos é o *chip* medidor ADE7753 da Analog Device, visto na Figura 20.

**Figura 20 – Chip medidor ADE7753.**

Fonte: Analog Devices, 2016.

Trata-se de um CI medidor de energia elétrica para circuitos monofásicos, que faz internamente todo o processamento dos sinais de entrada e fornece na saída as informações referentes ao consumo de energia ativa, reativa e aparente. Possui um erro de medição por volta de 0,1%, dispõe de saídas digitais para calibração, que é feita por meio de registradores através da comunicação SPI com o envio de valores a serem alterados para calibração de tensão, corrente e pulsos de energia. A Figura 21 apresenta a pinagem do ADE7753.

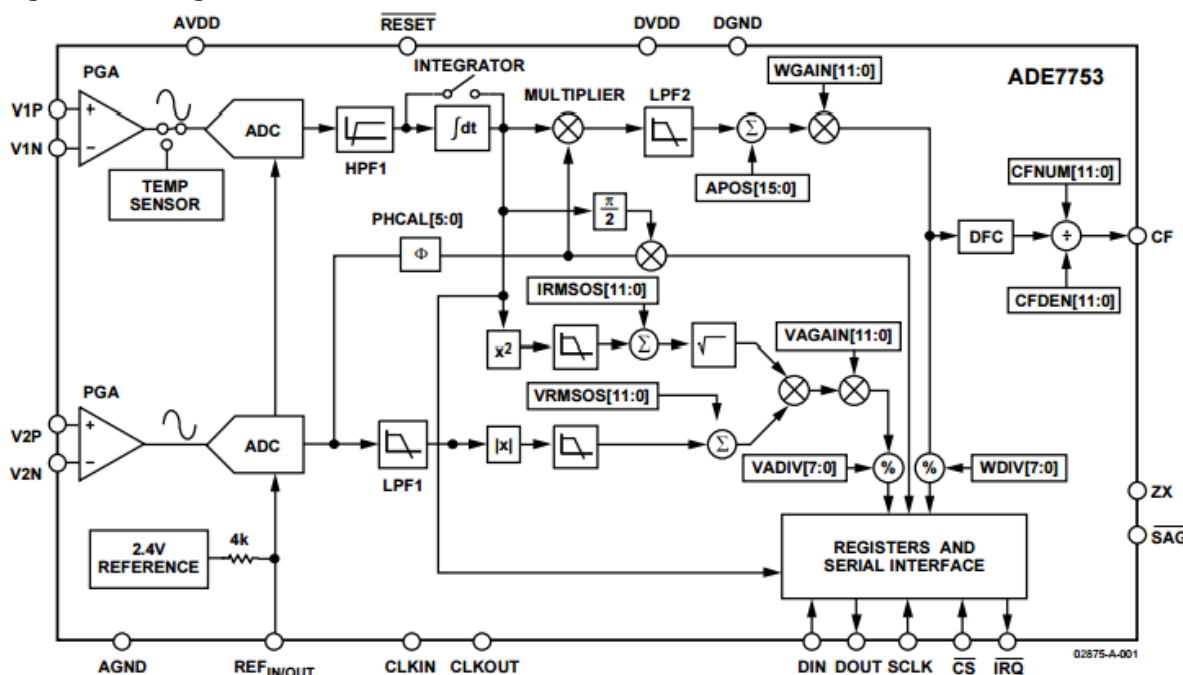
Figura 21 – Pinagem do chip medidor de energia.



Fonte: Analog Devices, 2016.

Na Figura 22 é possível ver o diagrama de blocos do medidor, onde contém todas as etapas do processamento dos valores. Os sinais instantâneos de tensão e corrente, provenientes dos TPs e TCs e devidamente condicionados para não exceder o valor de  $\pm 0,5V$ , são conectados nas entradas V1P/V1N e V2P/V2N, respectivamente. Eles então são amostrados através dos conversores analógico-digital presentes no medidor. Após amostrados, os valores digitais de tensão e de corrente são multiplicados para a obtenção do valor da potência. Por fim, a potência é integrada no tempo para se obter a energia consumida e os dados são gravados nos seus registradores.

Figura 22 – Diagrama de blocos medidor monofásico ADE7753.



Fonte: Analog Devices, 2016.

### 3.1.4 Microcontrolador

A fim de fazer integração e comunicação de todos os componentes do *hardware*, foi escolhido o microcontrolador PIC produzido pela Microchip Technology Inc. Esse microcontrolador é um componente integrado que em um único dispositivo contém todos os circuitos necessários para realizar um completo sistema programável.

Uma vantagem desse PIC é que todos os modelos de sua família possuem instruções bem parecidas, além de manter muitas semelhanças entre suas características básicas. O uso desse tipo de microcontrolador torna a migração para outros modelos da mesma família muito mais simples.

O microcontrolador da família PIC utilizado nesse projeto é o modelo 18F4520, ideal para utilização em projetos de baixa potência e aplicações que necessitem das conexões I2C, SPI, UART. Na Figura 23 é ilustrado o PIC utilizado.

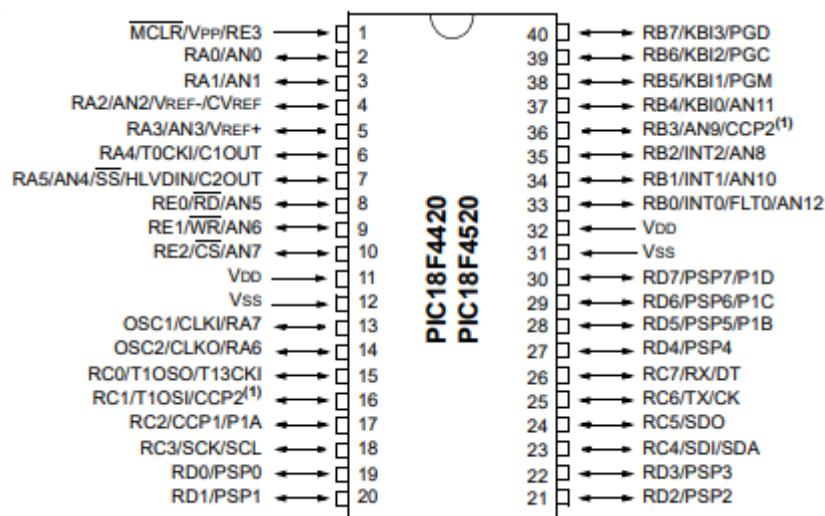
**Figura 23 – PIC utilizado no projeto.**



**Fonte: Autoria própria.**

O PIC selecionado é um modelo de 40 pinos, o que possibilita a montagem de um *hardware* complexo e capaz de interagir com diversos recursos ao mesmo tempo. Esse é um microcontrolador de 8 *bits*, com 32 *kBytes* de memória *flash*, 256 *bytes* de memória EEPROM, possui portas de 8 *bits* de I/O (*Input/Output*) para conexões com periféricos, subdivididas entre A, B, C, D e E, e 13 entradas ADC, sua alimentação é de 5V, além de fácil programação em linguagem C. A Figura 24 mostra a pinagem do PIC18F4520.

Figura 24 – Pinagem do microcontrolador PIC18F4520.



Fonte: Microchip Technology, 2003.

### 3.1.5 Módulo Wi-Fi

Pensando em um fácil acesso às informações que serão processadas pelo módulo medidor de energia, optou-se por transmitir os dados via *Wi-Fi* para serem vistos remotamente, em uma página *web*, pelo usuário. Decidiu-se por esse tipo de apresentação de informação, pelo fato de hoje em dia existirem vários dispositivos com conexão *Wi-Fi* que são de fácil aquisição e por estarem cada vez mais acessíveis.

Para tal, foi escolhido o módulo *Wi-Fi* ESP8266, produzido pela Espressif Systems. Esse módulo é um sistema em um chip, do inglês *System On Chip* (SOC), capaz de fazer conexões TCP/IP, além de possibilitar o acesso à sua rede *Wi-Fi* por qualquer microcontrolador. A Figura 25 mostra a imagem de um módulo ESP8266-12E.

Figura 25 – Módulo *Wi-Fi* ESP8266-12E.

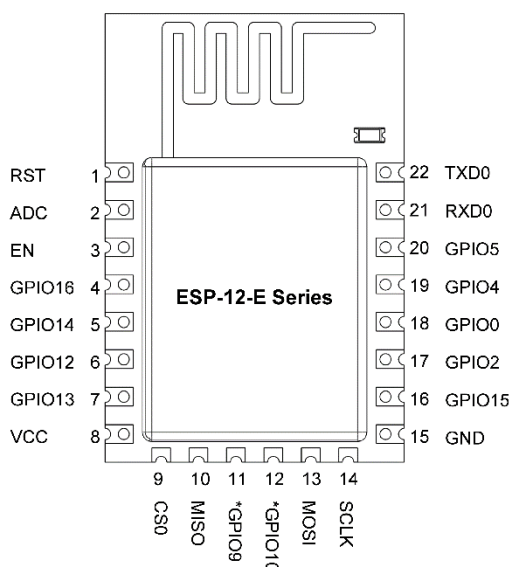


Fonte: Filipeflop, 2017.

Tido como um dos chips que irá impulsionar a evolução da *Internet of Things* (IoT), o ESP8266 é hoje amplamente utilizado em variados projetos. Por essa razão, possui uma vasta biblioteca de manuais e projetos em fóruns da internet. Sendo esse um dos motivos para a escolha desse módulo, pois a infinidade de materiais como referência proporciona uma melhor execução do projeto, além de ter um bom suporte para a solução de possíveis problemas que possam vir a aparecer durante a execução do mesmo.

Esse módulo tem um bom alcance de sinal, cerca de 90m, sua alimentação de operação é de 3,3V, tem suporte às redes padrão *Wi-Fi* 802.11b/g/n com uma taxa de transferência de 110-460800bps, e possui três modos de operação STA(*Station*)/AP(*Access Point*)/STA+AP(combinação dos anteriores), além de ser um *chip* pequeno e ocupar pouco espaço no projeto. A Figura 26 apresenta a pinagem do ESP8266-12E.

**Figura 26 – Pinagem do módulo ESP8266-12E.**



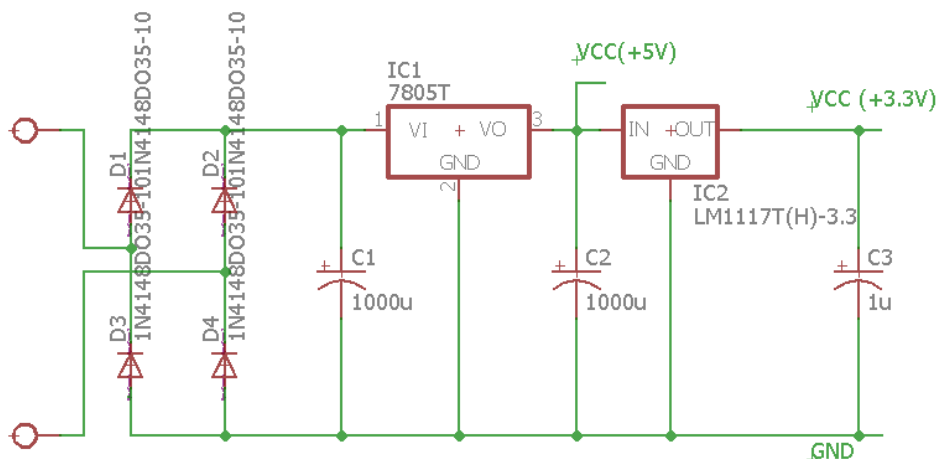
**Fonte: ESPRESSIF, 2017.**

### 3.1.6 Fonte de alimentação

Para o correto funcionamento dos circuitos eletrônicos, é necessário que o nível de tensão esteja adequado. Por esse motivo, foi feita uma fonte de alimentação, que é responsável por receber a tensão proveniente da rede elétrica e condicioná-la para a alimentação dos componentes. Foram então consultados os *datasheets* dos CIs presentes no projeto para o devido dimensionamento da fonte. Após as devidas

consultas, determinou-se que a fonte teria de fornecer dois níveis de tensão: 5V (para o microcontrolador e o medidor de energia) e 3,3V (para o módulo *Wi-Fi*). O circuito da fonte de alimentação é mostrado na Figura 27.

**Figura 27 – Circuito da fonte de alimentação.**



**Fonte: Autoria própria.**

Na entrada da fonte, vai conectado um transformador do tipo abaixador, ele recebe a tensão da rede elétrica 127V/220V e reduz o nível de tensão para 9V. Para a conversão da corrente alternada em corrente contínua, foi usada uma ponte retificadora constituída de quatro diodos. Além disso, foram usados capacitores como filtro a fim de reduzir a ondulação de *ripple* derivada da incompleta supressão da onda alternada. Na sequência, para estabilizar o nível de tensão adequado para os circuitos do projeto, foram utilizados dois reguladores de tensão. Suas especificações são mostradas na Tabela 4.

**Tabela 4 – Reguladores de tensão utilizados no projeto.**

Regulador de tensão	Tensão de entrada	Tensão de saída
LM7805	$7V \leq V_{in} \leq 25V$	+5V
LM1117	$2,6V \leq V_{in} \leq 15V$	+3,3V

**Fonte: Autoria própria.**

### 3.2 Softwares

Com o objetivo de desenvolver de um projeto desta complexidade, foi necessário o uso de diversos *softwares* para que se desse prosseguimento nas etapas de sua construção. Para tal, foram escolhidas ferramentas como: Eagle



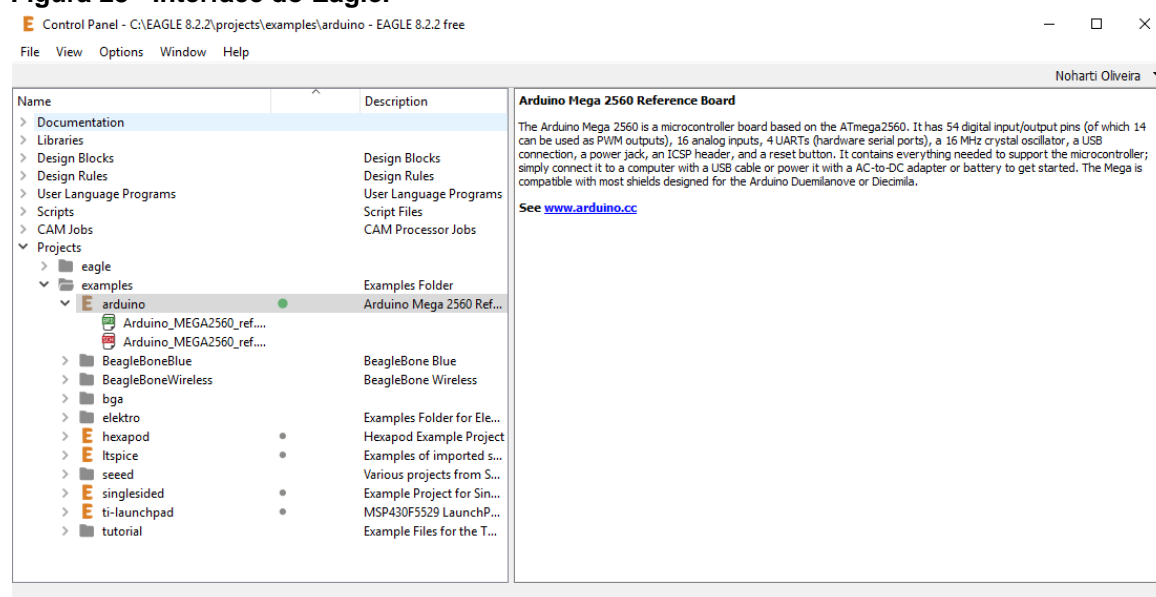
(desenvolvimento dos circuitos das placas), MPLAB X IDE (programação do microcontrolador), Visual Studio e outras para a criação e manutenção do *website*. A seguir serão detalhadas cada uma das ferramentas, explicando seus funcionamentos e a razão de terem sido escolhidas.

### 3.2.1 Eagle

Para o desenvolvimento das placas com os circuitos necessários para a integração de todos os componentes, foi utilizado o Eagle 8.2.2, que é uma das ferramentas CAD dedicada à criação de placas de circuito impresso mais conhecidas atualmente. Essa é uma versão gratuita, que possui algumas limitações, porém para esse projeto essas limitações não são proibitivas.

O Eagle abrange todas as etapas de desenvolvimento de uma PCI (Placa de Circuito Impresso), desde a elaboração do esquemático do circuito até a geração dos arquivos para confecção da placa (SILVA, 2016). Além dessas características, o *software* chama a atenção por possuir uma vasta biblioteca de componentes e também permitir que o usuário crie novos ou altere os já existentes. A figura 28 traz a captura de tela da página inicial desse *software*.

**Figura 28 - Interface do Eagle.**

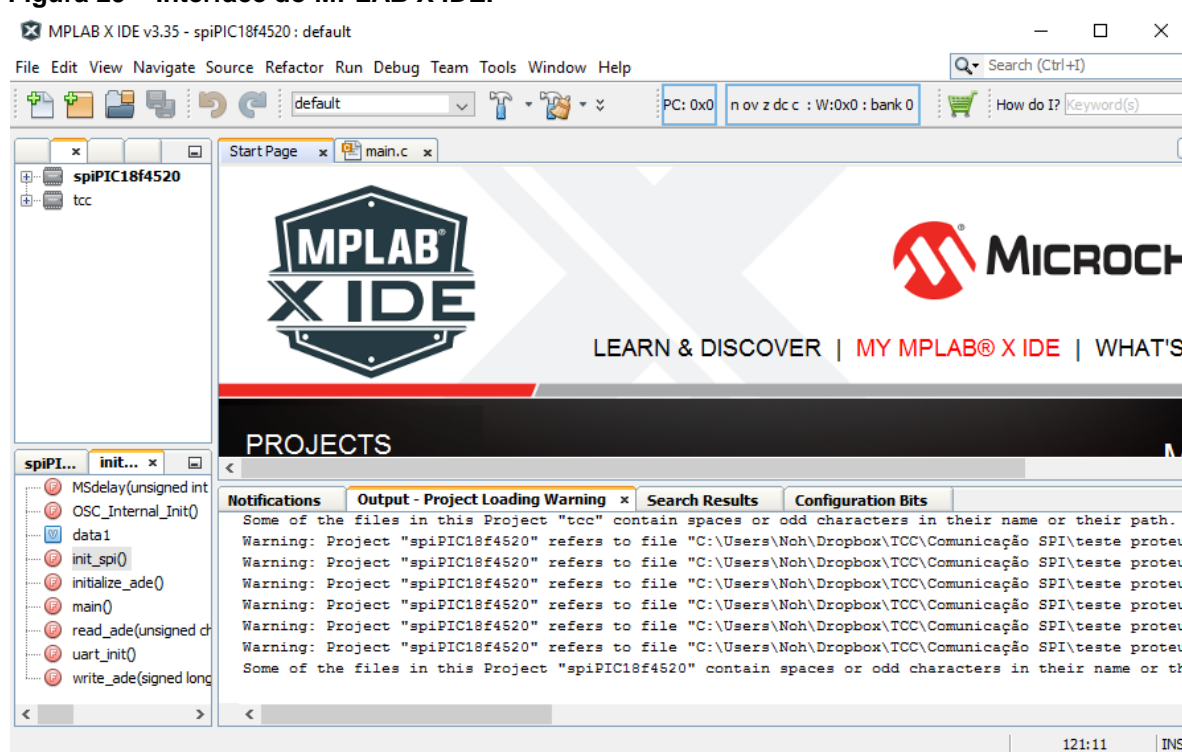


Fonte: Autoria própria.

### 3.2.2 MPLAB X IDE

A plataforma MPLAB X IDE (*Integrated Development Environment*) é um *software* fornecido pela empresa Microchip para o desenvolvimento de sistemas embarcados. Possui um ambiente de programação e gravação de microcontroladores e controladores digitais produzidos pela empresa. A versão que foi usada para a programação é gratuita, com funcionalidades suficientes para a criação do projeto. Na Figura 29 é possível ver este *software* aberto com as opções de criar projetos.

**Figura 29 – Interface do MPLAB X IDE.**



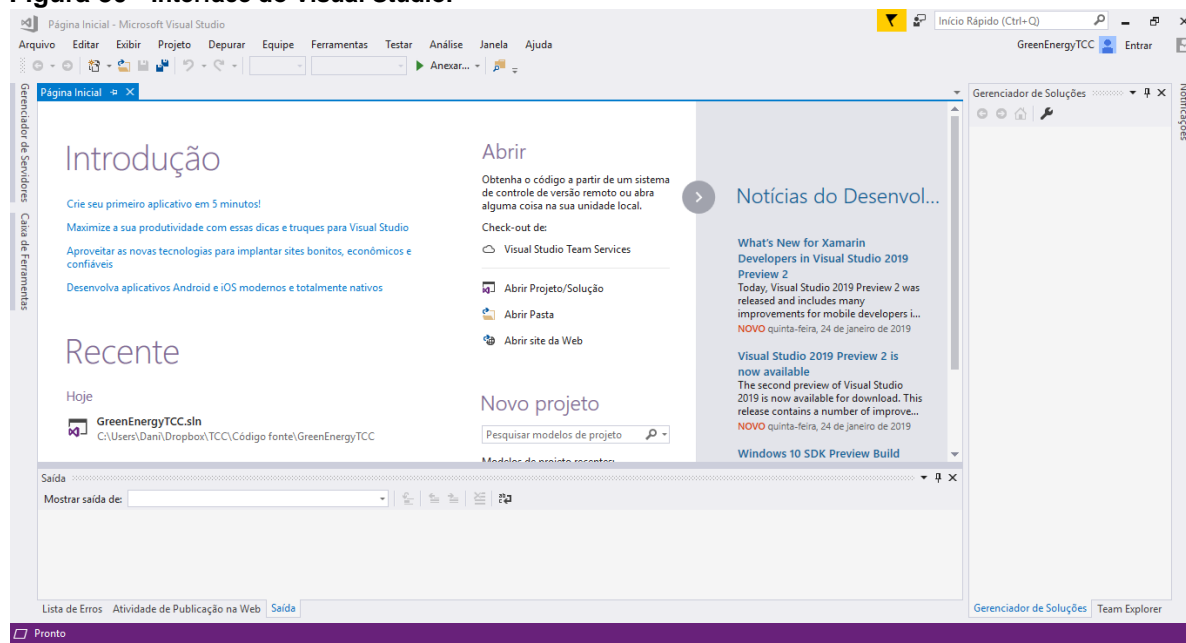
Fonte: Autoria própria.

### 3.2.3 Visual Studio

O Visual Studio é um ambiente de desenvolvimento integrado (IDE) da Microsoft especialmente dedicado ao desenvolvimento na área *web*, usando a plataforma do ASP.NET, como *websites*, aplicativos *web* e aplicativos móveis. Ele é utilizado para exibir e editar vários tipos de linguagens, tais como Visual Basic, C, C++, C# (C Sharp), além de compilar e publicar aplicativos para Android, IOS, Web, entre outros. Sua característica fundamental e mais importante é que facilita o desenvolvimento fazendo com que o ambiente fique interativo e intuitivo

(MICROSOFT, 2018). Foi utilizada nesse projeto a versão para estudante do Visual Studio 2017. Na Figura 30 é possível observar a página de início do Visual Studio.

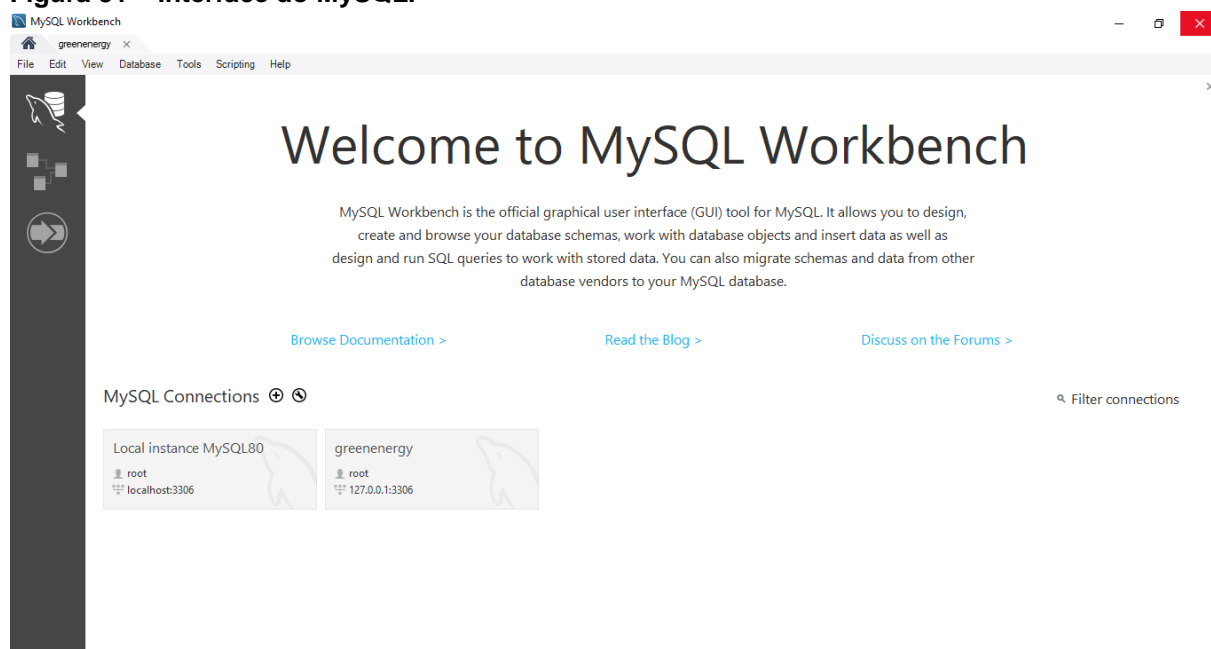
**Figura 30 - Interface do Visual Studio.**



**Fonte: Autoria própria.**

### 3.2.4 MySQL

O MySQL é um sistema de gerenciamento de banco de dados, que utiliza a linguagem SQL (Linguagem de Consulta Estruturada) como interface. É atualmente um dos sistemas de gerenciamento de bancos de dados mais populares, com mais de 10 milhões de instalações pelo mundo. Este sistema é conhecido por sua facilidade de uso, por esse motivo é usado por várias empresas pelo mundo, tais como: NASA, HP, Bradesco, Sony, etc. Sua interface simples, facilidade para programação e também sua capacidade de rodar em vários sistemas operacionais, são alguns dos motivos para o crescente uso deste programa (PACIEVITC, 2011). Nesse trabalho, o banco de dados MySQL irá armazenar as informações do consumo de energia elétrica do usuário, para que este possa acessar essa informação na data em que quiser. A Figura 31 mostra a interface do MySQL.

**Figura 31 – Interface do MySQL.**

Fonte: Autoria própria.

### 3.2.5 Google Charts

Os gráficos são de suma importância na exposição de informações, pois estes ajudam na visualização e entendimentos de dados colhidos de maneira clara e objetiva. Porém, a montagem de gráficos oferece uma certa dificuldade em qualquer linguagem. Então, para facilitar na programação da página *web*, optou-se por utilizar o Google Charts.

O Google Charts é um serviço da *web* interativo que cria gráficos a partir de informações fornecidas pelo usuário. Ele contém uma grande variedade de gráficos, como por exemplo, gráficos de linhas, gráficos *spline*, gráficos de área, gráficos de barras, gráficos de pizza e assim por diante. Para gerar um gráfico o usuário fornece dados e uma especificação de formatação em JavaScript, em resposta, o serviço envia uma imagem do gráfico (DOUGLAS,2012).

## 4. DESENVOLVIMENTO DO PROJETO

Neste capítulo é apresentado o desenvolvimento do projeto, abordando a integração de cada ferramenta ao projeto do protótipo do medidor.

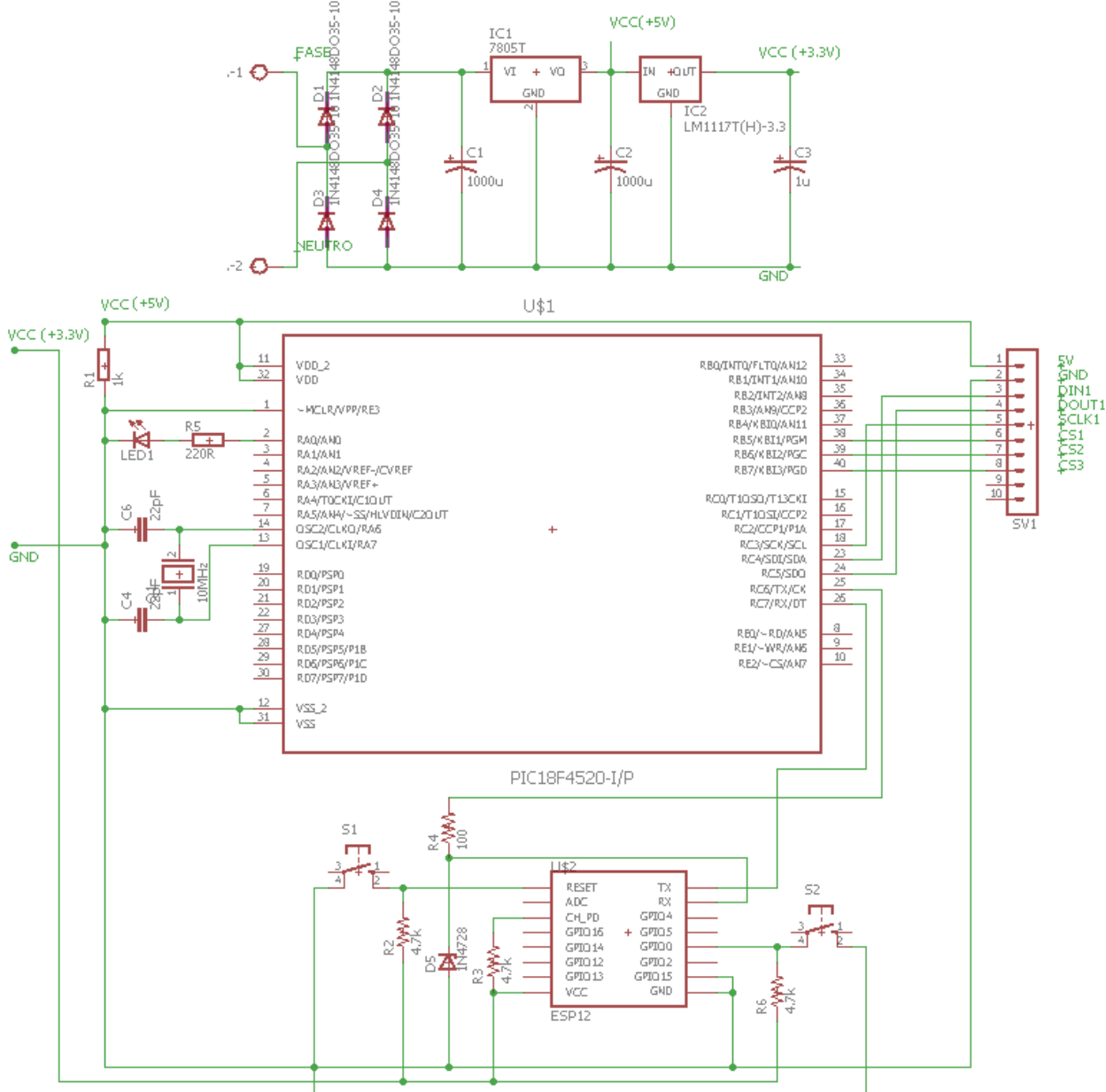
### 4.1 Placas de circuito impresso

Para a realização das medições foram projetadas duas placas, uma contendo a fonte de alimentação, o módulo *Wi-Fi* e o microcontrolador, denominada de placa de comunicação, e outra contendo os sensores de tensão e de corrente e o módulo medidor de energia, denominada de placa de medição.

#### 4.1.1 Placa de comunicação

Essa placa é responsável por fazer toda a comunicação entres os principais componentes do projeto, desde os valores registrados pelos medidores até o envio das informações, através do *Wi-Fi*, para o banco de dados do servidor, para que o usuário então possa acessá-las pela *internet*. Além disso, é nessa placa que vai a fonte de alimentação que condiciona os valores de tensão para o funcionamento dos componentes. Para a construção dos circuitos das placas foi utilizado o *software* Eagle já apresentado no capítulo 3. Na Figura 32 é possível observar o esquemático da placa de comunicação.

Figura 32 – Circuito da placa de comunicação.



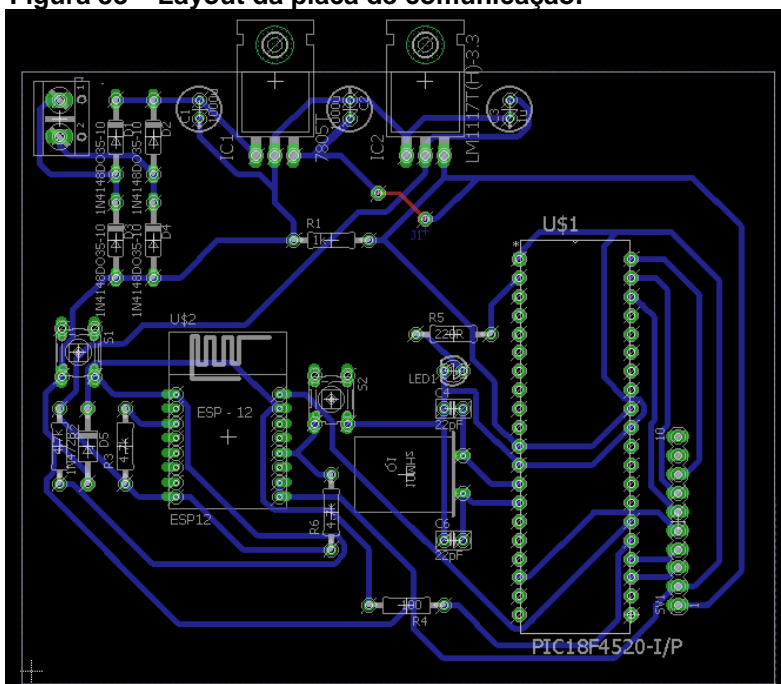
Fonte: Autoria própria.

Como é possível observar no esquemático acima, essa placa contém a fonte de alimentação de todo o sistema, o microcontrolador, o módulo *Wi-Fi*, elementos necessários para o bom funcionamento e comunicação entre os componentes (resistores, capacitores, reguladores de tensão, botões), além de uma barra de pinos fêmea.

Para a ligação entre as placas, são utilizados 8 pinos da barra de pinos. O primeiro pino é responsável por mandar 5V, proveniente da fonte, para também alimentar os componentes da placa de medição, já o segundo é o pino terra. O restante dos pinos é usado para a comunicação SPI. O terceiro pino é usado para

enviar os dados dos escravos para o mestre. O quarto é o pino usado para enviar os dados do mestre para os escravos. O quinto é o pino utilizado para mandar o sinal de *clock* gerado pelo mestre para os escravos. Os pinos de número seis, sete e oito são usados para o *chip select*, ou seja, seleção dos escravos. Na Figura 33 é mostrado o *layout* desta placa pronto para impressão e na Figura 34, uma foto desta placa finalizada com os componentes já soldados.

**Figura 33 – Layout da placa de comunicação.**



Fonte: Autoria própria.

**Figura 34 – Placa de comunicação finalizada.**

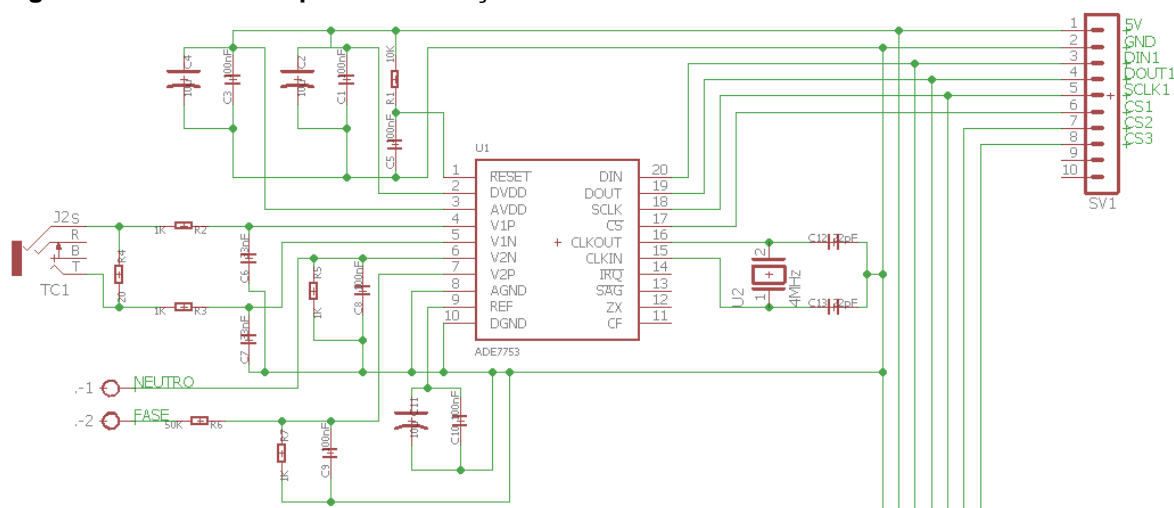


Fonte: Autoria própria.

#### 4.1.2 Placa de medição

Na placa de medição foram posicionados os sensores de tensão e de corrente, com os respectivos circuitos divisores de tensão. Nela estão inseridos também os *chips* medidores de energia, além de uma barra de pinos macho para fazer a conexão com a placa de comunicação. Na Figura 35 pode-se ver o esquemático de um dos três circuitos medidores, os outros dois possuem os mesmos componentes e ligações, diferenciando-se somente a ligação do *chip select*.

**Figura 35 – Circuito da placa de medição.**

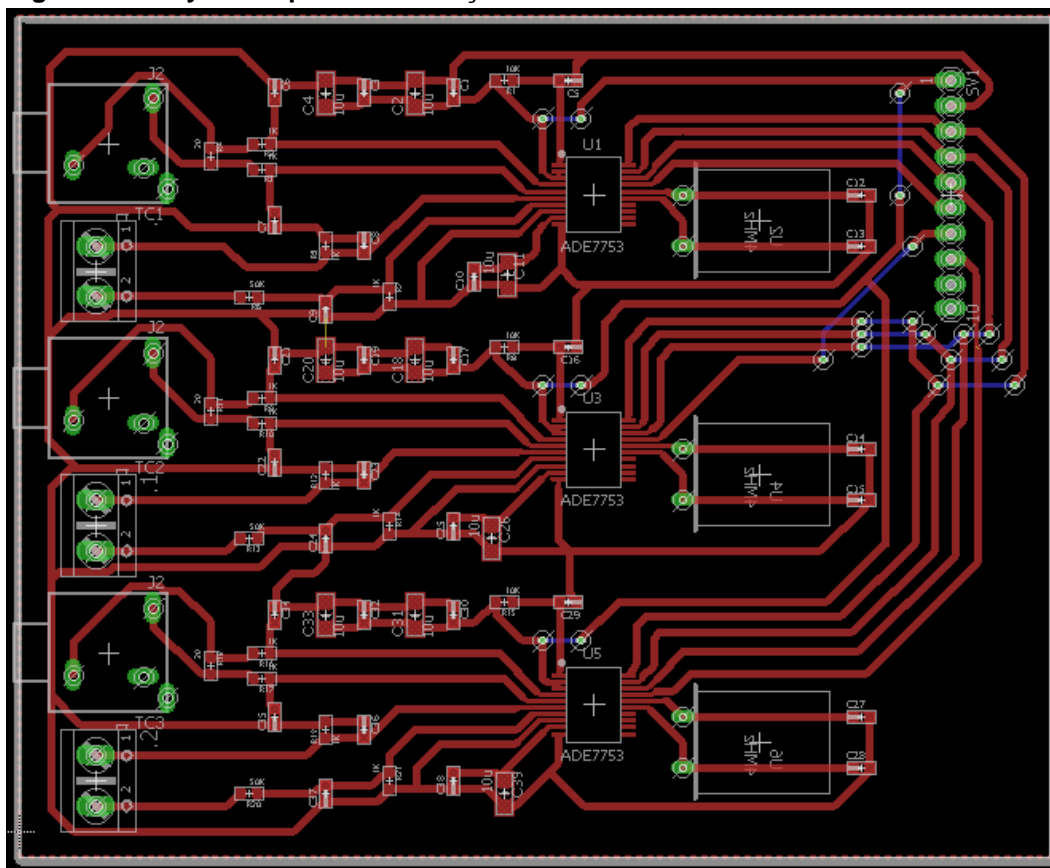


Fonte: Autoria própria.

Além dos componentes já mencionados, a placa de medição também possui três conectores Jack J2 para a ligação dos sensores de corrente (1 pra cada circuito), três conectores borne 2 vias, além de resistores e capacitores para condicionamento e filtragem dos sinais. O *layout* da placa de medição pode ser visto na Figura 36.



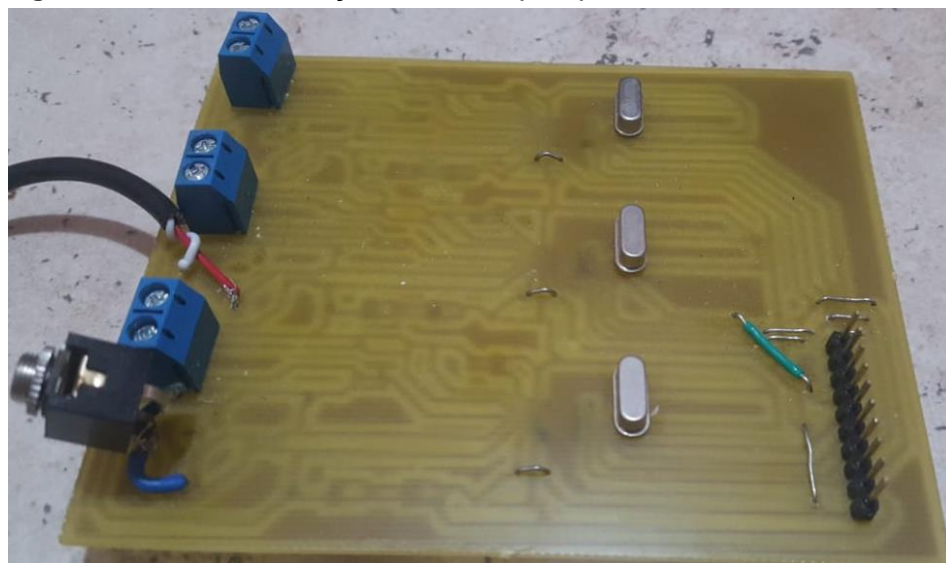
Figura 36 – Layout da placa de medição.



Fonte: Autoria própria.

Na Figura 37, uma foto da face de cima (TOP) desta placa finalizada com os componentes já soldados.

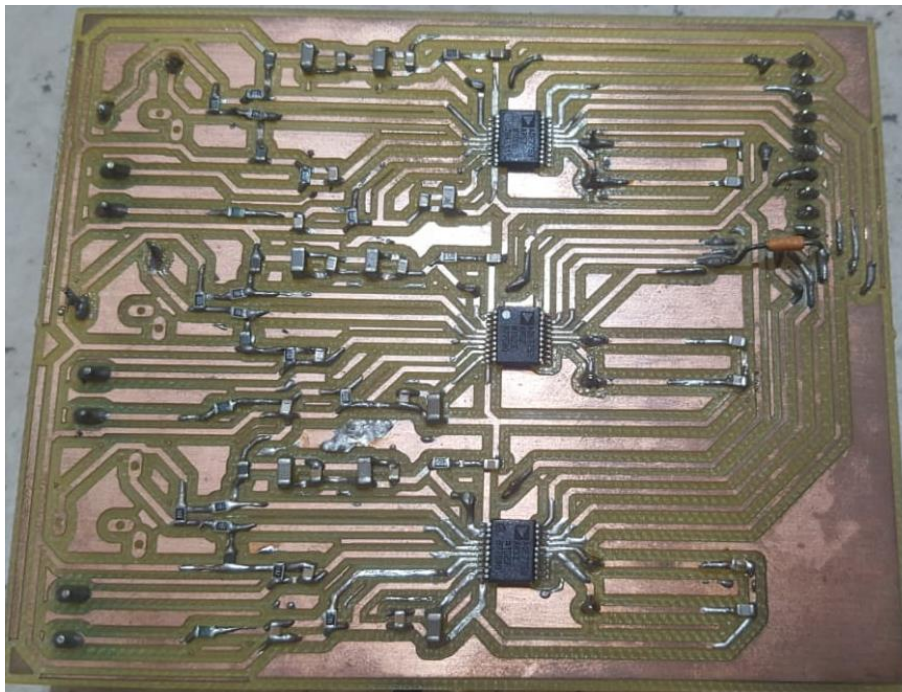
Figura 37 – Placa de medição finalizada (TOP).



Fonte: Autoria própria.

Já na Figura 38, é possível ver face de baixo (BOTTOM) da placa finalizada.

**Figura 38 – Placa de medição finalizada (BOTTOM).**



Fonte: Autoria própria.

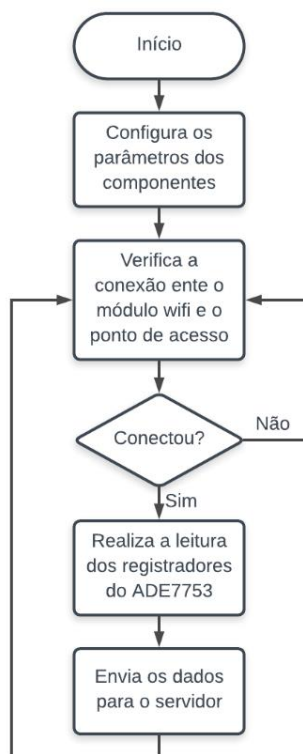
## 4.2 Programação

Com as placas completas em mãos, foi dado início às programações necessária para a integração e funcionamento do projeto.

### 4.2.1 Programa embarcado no microcontrolador

Toda a programação do microcontrolador foi feita em linguagem C, através do *software* MPLAB X IDE. O funcionamento esperado do microcontrolador e dos componentes integrados a ele pode ser esquematizado pelo fluxograma do *firmware* mostrado na Figura 39.

**Figura 39 – Fluxograma do Firmware.**



**Fonte: Autoria própria.**

Como pode ser visto na Figura 39, após a inicialização do microcontrolador, são realizadas as configurações básicas e essenciais para o funcionamento de cada componente integrado ao projeto. O programa possui códigos que inicializam as bibliotecas para o funcionamento correto dos medidores ADE7753 e do módulo ESP8266, além das bibliotecas de suporte para as comunicações SPI e USART. Todo o código desenvolvido pode ser visto nos Apêndices no final do trabalho.

Na função *main* do código são chamadas as funções de parametrização e de inicialização, que foram criadas separadamente para facilitar na programação e leitura do código. As funções de parametrização configuram os parâmetros necessários para que os dispositivos se comuniquem corretamente. Já as funções de inicialização, como o próprio nome diz, são as funções que fazem a inicialização dos dispositivos.

Em seguida, o módulo ESP8266 faz uma tentativa de conexão ao ponto de acesso, se ele obtiver sucesso entrará no loop que realiza a leitura dos registradores dos módulos medidores de energia, caso contrário ele fará uma nova tentativa de conexão. Então, o PIC manda as informações coletadas dos registradores para o

módulo *Wi-Fi*, via comunicação USART. Na sequência, o ESP8266 envia os valores para o servidor.

O envio dos dados para o servidor será feito por meio do protocolo HTTP, responsável pela comunicação entre os *sites* na *web*. Esse protocolo baseia-se em requisições e respostas entre clientes e servidores. O cliente (aplicação embarcada) fará a requisição ao servidor (neste caso, o *site web*) para que os dados coletados sejam salvos e o servidor, por sua vez, será responsável por armazená-los no banco de dados e torná-los disponíveis para as consultas realizadas através do *site*.

Para realizar esta requisição, será utilizado o método padrão de requisição HTTP: GET. Esse método é utilizado quando deseja-se passar uma pequena quantidade de informações (limite máximo de 1024 caracteres) para realizar uma pesquisa ou mesmo transmitir uma informação para outra página através da URL (*Uniform Resource Locator*) (VICTORIO, 2016).

No caso do *site* desenvolvido, o módulo envia as informações através da URL: `https://192.168.0.9:45456/Home/SalvarConsumo?idMedidor=1&energiaAtiva=10&potenciaReativa=10&energiaAparente=15&corrente=2&tensao=5`.

No exemplo acima, as informações que vêm logo após o sinal de interrogação demarcam o início dos dados passados (os parâmetros) através da URL (ex.: `idMedidor = 1`), separados pelo símbolo '&' para concatenar mais de um dado.

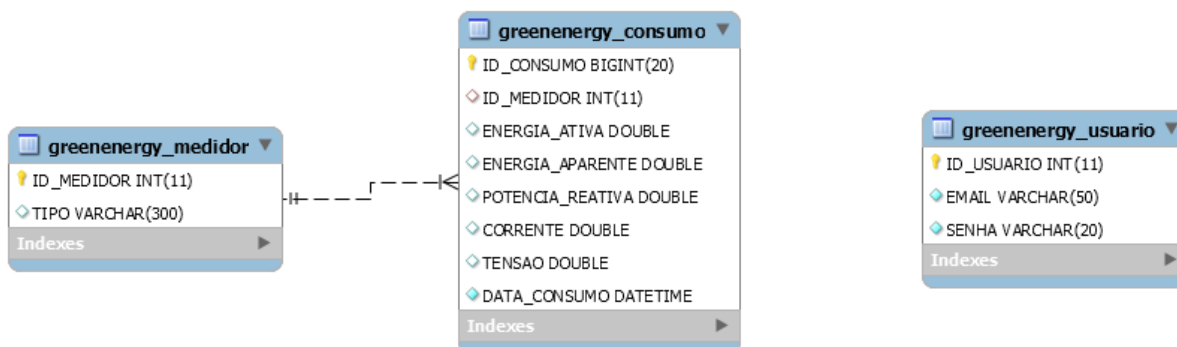
#### 4.2.2 Desenvolvimento do site

A interface gráfica do *site* foi pensada com o intuito de facilitar a leitura e entendimento do consumidor, para que este navegue pelas páginas encontrando as informações necessárias de forma clara e objetiva.

Para o desenvolvimento da plataforma *web*, foi utilizado o *software* Visual Studio citado anteriormente no Capítulo 3. O código foi criado utilizando as linguagens SQL, ASP.NET, JavaScript e CSS.

Para armazenar os dados obtidos pelos medidores e informações da página, tais como *login* e alteração de senha, foram criadas 3 tabelas no banco de dados relacional MySQL. A Figura 40 mostra o diagrama de blocos do banco de dados.

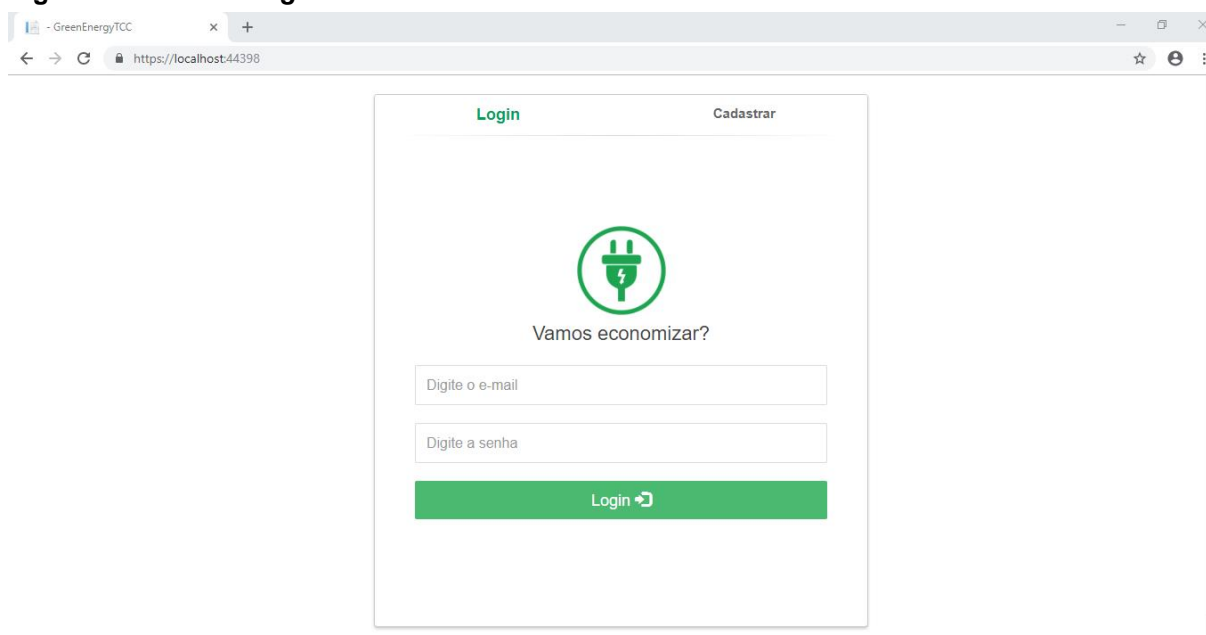
**Figura 40 – Diagrama do banco de dados MySQL.**



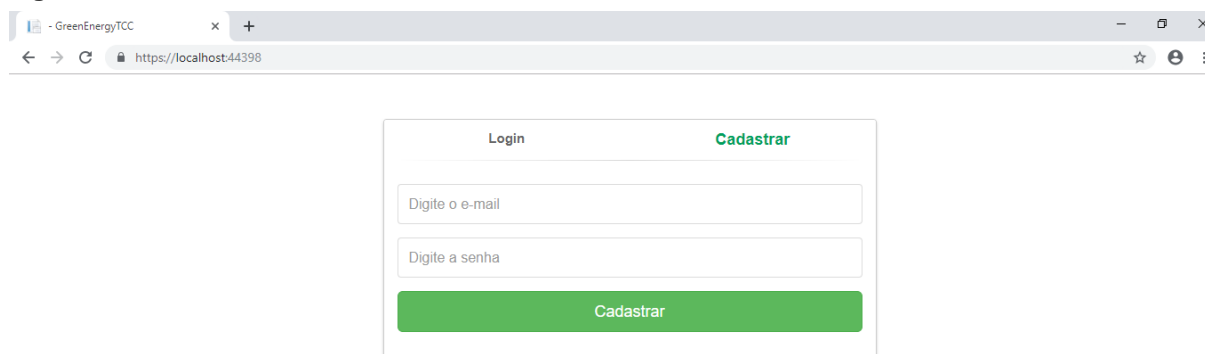
**Fonte: Autoria própria.**

A página inicial do *site* contém uma tela de *login* e uma tela para cadastrar o usuário. Na tela de cadastro o usuário tem a opção de digitar um *e-mail* e uma senha para ter acesso ao conteúdo do *site*. As telas iniciais e de cadastro podem ser vistas nas Figuras 41 e 42.

**Figura 41 – Tela de *login* do site.**

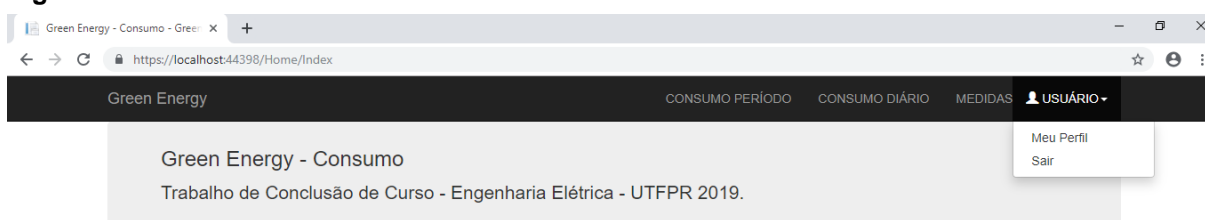


**Fonte: Autoria própria.**

**Figura 42 – Tela de cadastro do site.**

**Fonte: Autoria própria.**

Ao inserir corretamente as informações cadastradas na tela de *login*, o usuário terá acesso à página principal do *site*, que contém os menus para acessar os medidores, às informações de potências e uma tela em que o mesmo poderá alterar os dados de *login*. A Figura 43 mostra a página principal.

**Figura 43 – Tela de inicial do site.**

**Fonte: Autoria própria.**

O *site* possui dois menus para a leitura das informações de consumo: consumo diário e consumo por período. Na tela de consumo diário, o usuário consegue visualizar as informações do consumo de cada medidor em tempo real ou pelo período de um dia predeterminado por ele. A página possui um calendário interativo no qual o usuário pode selecionar a data desejada para a visualização do consumo. Os dados são mostrados através de dois gráficos, um que apresenta o kW por hora e um que apresenta o valor em reais gasto por hora, além de uma *box* com o valor total consumido no dia. Ainda, o consumidor pode selecionar o tipo de bandeira tarifária que está vigente no período de consulta, podendo fazer comparação do quanto gastaria nos diferentes tipos de bandeiras tarifárias disponíveis para o consumidor residencial. Os gráficos foram desenvolvidos através da ferramenta

Google Charts, citada no Capítulo 3. Na Figura 44 é apresentada a tela do consumo diário.

**Figura 44 – Tela de consumo diário.**

**Fonte: Autoria própria.**

Já na tela de consumo por período, o usuário consegue acessar os dados de cada medidor por um período de tempo determinado por ele. Essa tela possui entrada para duas datas, uma com a data inicial e uma com a data final, para que o usuário tenha flexibilidade na sua consulta, uma vez que a fatura de energia elétrica nem sempre se dá no início do mês. Assim como na tela anterior, o usuário visualiza dois gráficos e uma *box* com o valor total em reais consumido no período. É possível também selecionar uma das opções de bandeira tarifária. A tela do menu de consumo por período pode ser vista na Figura 45.

**Figura 45 - Tela de consumo por período.**

**Fonte: Autoria própria.**

Além dos menus de informação do consumo, o usuário tem um menu com dados das grandezas elétricas sendo medidas em cada medidor. Nessa tela é possível visualizar gráficos com os dados da tensão, corrente, potência reativa, potência aparente e fator de potência registrados no dia selecionado pelo usuário. A Figura 46 mostra a tela do menu medidas.

**Figura 46 – Tela de medidas.**

Green Energy - Medidas  
Trabalho de Conclusão de Curso - Engenharia Elétrica - UTFPR 2019.

**Medidores**  
Medidor 3

**Grandeza elétrica**  
Corrente

**Data**  
09/02/2019

Visualizar

Não há informações cadastradas para a data selecionado!

**Fonte: Autoria própria.**

Por fim, o usuário tem uma tela em que ele pode alterar a sua senha de acesso quando quiser. É possível ver a tela na Figura 47.

**Figura 47 - Tela do usuário.**

Green Energy - Consumo  
Trabalho de Conclusão de Curso - Engenharia Elétrica - UTFPR 2019.

**Meu Perfil**

**Email**  
noharti@gmail.com

**Alterar senha**  
Digite a senha

Salvar

**Fonte: Autoria própria.**



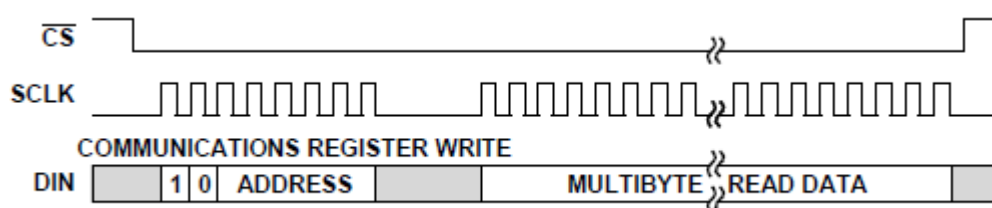
## 5. RESULTADOS

Após realizadas a confecção e montagem das placas, foram feitos testes com os medidores de energia ADE7753 e com o módulo *Wi-Fi* para verificar os seus funcionamentos e validar as medições.

O ADE7753 possui vários registradores que podem ser acessados para configurar todas as suas funcionalidades. Esses registradores devem ser ajustados com valores que possibilitam uma interpretação correta dos dados processados pelo *chip*. O endereço desses registradores possui um padrão de 8 *bits*, ao passo que a resolução dos mesmos varia de 6 a 24 *bits*. O acesso a esses endereços para escrita e leitura é feito através da comunicação SPI.

Para o ajuste dos valores a serem medidos pelo CI, o microcontrolador PIC18F4520 foi configurado como dispositivo mestre, enquanto que o CI é definido como dispositivo escravo. A comunicação com o ADE7753 é sempre iniciada com uma operação de escrita. Para escrever nos registradores do ADE7753, é necessário enviar o MSB (*Most Significant Bit*) com o valor de 1. Portanto, para se configurar um registrador, o mestre tem que enviar um dado contendo um *byte* com o MSB = 1 e o restante sendo o endereço do registrador, como pode ser visto na Figura 48.

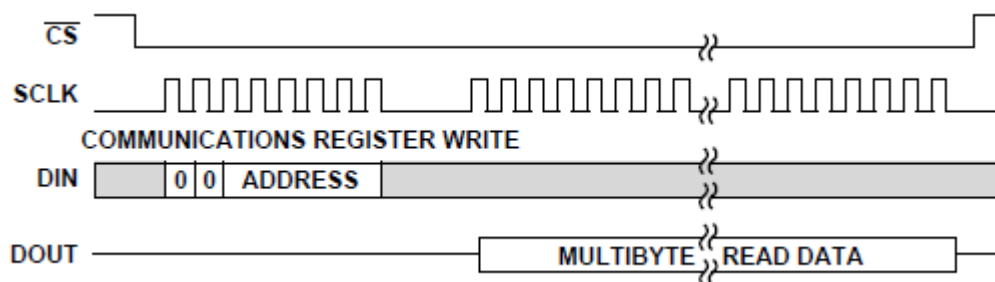
Figura 48 – Escrita de dados no padrão da comunicação SPI do CI ADE7753.



Fonte: Analog Devices, 2016.

Já para a operação de leitura do conteúdo dos registradores do ADE7753, é necessário enviar o MSB com o valor de 0 mais o endereço do registrador, para que seja permitida a operação de leitura na próxima transferência. O mestre então envia um *Dummy Byte* (byte lixo) para que o CI lhe envie o conteúdo do registrador. Na Figura 49 é mostrada a operação de leitura dos registradores do ADE7753.

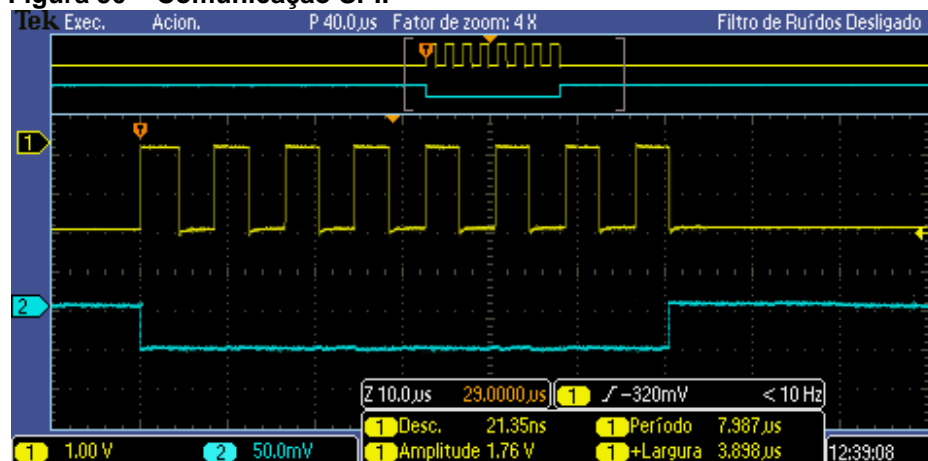
Figura 49 – Leitura de dados no padrão da comunicação SPI do CI ADE7753.



Fonte: Analog Devices, 2016.

Foi então montado um rápido teste para a avaliação do correto funcionamento da comunicação SPI entre PIC e o ADE7753. Para isso, foi mandado do PIC18F para o chip medidor de energia um comando de leitura de um registrador de 8 bits. Os sinais de *clock* e *chip select* foram amostrados com o osciloscópio digital da Tektronix MISO 2024 com dois canais e são mostrados na Figura 51. O sinal com cor amarelo (CH1) representa o sinal de *clock* e o sinal com cor azul (CH2) caracteriza o sinal *chip select*.

Figura 50 – Comunicação SPI.



Fonte: Autoria própria.

Com a comunicação funcionando, foram feitos testes de medição de corrente com algumas cargas com o intuito de verificar o erro do valor medido. De acordo com o módulo 5 do Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional (PRODIST), o valor máximo de incerteza na medição é dado pela Equação 14:

$$e(\%) = 0,05 \pm \sqrt{M^2 + T_C^2 + T_P^2} \quad (14)$$

Onde:

- $M$  é a incerteza padrão do medidor;
- $T_C$  é a incerteza padrão do transformador de corrente (TC);
- $T_P$  é a incerteza padrão do transformador de tensão (TP);
- 0,05 é o erro sistemático, que é imposto pela resistência do cabo do TP.

Conforme definido nesse módulo, para medições de energia dos consumidores do grupo B, devem ser utilizados TC's e TP's classe 0,6, ou seja, a precisão desses dispositivos deve ser de 0,6%. Já o medidor, deve ter uma incerteza definida em 0,05%. Utilizando a Equação 14, é possível obter um desvio de aproximadamente 1,03%.

Para a alimentação das cargas de teste, foi utilizada a tensão de 127V, disponível na rede do laboratório. Para efeito de comparação, foi utilizado o multímetro digital Minipa ET-2507A para fazer a medição das correntes. O erro relativo das medições, expresso pela Equação 15, foi calculado em relação ao valor esperado.

$$\text{erro relativo}(\%) = \frac{\text{valor medido} - \text{valor referência}}{\text{valor referência}} * 100 \quad (15)$$

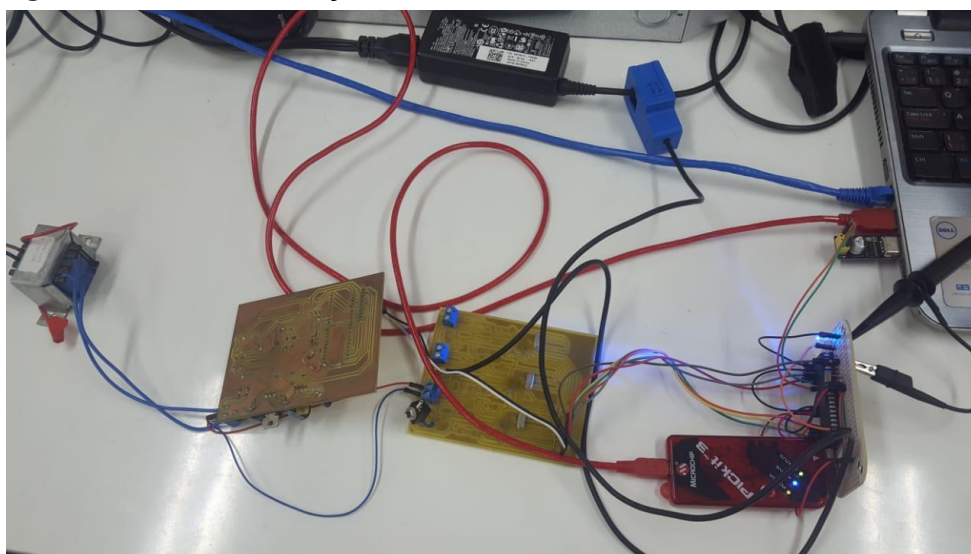
Foi criada uma interface rápida no terminal serial do Arduino apenas para poderem ser visualizados os valores medidos pelo medidor de energia. A montagem para a medição de uma das cargas pode ser vista nas Figuras 51 e 52.

**Figura 51 – Montagem do protótipo para testes.**



Fonte: Autoria própria.

**Figura 52 – Teste de medição da corrente da fonte do notebook.**



**Fonte: Autoria própria.**

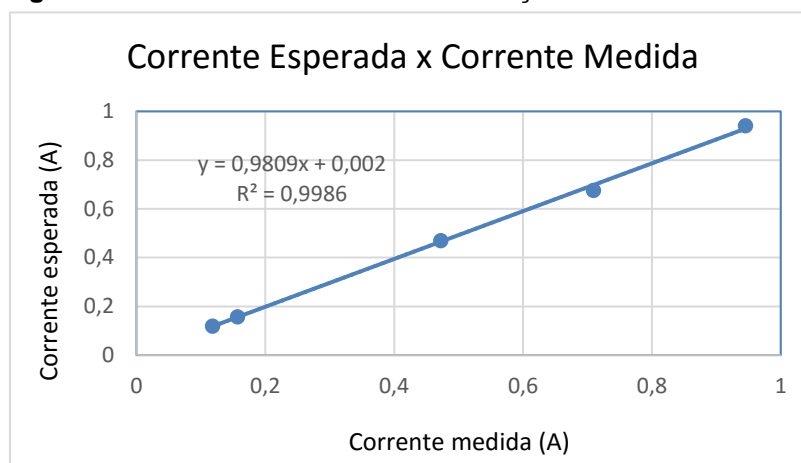
A partir dos valores de erros obtidos das medições, é possível afirmar que o medidor trabalha com uma tolerância aceitável na medição de corrente. A linearidade das amostras é quase que totalmente garantida, pois  $R^2 = 0,9986$  (quanto mais perto de 1, mais garantida é a linearidade). Além disso, o maior erro obtido nos testes de medição de corrente foi de 1,19%, muito pouco acima do valor calculado recomendado pelo PRODIST (1,03%). Na Tabela 5 são apresentadas todas as cargas utilizadas no teste, suas respectivas potências, assim como os valores das medições de corrente e os erros relativos. O gráfico ilustrado na Figura 53 dá uma ideia da linearidade das medições efetuadas.

**Tabela 5 – Comparativo entre medições de corrente com o ADE7753.**

<b>Carga</b>	<b>Potência Nominal</b>	<b>Corrente Esperada</b>	<b>Corrente Multímetro</b>	<b>Corrente Medidor</b>	<b>Erro %</b>
Lâmpada Incandescente	120W	0,945A	0,941A	0,942A	-0,32
Lâmpada Incandescente	60W	0,472A	0,469A	0,470A	-0,42
Lâmpada Fluorescente	20W	0,157A	0,154A	0,158A	0,64
Lâmpada LED	15W	0,118A	0,116A	0,119A	0,85
Fonte Notebook Dell Inspiron 15R	90W	0,709A	0,668A	0,676A	-1,19

**Fonte: Autoria própria.**

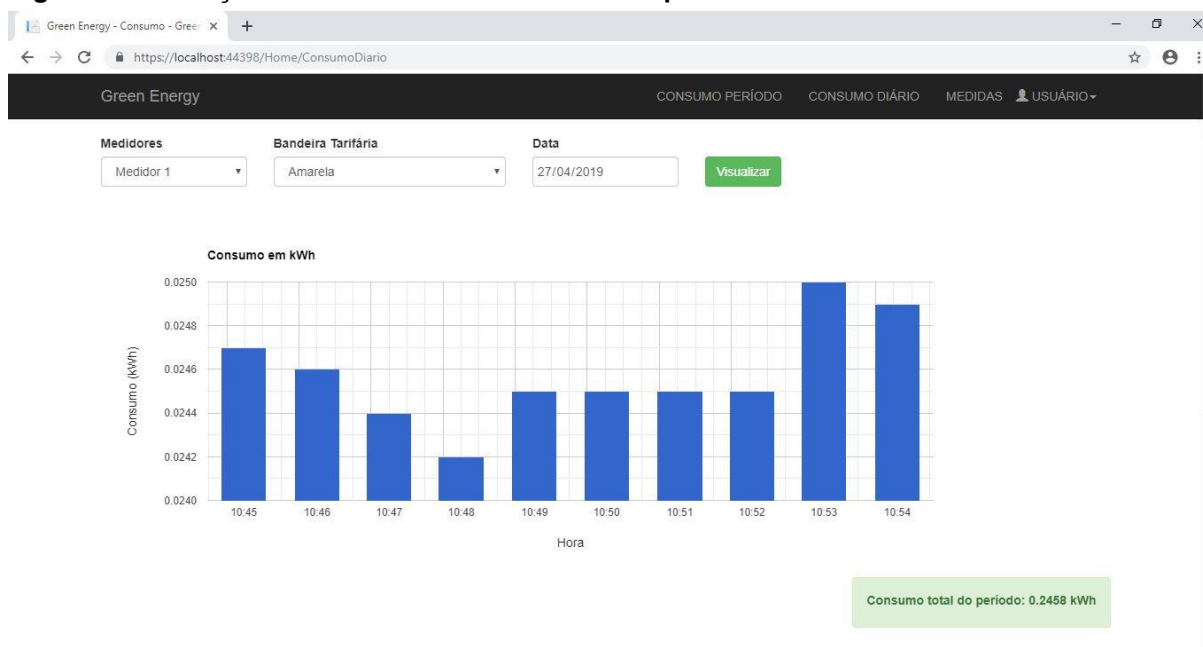
**Figura 53 – Gráfico linearidade das medições de corrente.**



Fonte: Autoria própria.

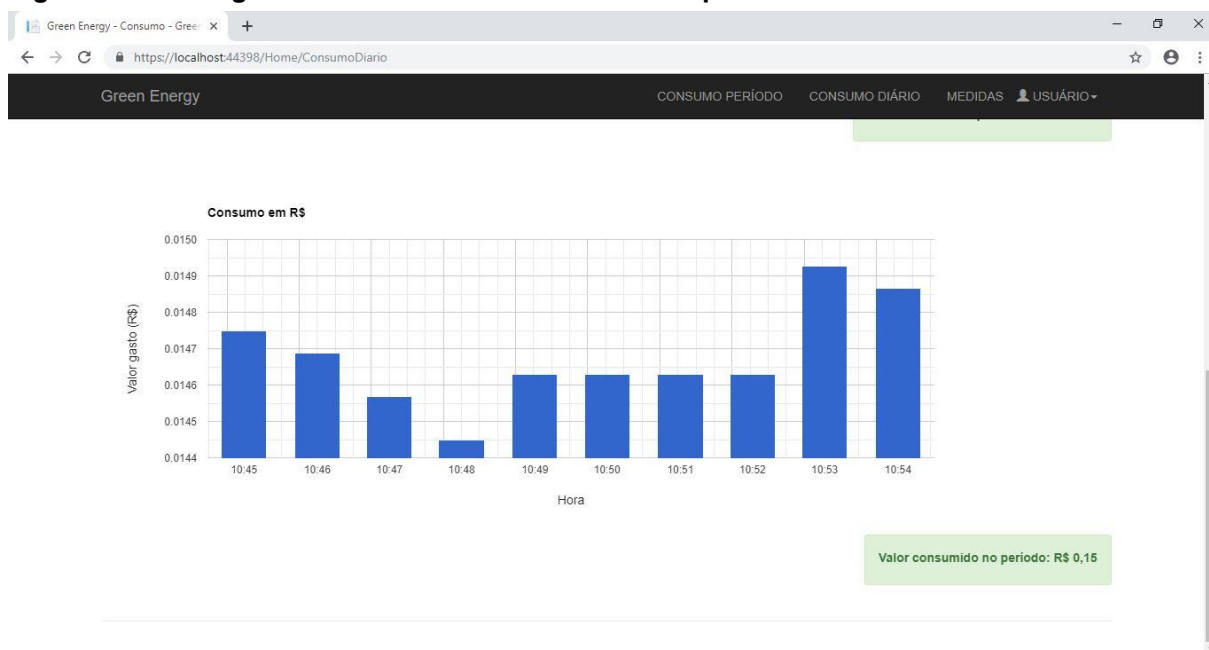
O próximo passo foi testar o envio das informações lidas para o banco de dados e a visualização das informações na página da *internet*. Para isso foram monitoradas algumas cargas. A primeira carga foi um ferro de passar roupas do modelo Philips Walita Azur Performer com potência de 1470W. Na sequência serão mostradas as figuras com os resultados obtidos da monitoração.

**Figura 54 – Medição do consumo diário do ferro de passar.**



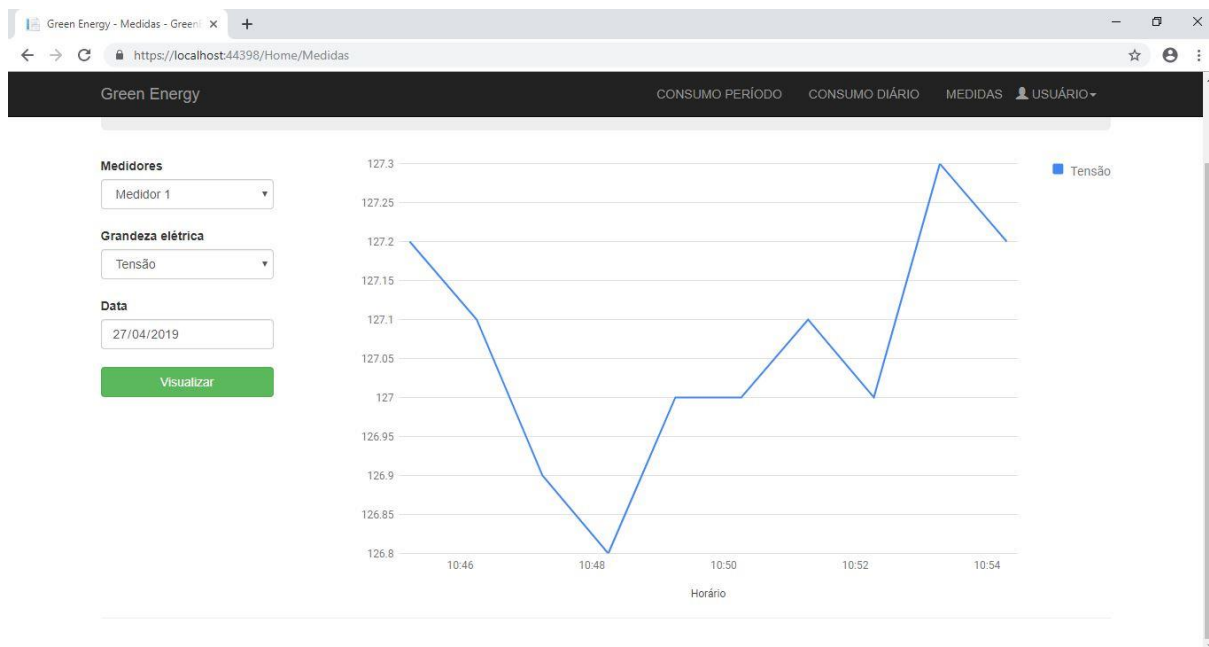
Fonte: Autoria própria.

**Figura 55 – Valor gasto no consumo diário do ferro de passar.**



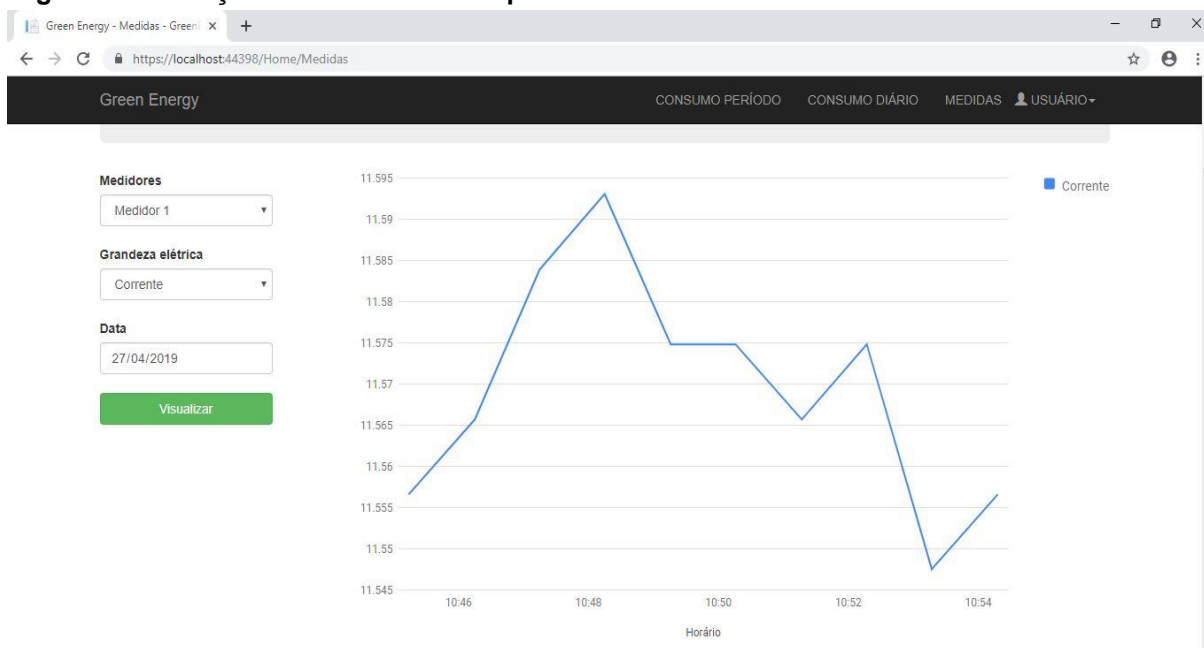
Fonte: Autoria própria.

**Figura 56 - Medição da tensão ferro de passar.**



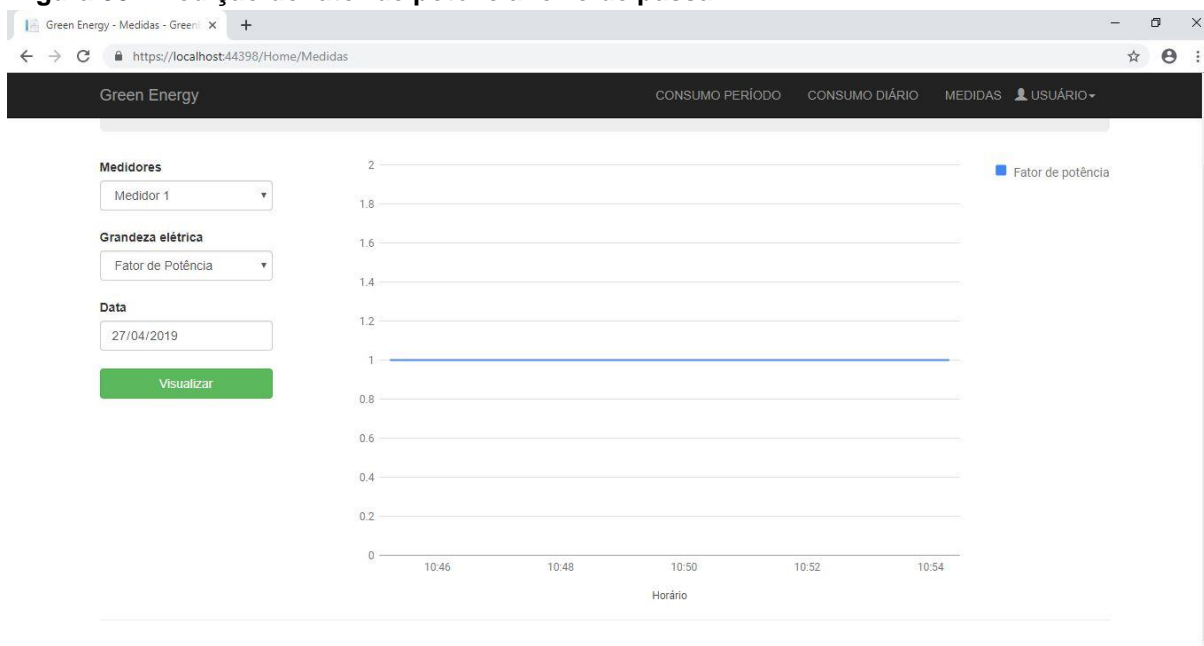
Fonte: Autoria própria.

**Figura 57 - Medição da tensão ferro de passar.**



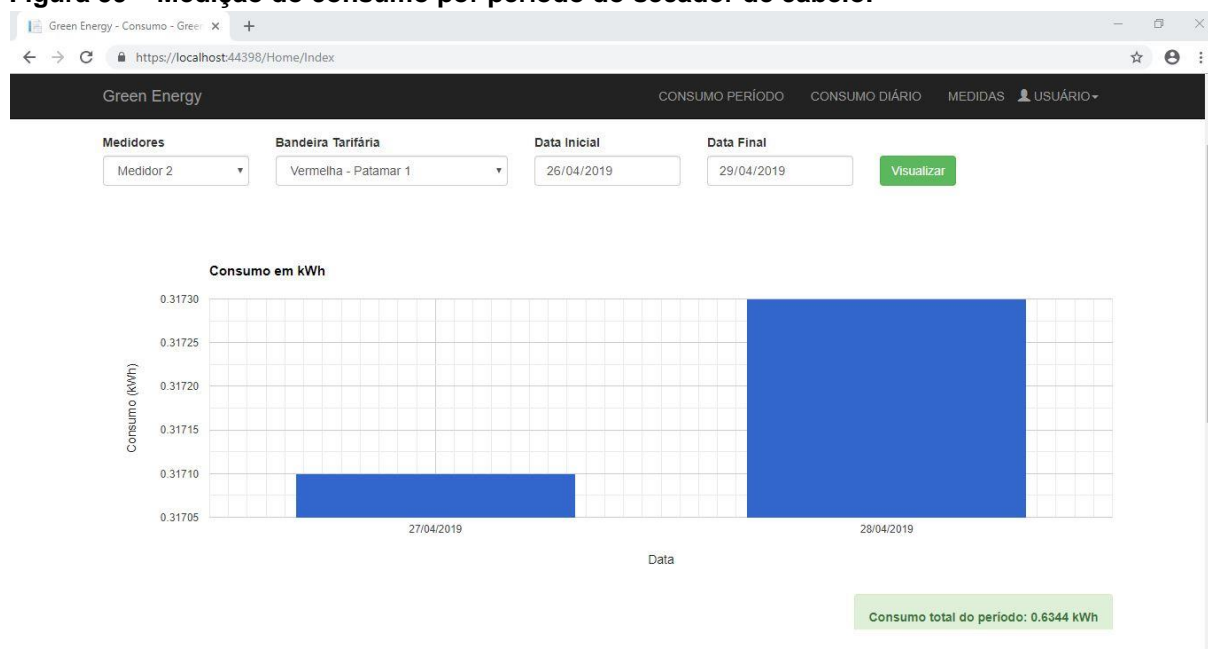
Fonte: Autoria própria.

**Figura 58 - Medição do fator de potência ferro de passar.**

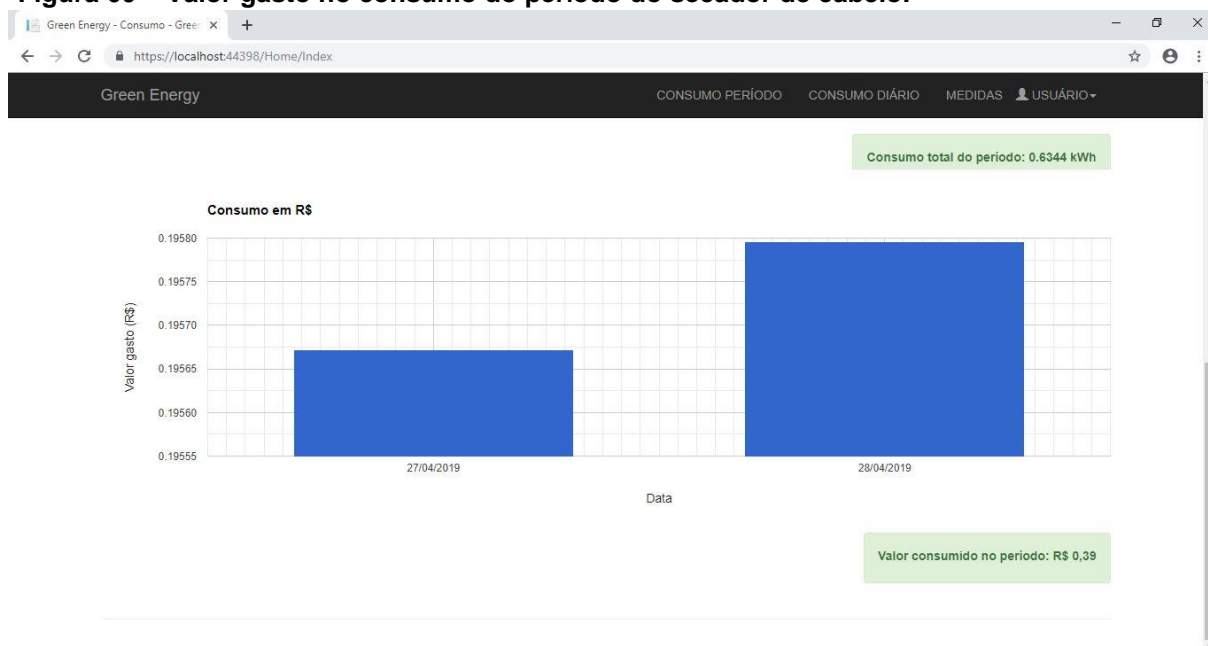


Fonte: Autoria própria.

A segunda carga analisada foi um secador de cabelo modelo Arno Power Dry com potência de 1900W. A seguir são mostradas as figuras com os resultados da monitoração.

**Figura 59 – Medição do consumo por período do secador de cabelo.**

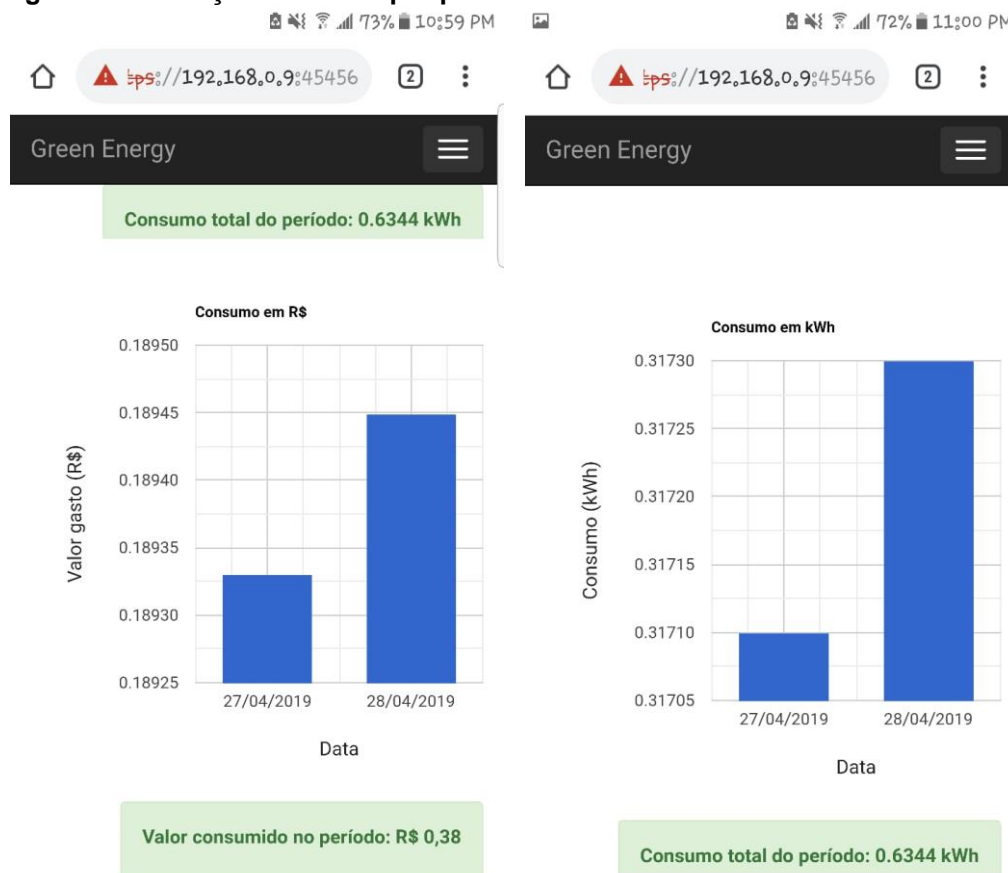
Fonte: Autoria própria.

**-Figura 60 – Valor gasto no consumo do período do secador de cabelo.**

Fonte: Autoria própria.

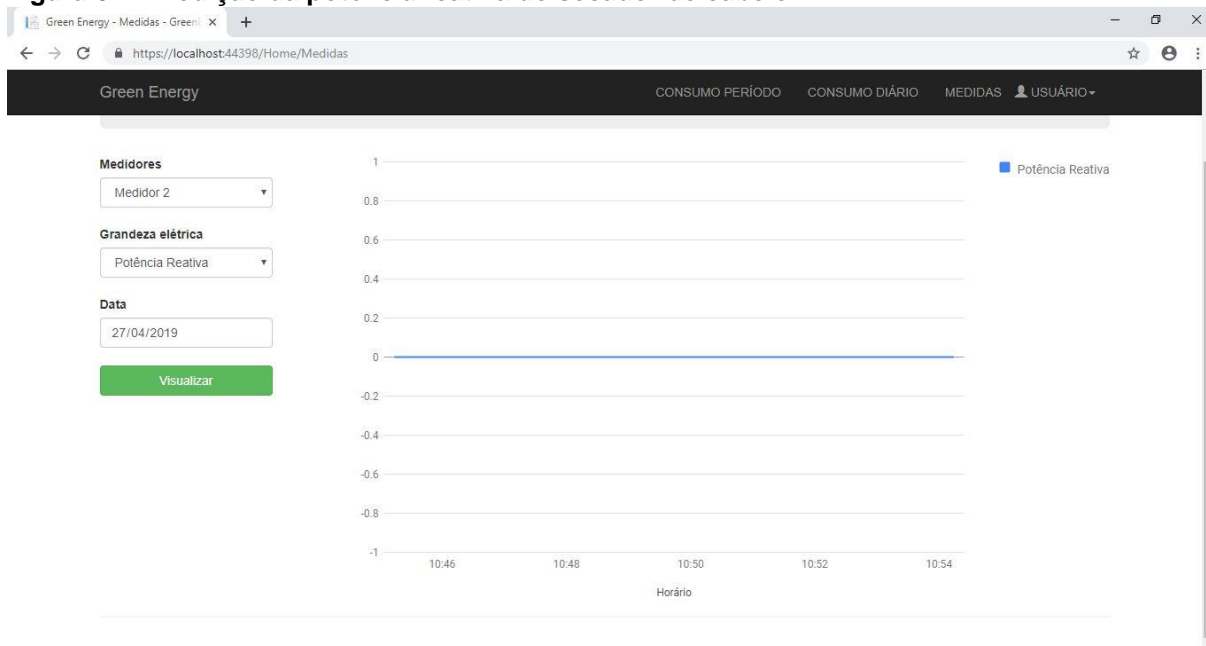


**Figura 61 – Medição consumo por período do secador de cabelo na tela do celular.**



Fonte: Autoria própria.

**Figura 62 – Medição da potência reativa do secador de cabelo.**



Fonte: Autoria própria.

## 5.1 Custos do projeto

A Tabela 6 contém o preço dos componentes utilizados no projeto, dando uma ideia geral do quanto foi gasto para a montagem do protótipo. É importante frisar que não foram considerados os custos de desenvolvimento.

**Tabela 6 – Custos do projeto.**

<b>Carga</b>	<b>Qtd.</b>	<b>Preço Unitário (R\$)</b>
Microcontrolador	1	25,00
Chip ADE7753	3	9,10
Sensor corrente SCT-013-050	3	21,80
Módulo Wi-Fi ESP8266 12E	1	6,80
Transformador 110/220V – 9V	1	22,00
Conector borne 2 vias	4	0,80
Regulador de tensão 5V 7805	1	1,80
Componentes em SMD	1	5,00
Demais componentes	1	4,50
	<b>Total</b>	161,00

**Fonte: Autoria própria.**

## 6. CONSIDERAÇÕES FINAIS

O desenvolvimento do medidor foi motivado pela necessidade de conscientização e controle da energia elétrica consumida nos lares brasileiros, pois cada vez mais se fala da escassez dos recursos e dos impactos ambientais gerados para atender a demanda de energia. No entanto, no Brasil a cultura de economia ainda não é consolidada, mas projetos como esse podem despertar o hábito no consumidor e ser uma das soluções para o atual cenário.

O presente trabalho descreve todo o processo do desenvolvimento de um sistema de supervisão do consumo de energia elétrica. Na fase de testes, constatou-se que as comunicações entre os dispositivos funcionaram conforme esperado. Para o fim do projeto proposto, os erros encontrados nas medidas de corrente se mostram satisfatórios, pois o mesmo foi projetado para que o consumidor tenha uma ideia bem aproximada do valor real que ele vai gastar na sua conta de energia. Porém, para melhor análise dos erros de medição, seria interessante realizar mais testes de medidas, fazendo comparações com analisadores de energia. Além do mais, os erros encontrados nas medições podem ser melhorados, tendo em vista que o ADE7753 possui canais de calibração, aumentando assim a sua precisão.

O método de acesso remoto ao sistema revelou ser de extrema utilidade. Ele permite que os usuários tenham o acesso às informações através de uma página da *internet*, podendo ser acessado em qualquer dispositivo que contenha um *browser*, sendo que este *software* vem incluído na maioria dos sistemas operativos.

Em relação ao custo total do projeto, alguns componentes possuem a possibilidade de redução do valor quando comprados em grande quantidade para a produção em série.

### 6.1. Sugestão de trabalhos futuros

Ainda que este sistema possua funcionalidades interessantes, o tema proposto na dissertação é de grande abrangência, podendo assim serem aperfeiçoadas as funcionalidades já existentes, bem como desenvolvidas e incorporadas outras, tais como:

- Operação remota de funcionalidades de desligar e ligar tomadas e interruptores;
- Exploração das funcionalidades de qualidade de energia do circuito integrado ADE7753;
- Possibilidade de usar o ADE7758 com três canais, para medições bifásicas/trifásicas;
- Utilizar divisores resistivos no lugar de transformadores na entrada de tensão do ADE para baratear os custos do projeto;
- Estudar a possibilidade de projetar uma fonte chaveada para diminuir o consumo e custo do projeto;
- Integração do projeto com outros tipos de tecnologias, tais como: sensores de alarmes, dispositivos de comunicação;
- Criação de relatórios de consumo e de metas para o consumidor;
- Melhorias na parte de segurança das informações.

## REFERÊNCIAS

ABESCO. Associação Brasileira de Empresas de Serviço de Conservação de Energia. **Desperdício de energia custou mais de R\$ 60 bi para o Brasil nos últimos três anos.** Disponível em: < <http://www.abesco.com.br/pt/novidade/desperdicio-de-energia-custou-mais-de-r-60-bi-para-o-brasil-nos-ultimos-tres-anos/>>. Acesso em: 23 de janeiro de 2018.

ALEXANDER, Charles K.; SADIKU, Matthew N. O. **Fundamentos de circuitos elétricos.** 5ª ed. Porto Alegre: AMGH, 2013.

ANALOG DEVICES. **Single phase multifunction metering IC with di/dt sensor interface.** Disponível em: <<https://www.analog.com/media/en/technical-documentation/data-sheets/ADE7753.pdf>>. Acesso em: 20 de outubro de 2017.

ANEEL. Agência Nacional de Energia Elétrica. **BIG - Banco de Informações de Geração.** Disponível em: <<http://www.aneel.gov.br/aplicacoes/capacidadebrasil/capacidadebrasil.cfm>>. Acesso em: 17 abril 2017.

ANEEL. Agência Nacional de Energia Elétrica. **Entendendo a Tarifa.** Disponível em: <[http://www.aneel.gov.br/entendendo-a-tarifa/-/asset\\_publisher/uQ5pCGhnyj0y/content/composicao-da-tarifa/654800?inheritRedirect=false&redirect=http://www.aneel.gov.br/entendendo-a-tarifa?p\\_p\\_id=101\\_INSTANCE\\_uQ5pCGhnyj0y&p\\_p\\_lifecycle=0&p\\_p\\_state=normal&p\\_p\\_mode=view](http://www.aneel.gov.br/entendendo-a-tarifa/-/asset_publisher/uQ5pCGhnyj0y/content/composicao-da-tarifa/654800?inheritRedirect=false&redirect=http://www.aneel.gov.br/entendendo-a-tarifa?p_p_id=101_INSTANCE_uQ5pCGhnyj0y&p_p_lifecycle=0&p_p_state=normal&p_p_mode=view)>. Acesso em: 25 nov. 2017.

ANEEL. Agência Nacional de Energia Elétrica. **Resolução normativa nº 414.** Disponível em: <[http://www2.aneel.gov.br/aplicacoes/audiencia/arquivo/2012/065/documento/ren\\_-\\_alteracoes\\_-\\_fp\\_-\\_11-06-2012.pdf](http://www2.aneel.gov.br/aplicacoes/audiencia/arquivo/2012/065/documento/ren_-_alteracoes_-_fp_-_11-06-2012.pdf)> Acesso em: 02 abril 2018.

ANEEL. Agência Nacional de Energia Elétrica. **Tarifa branca é a nova opção para os consumidores a partir de 2018**. Disponível em: <[http://www.aneel.gov.br/sala-de-imprensa-exibicao/-/asset\\_publisher/XGPXSqdMFHrE/content/tarifa-branca-e-nova-opcao-para-os-consumidores-a-partir-de-2018/656877?inheritRedirect=false](http://www.aneel.gov.br/sala-de-imprensa-exibicao/-/asset_publisher/XGPXSqdMFHrE/content/tarifa-branca-e-nova-opcao-para-os-consumidores-a-partir-de-2018/656877?inheritRedirect=false)>.

Acesso em: 03 abril 2018.

AYA, T. H. I.; EDGAR, M. M. U. **Análise dos impactos ambientais na produção de energia**. Instituto de Eletrotécnica e Energia da Universidade de São Paulo (IEE/USP). São Paulo, p. 5. 2005.

CEBALLOS, José J. O. **Clases de corriente eléctrica**. Disponível em: <<https://eudotec.wordpress.com/tag/clases-de-corriente-electrica/>>. Acesso em: 20 de janeiro de 2018.

CHAPMAN, Stephen J. **Fundamentos de máquinas elétricas**. 5ª ed. Porto Alegre: AMGH, 2013.

CIRCUIT BASICS. **Basic of UART communication**. Disponível em: <<http://www.circuitbasics.com/basics-uart-communication/>> Acesso em: 10 de março de 2018.

COPEL. **Entenda as bandeiras tarifárias que começam em janeiro**. Disponível em: <<http://www.copel.com/hpcopel/root/nivel2.jsp?endereco=%2Fhpcopel%2Froot%2Fpagcopel2.nsf%2Fdocs%2FC2F891926071E8F203257DC50046A10B>>. Acesso em: 19 abril 2017.

DAHLE, David. **Historical time line of electric meters**. Disponível em: <<http://watthourmeters.com/history.html>>. Acesso em: 06 de maio de 2017.

DOUGLAS, Allan. **Introdução à Google Chart Tools**. Disponível em: <<https://www.devmedia.com.br/introducao-a-google-chart-tools/26453>>. Acesso em: 16 de junho de 2018.

ELEKTRO. **Energia ativa e reativa**. Disponível em: <<https://www.elektro.com.br/seu-negocio/energia-ativa-e-reativa>>. Acesso em: 02 de abril de 2018.

ENERGISA. **Medidor eletrônico**. Disponível em: <<https://www.energisa.com.br/Paginas/informacoes/dicas/medidor-eletronico.aspx>>. Acesso em: 02 de abril de 2018.

ENGELÉTRICA. **Manual de correção do fator de potência**. Disponível em: <<http://engeletrica.com.br/fatordepotencia-manual-fatordepotencia.html>>. Acesso em: 29 de março de 2018.

ESPRESSIF. **ESP-12E Wi-Fi Module**. Disponível em: <<https://www.kloppenborg.net/images/blog/esp8266/esp8266-esp12e-specs.pdf>>. Acesso em: 09 de maio de 2017.

FILIFELOP. **Módulo WiFi ESP8266 ESP-12**. Disponível em: <<http://www.filieflop.com/pd-2c1441-modulo-wifi-esp8266-esp-12e.html>>. Acesso em: 02 de maio de 2017.

GALVÃO, Jucilene; BERMANN, Célio. **Crise hídrica e energia: conflitos no uso múltiplo das águas**. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0103-40142015000200043](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-40142015000200043)>. Acesso em: 19 abril 2017.

MICROCHIP. **Microchip PIC18F2420/2520/4420/4520 datasheet**. Disponível em: <<http://ww1.microchip.com/downloads/en/devicedoc/39631e.pdf>>. Acesso em: 09 de maio de 2017.

MICROSOFT. **Visão geral do IDE do Visual Studio**. Disponível em: <<https://docs.microsoft.com/pt-br/visualstudio/ide/visual-studio-ide#extend-visual-studio>>. Acesso em: 29 de março de 2018.

MORO, Jefferson Z. **Medidor de consumo de energia elétrica descentralizado e com interface web**. Campinas 2013. Dissertação de Mestrado em Engenharia Elétrica. Universidade Estadual de Campinas.

MUXFELDT, Pedro. **Como funciona o Wi-Fi**. Disponível em: <<https://br.ccm.net/contents/790-como-funciona-o-wi-fi>>. Acesso em: 08 de junho de 2019.

NILSSON, James W. **Circuitos elétricos**. 8ª ed. São Paulo: Pearson Prentice Hall, 2009.

NOURIA, Lucia M. **O que é Internet?**. Disponível em: <<https://br.ccm.net/faq/12069-o-que-e-internet>>. Acesso em: 13 de junho de 2019.

O que é TCP/IP e como funciona. **Infonova**, 2018. Disponível em: <<https://www.infonova.com.br/artigo/o-que-e-tcp-ip-e-como-funciona/>>. Acesso em: 09 de junho de 2019.

PACIEVITC, Yuri. **MySQL**. Disponível em: <<https://www.infoescola.com/informatica/mysql/>>. Acesso em: 29 de março de 2018.



PROCEL. **Lei de eficiência energética.** Disponível em: <<http://www.procelinfo.com.br/resultadosprocel2014/lei.pdf>>. Acesso em: 24 de abril de 2017.

SACCO, Francisco. **Comunicação SPI – Parte1.** Disponível em: <<https://www.embarcados.com.br/spi-parte-1/>>. Acesso em: 20 de março de 2018.

SILVA, Tarcisio. **Introdução ao cadsoft Eagle.** Disponível em: <<https://www.embarcados.com.br/tutorial-eagle-introducao/>>. Acesso em: 15 de outubro de 2017.

VICTORIO, Julio. **Requisição GET e POST – principais diferenças.** Disponível em: <<http://tecnologiaeinovacao.com.br/blog/2016/07/24/requisicoes-get-e-post-principais-diferencas/>>. Acesso em: 07 de abril de 2018.

YHDC. **Single phase multifunction metering IC with di/dt sensor interface.** Disponível em: <<https://nicegear.nz/obj/pdf/SCT-050-datasheet.pdf>>. Acesso em: 15 de outubro de 2017.

## APÊNDICE A – Código do PIC18F4520.

```
//=====
// main.c

#include <xc.h>
#include <plib/usart.h>
#include <plib/pconfig.h>
#include <plib/spi.h>
#include <delays.h>
#include "esp8266.h"
#include "ade7753.h"
#include "main.h"

#pragma config OSC = INTIO67
#pragma config MCLRE = OFF
#pragma config WDT = OFF

/* FUNÇÃO DE DELAY PARA CLOCK DE 8MHZ*/
void MSdelay(unsigned int val)
{
    unsigned int i,j;
    for(i = 0; i <= val; i++)
        for(j = 0; j < 165; j++);           //1 ms delay para frequência de 8MHz
}

/* FUNÇÃO DE INTERRUPÇÃO DA USART*/
void interrupt high_isr(void)
{
    if (PIR1bits.RCIF)                     // Dado recebido pela USART
    {
        RESPONSE_BUFFER[Counter] = RCREG; // Copia o dado para o buffer e incrementa o
        contador
        Counter++;

        if(RCSTAbits.OERR)                 // verifica se alguma colisão ocorre durante à
        recepção contínua
        {
            RCSTAbits.CREN = 0;           // Limpa o erro
            NOP();
            RCSTAbits.CREN = 1;           // Habilita o receptor
        }

        if(Counter == DEFAULT_BUFFER_SIZE) {
            Counter = 0;
            pointer = 0;
        }

        PIR1bits.RCIF = 0;                // Limpa a flag de interrupção
    }
}

void Configura_USART(void)
{
    TRISCbits.TRISC6 = 0; // Porta C6 UART TX
    TRISCbits.TRISC7 = 1; // Porta C7 UART RX
}
```

```

baudUSART(
    BAUD_IDLE_CLK_LOW &
    BAUD_IDLE_RX_PIN_STATE_HIGH &
    BAUD_16_BIT_RATE &
    BAUD_WAKEUP_OFF &
    BAUD_AUTO_OFF);

OpenUSART(
    USART_TX_INT_OFF &
    USART_RX_INT_ON &
    USART_ASYNC_MODE &
    USART_EIGHT_BIT &
    USART_CONT_RX &
    USART_ADDEN_OFF &
    USART_BRGH_HIGH,
    16);

//CONFIGURAÇÕES DE INTERRUPTÃO DA USART
RCONbits.IPEN = 1; // Habilita prioridades de interrupção
INTCON = 0x80; // Habilita interrupção global
INTCON3 = 0x00; // Desabilita interrupções INT*
IPR1bits.RC1IP = 1; // Interrupção da EUSART é de alta prioridade
}

void Configura_Oscilador(void)
{
    OSCCONbits.IRCF2 = 1;
    OSCCONbits.IRCF1 = 1;
    OSCCONbits.IRCF0 = 1; // Seleciona clock interno de 8MHz
    while(!OSCCONbits.IOFS); // Espera o oscilador estabilizar
}

void Configura_SPI(void)
{
    TRISCbits.TRISC5 = 0; //configura SDO como saída
    TRISCbits.TRISC4 = 1; //configura SDI como entrada
    TRISCbits.TRISC3 = 0; //configura SCK como saída
    TRISBbits.TRISB3 = 0; //configura CS3 como saída
    TRISBbits.TRISB4 = 0; //configura CS2 como saída
    TRISBbits.TRISB5 = 0; //configura CS1 como saída
    ADE7753_CS1 = DESABILITA_CS; //desabilita chip select 1
    ADE7753_CS2 = DESABILITA_CS; //desabilita chip select 2
    ADE7753_CS3 = DESABILITA_CS; //desabilita chip select 3
    SSPCON1 = 0b00100010;
    SSPSTAT = 0b11000000;
}

void Configura_Registradores(void)
{
    TRISAbits.TRISA0 = 0;
    PORTBbits.RB5 = 1;
    PORTBbits.RB4 = 1;
    PORTBbits.RB3 = 1;
    ADCON1 = 0x0F;
}

void Configura_ADE(void)
{
    ADE7753_CS1 = HABILITA_CS;
    WriteSPI(0x89);
    MSdelay(10);
    WriteSPI(0x80A4);
}

```

```

MSdelay(10);
WriteSPI(0x9C);
MSdelay(10);
WriteSPI(0x07D0);
MSdelay(10);
WriteSPI(0x92);
MSdelay(10);
WriteSPI(0x0000);
ADE7753_CS1 = DESABILITA_CS;

ADE7753_CS2 = HABILITA_CS;
WriteSPI(0x89);
MSdelay(10);
WriteSPI(0x80A4);
MSdelay(10);
WriteSPI(0x9C);
MSdelay(10);
WriteSPI(0x07D0);
MSdelay(10);
WriteSPI(0x92);
MSdelay(10);
WriteSPI(0x0000);
ADE7753_CS2 = DESABILITA_CS;

ADE7753_CS3 = HABILITA_CS;
WriteSPI(0x89);
MSdelay(10);
WriteSPI(0x80A4);
MSdelay(10);
WriteSPI(0x9C);
MSdelay(10);
WriteSPI(0x07D0);
MSdelay(10);
WriteSPI(0x92);
MSdelay(10);
WriteSPI(0x0000);
ADE7753_CS3 = DESABILITA_CS;
}

void Configura_ESP8862(void)
{
    while(!ESP8266_Begin());
        ESP8266_WIFIMode(BOTH_STATION_AND_ACCESPOINT); /* 3 = Both (AP and STA) */
        ESP8266_ConnectionMode(SINGLE); /* 0 = Single; 1 = Multi */
        ESP8266_ApplicationMode(NORMAL); /* 0 = Normal Mode; 1 =
Transperant Mode */
        if(ESP8266_connected() == ESP8266_NOT_CONNECTED_TO_AP)
            ESP8266_JoinAccessPoint(SSID, PASSWORD);
            ESP8266_Start(0, DOMAIN, PORT);
}

void VerificaConexao(void)
{
    uint8_t Connect_Status;
    Connect_Status = ESP8266_connected();
    if(Connect_Status == ESP8266_NOT_CONNECTED_TO_AP)
        ESP8266_JoinAccessPoint(SSID, PASSWORD);
    if(Connect_Status == ESP8266_TRANSMISSION_DISCONNECTED)
        ESP8266_Start(0, DOMAIN, PORT);
}

```

```

void main(void) {

    char _buffer[150];
    long potenciaAtiva, potenciaReativa, potenciaAparente, corrente, tensao;

    //Configura os registradores
    Configura_Registradores();

    //Configura a USART
    Configura_USART();

    //Configura a comunicação SPI
    Configura_SPI();

    //Configura o oscilador interno
    Configura_Oscilador();

    //Configura o ADE
    Configura_ADE();

    //Configura o ESP8266
    Configura_ESP8862();

    MSdelay(10);

    while(1)
    {
        PORTAbits.RA0 = 1;

        VerificaConexao();

        //Medidor 1
        memset(_buffer, 0, 150);
        potenciaAtiva = Medidor1_ADE7753_Read24(_AENERGY)*0,005581;
        potenciaReativa = Medidor1_ADE7753_Read24(_LVARENERGY)*0,02589;
        potenciaAparente = Medidor1_ADE7753_Read24(_VAENERGY)*0,003546;
        corrente = Medidor1_ADE7753_Read24(_IRMS)*0,0000185;
        tensao = Medidor1_ADE7753_Read24(_VRMS)*0,0001428;
        sprintf(_buffer, "GET
/Home/SalvarConsumo?idMedidor=1&energiaAtiva=%s&potenciaReativa=%s&energiaAparente=%s&
corrente=%s&tensao=%s", potenciaAtiva, potenciaReativa, potenciaAparente, corrente, tensao);

        //Medidor 2
        memset(_buffer, 0, 150);
        potenciaAtiva = Medidor2_ADE7753_Read24(_AENERGY)*0,005581;
        potenciaReativa = Medidor2_ADE7753_Read24(_LVARENERGY)*0,02589;
        potenciaAparente = Medidor2_ADE7753_Read24(_VAENERGY)*0,003546;
        corrente = Medidor2_ADE7753_Read24(_IRMS)*0,0000185;
        tensao = Medidor2_ADE7753_Read24(_VRMS)*0,0001428;
        sprintf(_buffer, "GET
/Home/SalvarConsumo?idMedidor=2&energiaAtiva=%s&potenciaReativa=%s&energiaAparente=%s&
corrente=%s&tensao=%s", potenciaAtiva, potenciaReativa, potenciaAparente, corrente, tensao);
        ESP8266_Send(_buffer);
        MSdelay(3600);

        PORTAbits.RA0 = 0;
    }

    return;
}

```

```
//=====
// main.h

#ifndef MAIN_H
#define MAIN_H

#ifdef __cplusplus
extern "C" {
#endif

#ifdef __cplusplus
}
#endif

#endif /* MAIN_H */

#include <string.h>
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <stdbool.h>

#include <xc.h>

/* Conexão Wi-Fi */
#define DOMAIN          "192.168.0.9"
#define PORT            "45456"
#define SSID            "IRACEMA CACERES"
#define PASSWORD        "86698300"

/* Pinos comunicação SPI */
#define ADE7753_CS1     PORTBbits.RB5 // Porta do chip select 1
#define ADE7753_CS2     PORTBbits.RB4 // Porta do chip select 2
#define ADE7753_CS3     PORTBbits.RB3 // Porta do chip select 3

void MSdelay(unsigned int val); // Função delay milisegundo
```

## APÊNDICE B – Código do ESP8266.

```
//=====
// esp8266.c

#include "esp8266.h"
#include <string.h>
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <stdbool.h>

void Read_Response(const char* _Expected_Response)
{
    uint8_t EXPECTED_RESPONSE_LENGTH = strlen(_Expected_Response);
    uint32_t TimeCount = 0, ResponseBufferLength;
    char RECEIVED_CRLF_BUF[30];

    while(1)
    {
        if(TimeCount >= (DEFAULT_TIMEOUT + TimeOut) )
        {
            TimeOut = 0;
            Response_Status = ESP8266_RESPONSE_TIMEOUT;
            return;
        }

        if(Response_Status == ESP8266_RESPONSE_STARTING)
        {
            Response_Status = ESP8266_RESPONSE_WAITING;
        }

        ResponseBufferLength = strlen(RESPONSE_BUFFER);

        if (ResponseBufferLength)
        {
            MSdelay(1);
            TimeCount++;

            if (ResponseBufferLength == strlen(RESPONSE_BUFFER))
            {
                for (uint16_t i = 0; i < ResponseBufferLength; i++)
                {
                    memmove(RECEIVED_CRLF_BUF, RECEIVED_CRLF_BUF
+ 1, EXPECTED_RESPONSE_LENGTH - 1);
                    RECEIVED_CRLF_BUF[EXPECTED_RESPONSE_LENGTH
- 1] = RESPONSE_BUFFER[i];

                    if(!strcmp(RECEIVED_CRLF_BUF, _Expected_Response,
EXPECTED_RESPONSE_LENGTH))
                    {
                        TimeOut = 0;
                        Response_Status =
ESP8266_RESPONSE_FINISHED;
                        return;
                    }
                }
            }
        }
    }
}
```

```

        MSdelay(1);
        TimeCount++;
    }
}

void ESP8266_Clear()
{
    memset(RESPONSE_BUFFER, 0, DEFAULT_BUFFER_SIZE);
    Counter = 0;
    pointer = 0;
}

void Start_Read_Response(const char* _ExpectedResponse)
{
    Response_Status = ESP8266_RESPONSE_STARTING;
    do {
        Read_Response(_ExpectedResponse);
    } while(Response_Status == ESP8266_RESPONSE_WAITING);
}

void GetResponseBody(char* Response, uint16_t ResponseLength)
{
    uint16_t i = 12;
    char buffer[5];
    while(Response[i] != '\r')
        ++i;

    strncpy(buffer, Response + 12, (i - 12));
    ResponseLength = atoi(buffer);

    i += 2;
    uint16_t tmp = strlen(Response) - i;
    memcpy(Response, Response + i, tmp);

    if(!strncmp(Response + tmp - 6, "\r\nOK\r\n", 6))
        memset(Response + tmp - 6, 0, i + 6);
}

bool WaitForExpectedResponse(const char* ExpectedResponse)
{
    Start_Read_Response(ExpectedResponse);    /* First read response */
    if((Response_Status != ESP8266_RESPONSE_TIMEOUT))
        return true;                          /* Return true for success */
    return false;                             /* Else return false */
}

void ESP8266_TxChar(char data)
{
    while(!TRMT); // Wait here till transmission is complete
    TXREG = data; // Load the transmitter buffer with the received value*/
}

void ESP8266_SendString(const char *str)
{
    while(*str != '\0') /* Transmit data until null */
    {
        ESP8266_TxChar(*str);
        str++;
    }
}

```



```

bool SendATandExpectResponse(char* ATCommand, const char* ExpectedResponse)
{
    ESP8266_Clear();
    ESP8266_SendString(ATCommand);           /* Send AT command to ESP8266 */
    ESP8266_SendString("\r\n");
    return WaitForExpectedResponse(ExpectedResponse);
}

bool ESP8266_ApplicationMode(uint8_t Mode)
{
    char _atCommand[20];
    memset(_atCommand, 0, 20);
    sprintf(_atCommand, "AT+CIPMODE=%d", Mode);
    _atCommand[19] = 0;
    return SendATandExpectResponse(_atCommand, "\r\nOK\r\n");
}

bool ESP8266_ConnectionMode(uint8_t Mode)
{
    char _atCommand[20];
    memset(_atCommand, 0, 20);
    sprintf(_atCommand, "AT+CIPMUX=%d", Mode);
    _atCommand[19] = 0;
    return SendATandExpectResponse(_atCommand, "\r\nOK\r\n");
}

bool ESP8266_Begin()
{
    for (uint8_t i = 0; i < 5; i++)
    {
        if(SendATandExpectResponse("ATE0","\r\nOK\r\n")||SendATandExpectResponse("AT","\r\nOK\r\n"))
            return true;
        }
    return false;
}

bool ESP8266_Close()
{
    return SendATandExpectResponse("AT+CIPCLOSE=1", "\r\nOK\r\n");
}

bool ESP8266_WIFIMode(uint8_t _mode)
{
    char _atCommand[20];
    memset(_atCommand, 0, 20);
    sprintf(_atCommand, "AT+CWMODE=%d", _mode);
    _atCommand[19] = 0;
    return SendATandExpectResponse(_atCommand, "\r\nOK\r\n");
}

uint8_t ESP8266_JoinAccessPoint(const char* _SSID, const char* _PASSWORD)
{
    char _atCommand[60];
    memset(_atCommand, 0, 60);
    sprintf(_atCommand, "AT+CWJAP=\"%s\", \"%s\"", _SSID, _PASSWORD);
    _atCommand[59] = 0;

    if(SendATandExpectResponse(_atCommand, "\r\nWIFI CONNECTED\r\n")) {
        return ESP8266_WIFI_CONNECTED;
    }
}

```

```

    }
    else {

        if(strstr(RESPONSE_BUFFER, "+CWJAP:1"))
            return ESP8266_CONNECTION_TIMEOUT;
        else if(strstr(RESPONSE_BUFFER, "+CWJAP:2"))
            return ESP8266_WRONG_PASSWORD;
        else if(strstr(RESPONSE_BUFFER, "+CWJAP:3"))
            return ESP8266_NOT_FOUND_TARGET_AP;
        else if(strstr(RESPONSE_BUFFER, "+CWJAP:4"))
            return ESP8266_CONNECTION_FAILED;
        else
            return ESP8266_JOIN_UNKNOWN_ERROR;
    }
}

uint8_t ESP8266_connected()
{
    SendATandExpectResponse("AT+CIPSTATUS", "\r\nOK\r\n");

    if(strstr(RESPONSE_BUFFER, "STATUS:2"))
        return ESP8266_CONNECTED_TO_AP;
    else if(strstr(RESPONSE_BUFFER, "STATUS:3"))
        return ESP8266_CREATED_TRANSMISSION;
    else if(strstr(RESPONSE_BUFFER, "STATUS:4"))
        return ESP8266_TRANSMISSION_DISCONNECTED;
    else if(strstr(RESPONSE_BUFFER, "STATUS:5"))
        return ESP8266_NOT_CONNECTED_TO_AP;
    else
        return ESP8266_CONNECT_UNKNOWN_ERROR;
}

uint8_t ESP8266_Start(uint8_t _ConnectionNumber, const char* Domain, const char* Port)
{
    bool _startResponse;
    char _atCommand[60];
    memset(_atCommand, 0, 60);
    _atCommand[59] = 0;

    if(SendATandExpectResponse("AT+CIPMUX?", "CIPMUX:0"))
        sprintf(_atCommand, "AT+CIPSTART=\"%s\", \"%s\", %s", Domain, Port);
    else
        sprintf(_atCommand, "AT+CIPSTART=\"%d\", \"%s\", \"%s\", %s",
        _ConnectionNumber, Domain, Port);

    _startResponse = SendATandExpectResponse(_atCommand, "CONNECT OK\r\n");

    if(!_startResponse)
    {
        if(Response_Status == ESP8266_RESPONSE_TIMEOUT)
            return ESP8266_RESPONSE_TIMEOUT;
        return ESP8266_RESPONSE_ERROR;
    }

    return ESP8266_RESPONSE_FINISHED;
}

uint8_t ESP8266_Send(char* Data)
{
    char _atCommand[20];
    memset(_atCommand, 0, 20);

```

```

    sprintf(_atCommand, "AT+CIPSEND=%d", (strlen(Data)+2));
    _atCommand[19] = 0;

    SendATandExpectResponse(_atCommand, "\r\nOK\r\n>");

    if(!SendATandExpectResponse(Data, "\r\nSEND OK\r\n"))
    {
        if(Response_Status == ESP8266_RESPONSE_TIMEOUT)
            return ESP8266_RESPONSE_TIMEOUT;
        return ESP8266_RESPONSE_ERROR;
    }

    return ESP8266_RESPONSE_FINISHED;
}

int16_t ESP8266_DataAvailable()
{
    return (Counter - pointer);
}

uint8_t ESP8266_DataRead()
{
    if(pointer < Counter){
        return RESPONSE_BUFFER[pointer++];
    }
    else {
        ESP8266_Clear();
        return 0;
    }
}

uint16_t Read_Data(char* _buffer)
{
    uint16_t len = 0;
    MSdelay(100);

    while(ESP8266_DataAvailable() > 0)
        _buffer[len++] = ESP8266_DataRead();

    return len;
}

//=====
// esp8266.h

#ifndef ESP8266_H
#define ESP8266_H

#ifdef __cplusplus
extern "C" {
#endif

#ifdef __cplusplus
}
#endif

#endif /* ESP8266_H */

#include "main.h"

```

```

#define DEFAULT_BUFFER_SIZE          160
#define DEFAULT_TIMEOUT              10000

/* Connection Mode */
#define SINGLE                       0
#define MULTIPLE                     1

/* Application Mode */
#define NORMAL                       0
#define TRANSPERANT                  1

/* Application Mode */
#define STATION                      1
#define ACCESSPOINT                  2
#define BOTH_STATION_AND_ACCESPOINT 3

enum ESP8266_RESPONSE_STATUS{
    ESP8266_RESPONSE_WAITING,
    ESP8266_RESPONSE_FINISHED,
    ESP8266_RESPONSE_TIMEOUT,
    ESP8266_RESPONSE_BUFFER_FULL,
    ESP8266_RESPONSE_STARTING,
    ESP8266_RESPONSE_ERROR
};

enum ESP8266_CONNECT_STATUS {
    ESP8266_CONNECTED_TO_AP,
    ESP8266_CREATED_TRANSMISSION,
    ESP8266_TRANSMISSION_DISCONNECTED,
    ESP8266_NOT_CONNECTED_TO_AP,
    ESP8266_CONNECT_UNKNOWN_ERROR
};

enum ESP8266_JOINAP_STATUS {
    ESP8266_WIFI_CONNECTED,
    ESP8266_CONNECTION_TIMEOUT,
    ESP8266_WRONG_PASSWORD,
    ESP8266_NOT_FOUND_TARGET_AP,
    ESP8266_CONNECTION_FAILED,
    ESP8266_JOIN_UNKNOWN_ERROR
};

int8_t Response_Status;
volatile int16_t Counter = 0, pointer = 0;
uint32_t TimeOut = 0;
char RESPONSE_BUFFER[DEFAULT_BUFFER_SIZE];

void ESP8266_TxChar(char data);
void ESP8266_SendString(const char *str);

void Read_Response(const char* _Expected_Response);
void Start_Read_Response(const char* _ExpectedResponse);
void GetResponseBody(char* Response, uint16_t ResponseLength);
void ESP8266_Clear();

bool WaitForExpectedResponse(const char* ExpectedResponse);
bool SendATandExpectResponse(char* ATCommand, const char* ExpectedResponse);
bool ESP8266_ApplicationMode(uint8_t Mode);
bool ESP8266_ConnectionMode(uint8_t Mode);
bool ESP8266_WIFIMode(uint8_t _mode);

```

```
uint8_t ESP8266_JoinAccessPoint(const char* _SSID, const char* _PASSWORD);  
uint8_t ESP8266_connected();  
uint8_t ESP8266_Start(uint8_t _ConnectionNumber, const char* Domain, const char* Port);  
uint8_t ESP8266_Send(char* Data);  
uint8_t ESP8266_DataRead();  
  
int16_t ESP8266_DataAvailable();  
  
bool ESP8266_Begin();  
bool ESP8266_Close();
```

## APÊNDICE C – Código do ADE7753.

```
//=====
// ade7753.c

#include "main.h"
#include <p18f4520.h>
#include "ade7753.h"

unsigned char spi_transfer(unsigned char data)
{
    SSPBUF = data;
    while(!SSPSTATbits.BF);
    data = SSPBUF;
    return data;
}

//MEDIDOR 1
unsigned char Medidor1_ADE7753_Read8(unsigned char reg)
{
    ADE7753_CS1 = HABILITA_CS;
    unsigned char b0;
    MSdelay(10);
    b0 = WriteSPI(reg);
    MSdelay(10);
    ADE7753_CS1 = DESABILITA_CS;

    return b0;
}

uint16_t Medidor1_ADE7753_Read16(unsigned char reg)
{
    ADE7753_CS1 = HABILITA_CS;
    unsigned char b1, b0;
    MSdelay(10);
    b1 = spi_transfer(reg);
    MSdelay(10);
    b0 = spi_transfer(reg);
    ADE7753_CS1 = DESABILITA_CS;

    return (uint16_t)b1<<8 | (uint16_t)b0;
}

uint32_t Medidor1_ADE7753_Read24(unsigned char reg)
{
    ADE7753_CS1 = HABILITA_CS;
    unsigned char b2, b1, b0;
    MSdelay(10);
    b2 = spi_transfer(reg);
    MSdelay(10);
    b1 = spi_transfer(reg);
    MSdelay(10);
    b0 = spi_transfer(reg);
    ADE7753_CS1 = DESABILITA_CS;

    return (uint32_t)b2<<16 | (uint32_t)b1<<8 | (uint32_t)b0;
}

//MEDIDOR 2
unsigned char Medidor2_ADE7753_Read8(unsigned char reg)
```

```

{
  ADE7753_CS2 = HABILITA_CS;
  unsigned char b0;
  MSdelay(10);
  b0 = WriteSPI(reg);
  MSdelay(10);
  ADE7753_CS2 = DESABILITA_CS;

  return b0;
}

uint16_t Medidor2_ADE7753_Read16(unsigned char reg)
{
  ADE7753_CS2 = HABILITA_CS;
  unsigned char b1, b0;
  MSdelay(10);
  b1 = spi_transfer(reg);
  MSdelay(10);
  b0 = spi_transfer(reg);
  ADE7753_CS2 = DESABILITA_CS;

  return (uint16_t)b1<<8 | (uint16_t)b0;
}

uint32_t Medidor2_ADE7753_Read24(unsigned char reg)
{
  ADE7753_CS2 = HABILITA_CS;
  unsigned char b2, b1, b0;
  MSdelay(10);
  b2 = spi_transfer(reg);
  MSdelay(10);
  b1 = spi_transfer(reg);
  MSdelay(10);
  b0 = spi_transfer(reg);
  ADE7753_CS2 = DESABILITA_CS;

  return (uint32_t)b2<<16 | (uint32_t)b1<<8 | (uint32_t)b0;
}

//MEDIDOR 3
unsigned char Medidor3_ADE7753_Read8(unsigned char reg)
{
  ADE7753_CS3 = HABILITA_CS;
  unsigned char b0;
  MSdelay(10);
  b0 = WriteSPI(reg);
  MSdelay(10);
  ADE7753_CS3 = DESABILITA_CS;

  return b0;
}

uint16_t Medidor3_ADE7753_Read16(unsigned char reg)
{
  ADE7753_CS3 = HABILITA_CS;
  unsigned char b1, b0;
  MSdelay(10);
  b1 = spi_transfer(reg);
  MSdelay(10);
  b0 = spi_transfer(reg);
  ADE7753_CS3 = DESABILITA_CS;
}

```

```

    return (uint16_t)b1<<8 | (uint16_t)b0;
}

uint32_t Medidor3_ADE7753_Read24(unsigned char reg)
{
    ADE7753_CS3 = HABILITA_CS;
    unsigned char b2, b1, b0;
    MSdelay(10);
    b2 = spi_transfer(reg);
    MSdelay(10);
    b1 = spi_transfer(reg);
    MSdelay(10);
    b0 = spi_transfer(reg);
    ADE7753_CS3 = DESABILITA_CS;

    return (uint32_t)b2<<16 | (uint32_t)b1<<8 | (uint32_t)b0;
}

```

```

//=====
// ade7753.h

#ifndef ADE7753_H
#define ADE7753_H

#ifdef __cplusplus
extern "C" {
#endif

#ifdef __cplusplus
}
#endif

#endif /* ADE7753_H */

#include "main.h"

//Dados do datasheet do ADE7753
//Nome do Registrador      Endereço Bits
#define _WAVEFORM           0x01 //24
#define _AENERGY           0x02 //24
#define _RAENERGY          0x03 //24
#define _LAENERGY          0x04 //24
#define _VAENERGY          0x05 //24
#define _RVAENERGY         0x06 //24
#define _LVAENRGY          0x07 //24
#define _LVARENERGY        0x08 //16
#define _MODE               0x09 //16
#define _IRQEN              0x0A //16
#define _STATUS             0x0B //16
#define _RSTSTATUS         0x0C //16
#define _CH1OS              0x0D //8
#define _CH2OS              0x0E //8
#define _GAIN               0x0F //8
#define _PHCAL              0x10 //6
#define _APOS               0x11 //16
#define _WGAIN              0x12 //12
#define _WDIV               0x13 //8
#define _CFNUM              0x14 //12
#define _CFDEN              0x15 //12

```



```

#define _IRMS          0x16  //24
#define _VRMS          0x17  //24
#define _IRMSOS        0x18  //12
#define _VRMSOS        0x19  //12
#define _VAGAIN        0x1A  //12
#define _VADIV         0x1B  //8
#define _LINECYC       0x1C  //16
#define _ZXTOUT        0x1D  //12
#define _SAGCYC        0x1E  //8
#define _SAGLVL        0x1F  //8
#define _IPKLVL        0x20  //8
#define _VPKLVL        0x21  //8
#define _IPEAK         0x22  //24
#define _RSTIPEAK      0x23  //24
#define _VPEAK         0x24  //24
#define _RSTVPEAK      0x25  //24
#define _TEMP          0x26  //8
#define _PERIOD        0x27  //16
#define _TMODE         0x3D  //8
#define _CHKSUM        0x3E  //6
#define _DIEREV        0x3F  //8

#define HABILITA_CS    0      // Coloca o chip select em low para habilitar
#define DESABILITA_CS 1      // Coloca o chip select em high para desabilitar

unsigned char spi_transfer(unsigned char data);

//MEDIDOR 1
unsigned char Medidor1_ADE7753_Read8(unsigned char reg); //Função de leitura de resitradores de
8bits
uint16_t Medidor1_ADE7753_Read16(unsigned char reg);      //Função de leitura de resitradores de
16bits
uint32_t Medidor1_ADE7753_Read24(unsigned char reg);      //Função de leitura de resitradores de
24bits

//MEDIDOR 2
unsigned char Medidor2_ADE7753_Read8(unsigned char reg); //Função de leitura de resitradores de
8bits
uint16_t Medidor2_ADE7753_Read16(unsigned char reg);      //Função de leitura de resitradores de
16bits
uint32_t Medidor2_ADE7753_Read24(unsigned char reg);      //Função de leitura de resitradores de
24bits

//MEDIDOR 3
unsigned char Medidor3_ADE7753_Read8(unsigned char reg); //Função de leitura de resitradores de
8bits
uint16_t Medidor3_ADE7753_Read16(unsigned char reg);      //Função de leitura de resitradores de
16bits
uint32_t Medidor3_ADE7753_Read24(unsigned char reg);      //Função de leitura de resitradores de
24bits

```