



**UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA**  
**DCET – DEPARTAMENTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**ÉdaSuaConta: UM SISTEMA DE MONITORAMENTO DE CONSUMO DE  
ENERGIA ELÉTRICA E AUTOMAÇÃO RESIDENCIAL COM HARDWARE E  
SOFTWARE LIVRE.**

**MATHEUS THIAGO MARQUES BARBOSA**

**JUNHO - 2017**

**VITORIA DA CONQUISTA**

MATHEUS THIAGO MARQUES BARBOSA

**ÉdaSuaConta: UM SISTEMA DE MONITORAMENTO DE CONSUMO DE  
ENERGIA ELÉTRICA E AUTOMAÇÃO RESIDENCIAL COM HARDWARE E  
SOFTWARE LIVRE.**

Monografia apresentada ao Curso de Graduação em  
Ciência da Computação da Universidade Estadual do  
Sudoeste da Bahia, como requisito parcial para  
obtenção do Grau de Bacharel em Ciência da  
Computação. Orientadores: Profa. Dra. Alexandra  
Oliveira Andrade; Prof. Msc. Marcos Gomes Prado.

**JUNHO - 2017**

**VITÓRIA DA CONQUISTA**

MATHEUS THIAGO MARQUES BARBOSA

**ÉdaSuaConta: UM SISTEMA DE MONITORAMENTO DE CONSUMO DE  
ENERGIA ELÉTRICA E AUTOMAÇÃO RESIDENCIAL COM HARDWARE E  
SOFTWARE LIVRE.**

Aprovada em \_\_\_/\_\_\_/\_\_\_\_\_

BANCA EXAMINADORA

---

Profa. Dra. Alexsandra Oliveira Andrade

Universidade Estadual do Sudoeste da Bahia - UESB

Orientadora

---

Prof. Msc. Marcos Gomes Prado

Universidade Estadual do Sudoeste da Bahia - UESB

Co-Orientador

---

Profa. Dra. Alzira Ferreira Silva

Universidade Estadual do Sudoeste da Bahia - UESB

Convidada

## **DEDICATÓRIA**

Dedico este trabalho em homenagem (*in memórian*) ao meu pai Floriano Rodrigues Barbosa e aos meus avós Maria Anita Carvalho Marques e José Arteiro Marques, os quais partiram durante o período do curso.

## **AGRADECIMENTOS**

Toda caminhada almeja um fim. Este trabalho de conclusão de curso é o resultado de mais uma caminhada. Agradecer àqueles que colaboraram nesse processo é o mínimo.

Primeiramente agradeço a Deus por ter me dado força e conhecimento em todos esses momentos difíceis longe de casa;

À minha Família, especialmente a minha mãe e irmãs, que fizeram com que fosse possível a realização dessa etapa na minha vida, me incentivando e dando amor incondicional nas horas que mais precisei;

À minha namorada Izabella pelo carinho e paciência nos momentos mais difíceis que passei na minha vida;

Aos meus irmãos de Curso do Laboratório de Inteligência em Dispositivos de Arquitetura Livre e Veículos Autônomos, especialmente a Iago Gomes, Matheus Lima, Rodrigo Lima, Wanderson Almeida e Yan Kaic que me ajudaram a construir o protótipo me dando ideias e até colocando a mão na massa;

Aos Professores que se transformaram em eternos parâmetros de conhecimento: Alzira Ferreira da Silva, Roque Mendes Prado Trindade, Stênio Longo Araújo, Marlos André Marques S. de Oliveira, E em especial aos meus orientadores Alexandra Oliveira Andrade e Marcos Gomes Prado pelo incentivo e disposição à construção desse trabalho;

Meus agradecimentos emocionados e verdadeiros para todos, que de alguma forma contribuíram para a construção desse trabalho.

## LISTA DE SIGLAS

A	Ampère
AC-DC	Alternating Current/Direct Current
ANEEL	Agência Nacional de Energia Elétrica
ARM	Advanced RISC Machine
ARPANET	Advanced Research Projects Agency Network
CERN	Organisation Européenne pour la Recherche Nucléaire
CPU	Central Processing Unit
GND	graduated neutral density filter
GPIO	General Purpose Input/Output
gW	gigawatt
gWh	gigawatt-hora
HTTP	HyperText Transfer Protocol
Hz	Hertz
IoT	Internet of Things
IDE	Integrated Development Environment
I2C	Inter-Integrated Circuit
kW	Quilowatt
kWh	Quilowatt-hora
KB	Quilo Byte
mA	mili-ampere
MB	Mega Byte
MHz	Mega Hertz

MIT	Instituto de Tecnologia de Massachusetts
MQTT	Messaging Queue Telemetry Transport
M2M	Machine-to-Machine
PCB	Printed Circuit Board
PWM	Pulse With Modulation
QoS	quality of service
USART	Universal Synchronous Asynchronous Receiver Transmitter
V	Volt
RAM	Random Access Memory
SGBD	Sistema Gerenciador de Banco de Dados
SPI	Serial Peripheral Interface
W	Watt
WPA	Wi-Fi Protected Access
WEP	Wired Equivalent Privacy
Wh	Watt-hora

## LISTA DE FIGURAS

FIGURA 1: AUMENTO NAS BANDEIRAS TARIFÁRIAS DE ENERGIA .....	17
FIGURA 2: REPRESENTAÇÃO DO SISTEMA MOBII, ONDE TEM-SE O CARRO COM A CÂMERA DE RECONHECIMENTO FACIAL E O PAINEL DE CONTROLE DO SISTEMA. ....	23
FIGURA 3: REPRESENTAÇÃO DO SISTEMA PROTEUS, COMPRIMIDO SENSOR, SISTEMA RECEPTOR E SMARTPHONE COM O SISTEMA DE ASSISTÊNCIA MÉDICA. ....	24
FIGURA 4: REPRESENTAÇÃO DO USO DO SHIMA COM O GPS. ....	24
FIGURA 5: O TOQUE NO TECIDO DA ROUPA É TRANSMITIDO SEM FIO A DISPOSITIVOS PARA CONTROLAR FUNÇÕES DESSES DISPOSITIVOS. ....	25
FIGURA 6: REFRIGERADOR FAMILY HUB E O APLICATIVO DE CELULAR. ....	25
FIGURA 7: CAFETEIRA SMARTER. ....	26
FIGURA 8: SMART TV SAMSUNG. ....	26
FIGURA 9: SISTEMA DA SAMSUNG SMART HOME. ....	27
FIGURA 10: REPRESENTAÇÃO DO SISTEMA DA OPENENERGYMONITOR. ....	27
FIGURA 11: A ARQUITETURA CENTRALIZADA DOS SISTEMAS INTEGRADOS. ....	29
FIGURA 12: A ARQUITETURA DESCENTRALIZADA DO SISTEMA COMPLEXO. ....	29
FIGURA 13: DIAGRAMA BÁSICO DE UM SISTEMA EMBARCADO DOTADO DE UM MICROCONTROLADOR MONITORANDO O AMBIENTE. ....	30
FIGURA 14: REPRESENTAÇÃO DOS ARDUINOS UNO, 101, PRO, MICRO E MINI. ....	32
FIGURA 15: REPRESENTAÇÃO DOS ARDUINOS MEGA E ZERO. ....	32
FIGURA 16: REPRESENTAÇÃO DOS ARDUINOS VESTÍVEIS: GEMMA, LILYPAD SIMPLES, LILYPAD MAIN BOARD, LILYPAD USB, E LILYPAD SIMPLE SNAP. ....	33
FIGURA 17: VARIANTES DO ESP8266 .....	34
FIGURA 18: ESP-01 E SEU CIRCUITO. ....	34
FIGURA 19: ESP-07 E SEU CIRCUITO. ....	35
FIGURA 20: ESP-12E E SEU CIRCUITO. ....	35
FIGURA 21: PINAGEM NODE MCU V3 .....	36
FIGURA 22: NODEMCU ESP-12E. ....	36
FIGURA 23: A PLACA RASPBERRY PI B+. ....	38
FIGURA 24: ARQUITETURA DE COMUNICAÇÃO CLIENTE-SERVIDOR. ....	41
FIGURA 25: EXEMPLO DE USO DA ARQUITETURA MQTT. ....	42
FIGURA 26: EXEMPLO DE USO DA ARQUITETURA MQTT. ....	43
FIGURA 27: TAMANHO DE TELAS SUPORTADAS NO SISTEMA DE GRIDS DO BOOTSTRAP. ....	45
FIGURA 28: DIAGRAMA DE CASO DE USO DO SISTEMA. ....	49
FIGURA 29: MODELO CONCEITUAL DO BANCO DE DADOS. ....	52
FIGURA 30: FORMA CORRETA DE ENVOLVER O SENSOR NO FIO. ....	58
FIGURA 31: CIRCUITO UTILIZADO PARA SER POSSÍVEL A LEITURA DOS DADOS DO SENSOR SCT 013-000. ....	60
FIGURA 32: CIRCUITO RESULTANTE PARA O SENSOR DE CORRENTE, QUE PRODUZ UM SINAL PROPORCIONAL À CORRENTE DA REDE, QUE CONVERTE O SINAL EM UM FORMATO QUE O ARDUINO PODE USAR. ....	60
FIGURA 33: PLACA DE FENOLITE E PERCLORETO DE FERRO EM UMA VASILHA DE PLÁSTICO. ....	62
FIGURA 34: A: SCHEMATIC DO CIRCUITO DO LEITOR DE CORRENTE. B: DESIGN DA PLACA A SER IMPRESSA. ....	63
FIGURA 35: A: FOLHA COM O DESIGN DA PLACA; B: DESIGN IMPRESSO E COLADO À PLACA DE FENOLITE; C: PLACA DE FENOLITE APÓS A CORROSÃO À ESQUERDA ANTES DE SER LAVADA E DEPOIS DE SER LAVADA À DIREITA. ....	64



FIGURA 36: A: PLACA PERFORADA; B: PLACA COM OS COMPONENTES SOLDADOS; C: PLACA FINALIZADA.....	64
FIGURA 37: A: PROTÓTIPO DO SISTEMA COLOCADO NA RESIDÊNCIA. B: PROTÓTIPO DO SISTEMA COLOCADO NO LABORATÓRIO. ....	65
FIGURA 38: ADAPTAÇÃO DO FILTRO DE LINHA AOS RELÉS .....	66
FIGURA 39: NODEMCU CONECTADO AO MÓDULO DE RELÉS .....	66
FIGURA 40: CAIXA DE DISJUNTORES DA RESIDÊNCIA. ....	67
FIGURA 41: PROTÓTIPO EM ATIVIDADE NA RESIDÊNCIA. ....	67
FIGURA 42: PROTÓTIPO INSTALADO NA RESIDÊNCIA.....	68
FIGURA 43: ARQUITETURA DO SISTEMA.....	71
FIGURA 44: TEMPO DE EXECUÇÃO DO COMANDO INSERT E SELECT EM UM BANCO MySQL EM UM COMPUTADOR E EM UM RASPBERRY PI.....	73
FIGURA 45: EXEMPLO DE COMUNICAÇÃO UTILIZADA PARA LIGAR UM DISPOSITIVO. ....	75
FIGURA 46: EXEMPLO DE COMUNICAÇÃO UTILIZADA PARA DESLIGAR UM DISPOSITIVO. ....	75
FIGURA 47: VERIFICAÇÃO DO STATUS DOS RELÉS .....	76
FIGURA 48: VERIFICAÇÃO DO STATUS A LUZ NEGRA LIGADA. ....	77
FIGURA 49: TESTE DE ACIONAMENTO DE DUAS LÂMPADAS .....	77
FIGURA 50: NOTA FISCAL DO CONSUMO DO MÊS DE MARÇO .....	78
FIGURA 51: GRÁFICO DE CONSUMO DO MÊS DE MARÇO. ....	79
FIGURA 52: NOTA FISCAL DO CONSUMO DO MÊS DE ABRIL.....	80
FIGURA 53: GRÁFICO DE CONSUMO DO MÊS DE ABRIL.....	80
FIGURA 54: SENSOR E AMPERÍMETRO ENVOLTOS AO MESMO FIO.....	81
FIGURA 55: GRÁFICO RESULTANTE DO TESTE DE VELOCIDADE.....	82
FIGURA 56: AMPERÍMETRO E CELULAR LADO A LADO PARA VERIFICAÇÃO DOS DADOS DE LEITURA DO CONSUMO.....	83
FIGURA 57: AMPERÍMETRO E CELULAR LADO A LADO PARA VERIFICAÇÃO DOS DADOS DE LEITURA DO CONSUMO DE UM CHUVEIRO ELÉTRICO. ....	84
FIGURA 58: VISUALIZAÇÃO DO SISTEMA EM CELULARES IOS E ANDROID.....	86
FIGURA 59: VISUALIZAÇÃO DO SISTEMA EM UM TABLET. ....	87
FIGURA 60: VISUALIZAÇÃO DO SISTEMA EM UM DESKTOP. ....	88

## LISTA DE TABELAS

TABELA 1: DIAGRAMA DE CASO DE USO DO SISTEMA .....	50
TABELA 2: QUADRO COM O COMANDO DE CRIAÇÃO DO BANCO DE DADOS `ENERGY` .....	53
TABELA 3: QUADRO COM O COMANDO DE CRIAÇÃO DA TABELA `MEDIDAS` .....	53
TABELA 4: QUADRO COM O COMANDO DE CRIAÇÃO DA TABELA `MEDIDASHORA` .....	54
TABELA 5: QUADRO COM O COMANDO DE CRIAÇÃO DA TABELA `USUARIOS` .....	54
TABELA 6: PREÇO DOS HARDWARES UTILIZADOS.....	57

## RESUMO

As situações econômicas e energéticas do Brasil passam por um momento difícil. Em época de crise, a capacidade de poupar e diminuir o desperdício monetário se torna essencial. Devido ao aumento das tarifas de eletricidade, a consciência da população com a gestão da energia residencial se faz necessária, pois os benefícios de sistemas que meçam e controlem os gastos podem mostrar-se eficazes quanto a diminuição do consumo. Quanto ao conforto e comodidade providos pela automação residencial, que para uma boa parte da população não passava de um sonho, a realidade pode ser modificada através de soluções envolvendo hardware e software de baixo custo.

Este trabalho apresenta o ÉdaSuaConta, que é um sistema web voltado para o monitoramento do consumo energético e automação de uma residência. O monitoramento do consumo, utiliza a plataforma Arduino para realizar a leitura da corrente por meio de sensores e acionamento de atuadores para o controle de dispositivos na residência, fica por conta do módulo NodeMCU ESP8266. Os valores do consumo são enviados para um servidor web em um Raspberry Pi, onde esses dados são armazenados em um banco de dados e apresentados graficamente. Para o controle de dispositivos o sistema contou com o protocolo de comunicação da Internet das Coisas, o MQTT. Esse sistema é desenvolvido utilizando plataformas open-source de baixo custo, viabilizando o acesso a automação residencial às classes menos favorecidas.

**Palavras-chave:** Raspberry Pi, Arduino, consumo de energia.

## **ABSTRACT**

The economic and energetic situation in Brazil is going through a difficult time. In times of crisis, the ability to save and reduce wastage money becomes essential. Once with the increase in electricity rates, population awareness with the management of residential energy is necessary, because the benefits of systems that measure and control the costs may prove effective as the decline in consumption. About the comfort and convenience provided by home automation, which for much of the population was only a dream, the reality can be modified through solutions involving hardware and low-cost software.

This paper presents *ÉdaSuaConta*, a web oriented system for monitoring energy consumption and automation of a residence. The consumption monitoring, uses an Arduino platform to realize the electric current reading by means of sensors; the control of actuators uses the module NodeMCU ESP8266 for the control of devices in the residence. Consumption values be sent to a web server on a Raspberry Pi, where the data be stored in a database and shown graphically. This system is developed using free platforms with low cost, enabling access to home automation to underprivileged classes.

**Keywords:** Raspberry Pi, Arduino, eletricity home consumption.

## SUMÁRIO

1. INTRODUÇÃO .....	15
1.1. Contextualização .....	15
1.2 Objetivos .....	19
1.2.1 Objetivo geral .....	19
1.2.2 Objetivos específicos .....	19
1.3 Justificativa .....	20
1.4 Estrutura do Trabalho .....	20
2. REFERENCIAL TEÓRICO .....	20
2.1 IoT – Internet das coisas .....	21
2.2 Automação residencial .....	28
2.3 Microcontroladores, Sensores e Atuadores .....	30
2.3.1 Plataforma Arduino .....	31
2.3.2 Plataforma ESP .....	33
2.4 NodeMCU ESP-12E .....	35
2.5 Plataforma Raspberry Pi .....	37
2.6 Protocolos de comunicação .....	38
2.6.1 Protocolo de Comunicação Serial .....	39
2.6.2 Protocolo HTTP .....	40
2.6.3 Protocolo MQTT .....	41
2.7 Desenvolvimento Web .....	44
3. METODOLOGIA .....	46
3.1 Planejamento .....	46
3.2 Modelagem do Sistema .....	47
3.2.1 Apresentação do sistema .....	47
3.2.2 Requisitos funcionais .....	47
3.2.3 Requisitos não funcionais .....	48
3.2.4 Atores do Sistema .....	50
3.2.5 Casos de Uso .....	50
3.2.6 Modelo do Banco de Dados .....	51
3.3 Metodologia de Testes .....	55
4. FERRAMENTAS UTILIZADAS .....	55
4.1 Hardware .....	55
4.1.1 Preço do Hardware Utilizado .....	56
4.1.2 Prototipação do circuito para a leitura do sensor de corrente .....	57
4.1.3 Construção da PCB Do Circuito Para a Leitura da Corrente .....	61

4.1.4 Montagem do protótipo .....	64
4.1.4 Instalação do protótipo do Leitor de Consumo.....	67
4.2 Software .....	68
4.2.1 Ferramentas Utilizadas no Front-End.....	68
4.2.2 Ferramentas Utilizadas no Back-End .....	69
4.3 Arquitetura do Sistema.....	71
5. CENÁRIOS, TESTES, REPAROS E RESULTADOS.....	72
5.1 Falta de alimentação e reinício automático do sistema.....	72
5.2 Demora na pesquisa e inserção do banco de dados – Criação de rotina de inserção a cada hora.....	72
5.3 Erro de leitura do leitor de consumo utilizando protoboard – Construção de uma PCB.....	73
5.4 Acúmulo de funções ao Arduino - NodeMCU e o protocolo MQTT .....	74
5.4.1 Teste de acionamento com duas lâmpadas.....	76
5.5 Teste de Leitura .....	78
5.6 Testes utilizando monitor de consumo instantâneo .....	81
5.6.1 Teste de Velocidade.....	82
5.6.2 Teste com uma sanduicheira .....	82
5.6.3 Teste com um chuveiro elétrico .....	83
5.6.4 Teste sem equipamentos conectados.....	84
5.7 Diferentes tamanhos de telas para acessar o sistema.....	85
5.7.1 Interfaces do sistema .....	85
6. CONCLUSÃO.....	89
7. TRABALHOS FUTUROS .....	91
8. REFERÊNCIAS.....	92
ANEXO 1.....	99

# 1. INTRODUÇÃO

Este capítulo apresenta as considerações que darão contexto ao trabalho, além dos seus objetivos, justificativa e estrutura do trabalho.

## 1.1. Contextualização

A energia é um bem básico e um fator determinante no desenvolvimento da sociedade (KONDO, 2016), ela pode ser aplicada em todas as áreas, desde a produção de alimentos, passando por comércio, indústria, transporte, inclusive moradia. Para que a economia de um país seja próspera, ele precisa de uma matriz energética que dê suporte à sua demanda por energia.

Existem várias fontes de energia no planeta, e elas basicamente se dividem em dois tipos; fontes renováveis e fontes não renováveis. As fontes de energia renováveis como o próprio nome já diz, podem ser renováveis, havendo a possibilidade de ser reaproveitadas sem a possibilidade de esgotamento. Exemplos desse tipo de fonte são: energia hídrica, solar, eólica, entre outras. Já as fontes de energias não renováveis têm recursos teoricamente limitados, e dependem dos recursos existentes no nosso planeta, como por exemplo, os combustíveis fósseis e combustíveis nucleares (PORTAL DA ENERGIA, 2016).

De acordo com dados da ANEEL, entre 4.547 de empreendimentos de energia em operação, 66,66% são de fontes hídricas, 17,53% são de combustíveis fósseis, 8,74% são fontes de biomassa, e os outros 7,07% restantes são de usinas eólicas, nucleares e solares.

O Brasil tem cerca de 12% da água doce superficial do planeta, e condições adequadas para a produção de energia hídrica. Com esse aparato, o país é um grande investidor e produtor a partir dessa fonte de energia, estando entre os cinco maiores do mundo no potencial técnico de aproveitamento da energia hidráulica; O potencial hidrelétrico é estimado em cerca de 260 gW, dos quais 40,5% estão localizados na Bacia Hidrográfica do Amazonas, 23% na Bacia do Paraná, 10,6% na bacia do Tocantins, e 10% na bacia do São Francisco. Contudo, apenas 63% do potencial foi

inventariado. A Região Norte, em especial, tem um grande potencial ainda por explorar (PORTAL BRASIL, 2016).

Por ter um recurso abundante, o Brasil se beneficia de suas hidrelétricas para a geração de energia. Entretanto, devido à necessidade de refrear o curso dos rios durante a construção das usinas, ocorre o alagamento de grandes áreas, causando desapropriação de moradores e impactos ambientais como perda da vegetação, da fauna e interferência direta na migração dos peixes (MEIO AMBIENTE TÉCNICO, 2016).

Além disso, em épocas de seca, a capacidade de geração de energia das usinas se reduz, fazendo com que seja necessário o acionamento de termelétricas para poupar água nos reservatórios. Por ser uma fonte de energia mais cara, o custo da geração é repassado ao consumidor pela forma de bandeiras tarifárias; uma relação entre o valor pago pelo consumidor e o custo atualizado pago pelas geradoras de energia. Essas bandeiras tarifárias são classificadas a partir das cores verde, amarelo e vermelho. A cor verde indica que o poder de geração está alto, não havendo aumento na conta de energia. A bandeira amarela indica um sinal de alerta, pelo motivo do custo de geração de energia estar aumentando. Já a bandeira vermelha mostra que o custo de geração está mais alto.

Em 2015, por conta de uma seca em algumas regiões do Brasil, a Agência Nacional de Energia Elétrica (ANEEL) aprovou uma proposta de aumento nas taxas das contas de luz. Nessa proposta os aumentos passaram a ser de 66% e 83% referentes às bandeiras tarifárias amarela e vermelha respectivamente, como pode ser visto na Figura 1 (EBC PORTAL, 2016).



Figura 1: Aumento nas bandeiras tarifárias de energia



Fonte: (EBC PORTAL, 2016).

O consumidor pode identificar qual a bandeira tarifária do mês, e utilizar a energia elétrica de forma mais consciente, sem desperdício. Diversas aplicações práticas de gestão e otimização do uso de energia podem ser aplicadas em todas as áreas, principalmente na área residencial. As contribuições dessas implementações são importantes para a segurança energética, modicidade tarifária, competitividade da economia e redução de impactos ambientais, entre eles as emissões de gases de efeito estufa (EPE, 2014).

A partir das tecnologias usadas na automação residencial, existem diversos projetos, em especial, um projeto de redes inteligentes, chamadas de *Smart Grids*, responsáveis por distribuir a energia de forma segura e com tolerância às possíveis falhas, e promover uma medição justa do consumo. Nesse conceito, o fluxo de energia elétrica e de informações se dá de forma bidirecional. Assim, a energia tradicionalmente gerada, transmitida e distribuída de forma radial a partir de instalações das concessionárias, poderá também, ser gerada e integrada às redes elétricas a partir de unidades consumidoras. Logo o consumidor passa a ser produtor e consumidor, pois produz e fornece energia à rede.

O objetivo principal desse projeto é dar ao usuário a capacidade de gerenciar e tomar decisões relacionadas ao seu consumo de energia, que hoje é feito pelas empresas de geração de energia. Este conceito de *Smart Grid* tenta mudar um sistema que permanece praticamente inalterado desde o século 19, com estações geradoras e um sistema de entrega de energia eletromecânico, operado por centrais de controle (SIDDQUI, 2008).

As possibilidades de consumo mais econômico e consciente de energia elétrica, contribuem também para a redução das emissões de gases de efeito estufa do sistema elétrico brasileiro (CEMIG, 2016). Dessa forma, é extremamente importante a utilização de aplicações de automação residencial que visam à diminuição de desperdícios energéticos.

A automação residencial ou Domótica, (da união do latim “*Domus*”, que significa casa e “robótica”- controle automatizado de algo) veio para permitir aos usuários um controle automatizado das residências, agregando mais facilidade e qualidade de vida às pessoas. O custo da implantação de sistemas que façam uma casa inteligente geralmente é alto, tornando-o inacessível a uma boa parcela da população. No entanto, houve o surgimento de novas tecnologias que permitem a construção de sistemas de automação similares aos comercializados em grande escala com um custo bastante menor. O conceito de IoT aliada a automação residencial, propicia aos usuários controle sobre suas residências mesmo à distância, a fim proporcionar uma nova experiência de vida aos moradores dessas “casas inteligentes”.

A IoT (*Internet of Things*, ou em tradução literal, Internet das Coisas), é um conceito que descreve objetos físicos que estão conectados à internet, sendo assim capaz de se interconectar a outros dispositivos, físicos ou virtuais. Digamos que a IoT é uma extensão da internet que utilizamos hoje, oferecendo o desenvolvimento de aplicações e serviços via web, computação na nuvem ou comunicações em rede, buscando uma maior experiência de uso. Em (DENNIS, 2013) e (MONK, 2014) são apresentados diversos projetos de automação residencial utilizando o microcontrolador Arduino e o microcomputador Raspberry Pi.

O Arduino foi criado em 2005 por um grupo de 5 pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis (THOMSEN, 2016a). Ele é um kit de desenvolvimento de prototipagem *open-source* que se baseia em software e hardware fáceis de usar (ARDUINO, 2016). O intuito do projeto Arduino é ser uma tecnologia de baixo custo, que tenha um desenvolvimento de projetos e aprendizagem facilitada. Com essa placa é possível o envio e recepção de informações de praticamente qualquer outro sistema eletrônico. Desta forma, é possível construir, por exemplo, um sistema de captação de dados de sensores, como temperatura e iluminação, processar e enviar esses dados para um servidor web em um Raspberry

Pi para automação residencial, como no trabalho de (VUJOVIC; MARKSIMOVIC, 2014).

O Raspberry Pi é um dos menores computadores do mundo. Ele tem dimensões aproximadas de um cartão de crédito, e pode ser utilizado nos mais diferentes projetos, como em centrais multimídia, Tablets (CASTOR, 2016), videogames portáteis (PIJUICE, 2016), impressora econômica e sem tinta (PIPSTA, 2016) entre outros. Pode ser utilizado também para a programação de diversas linguagens, entre elas o Python, Java, C/C++. Além disso, por oferecer portas GPIO é possível que seja usado como controlador de um robô, sistemas embarcados ou automação (NETTO, 2013).

A partir desse cenário, esse projeto tem como proposta o desenvolvimento de um sistema de monitoramento do consumo residencial, sendo o Arduino responsável por receber os dados do sensor de corrente e repassá-lo via comunicação serial para Raspberry Pi, que por sua vez lê esses valores, grava em um banco de dados e apresenta graficamente ao usuário. Além de monitorar, o sistema será capaz de controlar Relés e verificar o seu status com a placa NodeMCU comunicando com o Raspberry Pi wireless via protocolo MQTT.

## **1.2 Objetivos**

### **1.2.1 Objetivo geral**

Desenvolver um sistema para o monitoramento remoto do consumo de eletricidade e controle de forma remota de dispositivos de uma residência, utilizando hardware e software livre.

### **1.2.2 Objetivos específicos**

- Criar um sistema de leitura de consumo energético e automação residencial utilizando o Arduino, NodeMCU, sensores e atuadores;
- Desenvolver uma interface web responsiva;
- Desenvolver um sistema web em um Raspberry Pi;

- Traduzir os níveis de consumo recebidos pelo Arduino e passados para o sistema web no Raspberry Pi para apresentá-los ao usuário.

### **1.3 Justificativa**

A situação econômica e energética do Brasil passa por um momento difícil. E em época de crise, a capacidade de poupar e diminuir o desperdício monetário se torna essencial, uma vez que com o aumento das tarifas de eletricidade, a consciência da população com a gestão da energia residencial se faz necessária, pois os benefícios de sistemas que medem e controlam os gastos podem mostrar-se eficazes quanto à diminuição do consumo. Quanto ao conforto e comodidade providos pela automação residencial, que para uma boa parte da população não passavam de um sonho, a realidade pode ser modificada através de soluções envolvendo hardware e software livre.

Dessa forma, este trabalho se justifica pela busca de aplicações de monitoramento dos níveis de consumo de energia elétrica, e de sistemas de automação residencial de baixo custo.

### **1.4 Estrutura do Trabalho**

A parte textual deste trabalho está organizada em capítulos, onde o capítulo 1 apresenta a introdução, contextualização, objetivos e justificativa. No capítulo 2 está contido no texto o referencial teórico, onde são apresentados os assuntos necessários para o entendimento deste trabalho, em seguida o capítulo 3 é está especificada a metodologia utilizada seguido do Planejamento e Modelagem do Sistema. No capítulo 4, estão contidas no texto as ferramentas utilizadas para a construção do sistema. No capítulo 5 estão dispostos os Cenários, testes e resultados do sistema. Sendo finalizado pelos capítulos 6 e 7 que contem as conclusões e trabalhos futuros.

## **2. REFERENCIAL TEÓRICO**

Esse capítulo apresenta os conceitos teóricos que foram fundamentais para o desenvolvimento deste projeto.

## 2.1 IoT – Internet das coisas

A internet das coisas é uma tradução literal da expressão em inglês *Internet of Things* (IoT), termo criado por Kevin Ashton do Instituto de Tecnologia de Massachusetts (MIT). Apesar de ser um termo que não esteja popularizado, ele já está em muitos objetos do nosso dia-a-dia, como nos smartphones por exemplo. O seu conceito baseia-se na ideia de conectar itens de uso diário à internet, havendo comunicação entre os itens, e entre os itens e as pessoas (ASHTON, 2009).

Antes de compreender a importância da IoT, é necessário entender a diferença entre a Internet e a World Wide Web, termos que muitas vezes são utilizados como sinônimos.

Segundo TANENBAUM (2003): “A Internet não é de modo algum uma rede, mas sim um vasto conjunto de redes diferentes que utilizam certos protocolos e fornecem determinados serviços”. A sua infraestrutura é composta pela camada física da rede formada por switches, roteadores e outros equipamentos que interligam essas redes. Sua principal função é transportar informações de um ponto a outro de forma rápida, confiável e segura.

A World Wide Web, ou apenas Web, por outro lado, é uma camada de aplicação que opera no topo da Internet. Criada pelo físico da CERN, Tim Berners-Lee, tem como papel principal proporcionar uma interface que faz com que as informações que fluem através da Internet sejam mais facilmente utilizáveis, uma vez que a partir dela tornou-se possível a configuração de páginas de informações de um site utilizando textos e mídias (fotos, sons e vídeos). Um protocolo de aplicação amplamente utilizado é o HTTP (HyperText Transfer Protocol), que constitui a base para a Web. (TANENBAUM, 2003; EVANS, 2011). Este protocolo é responsável por gerenciar a transferência de dados em uma rede que conecta dois ou mais computadores. Quando se trata de comunicação de dois ou mais dispositivos, comunicação M2M (Machine-to-Machine), conectados a uma rede, há a necessidade de uso de protocolos de comunicação que sejam eficientes, considerando as características e limitações do ambiente. Um

protocolo muito utilizado em ambientes restritos, ambientes onde há perda de dados pela rede e capacidade computacional limitada, é o Messaging Queue Telemetry Transport (MQTT), chamado de protocolo de comunicação da internet das coisas, que é explicado na subseção 2.6.3.

A Internet está em um caminho contínuo de desenvolvimento e melhoria, mas quase não houve mudança em sua arquitetura. Mesmo com a Web, que facilitou o uso da internet, ela não se modificou, em sumo ela faz essencialmente a mesma coisa que ela foi projetada para fazer desde a ARPANET. Por esse motivo, a IoT é extremamente importante, pois, é a primeira evolução real da Internet. Ela faz com que sensores e atuadores sejam incorporados à Internet, permitindo uma interação não só entre os computadores “tradicionais”, mas entre objetos (sensores e atuadores) (EVANS, 2011).

Dessa forma, a Internet se torna uma grande janela de oportunidade para a criação de aplicações nas áreas de automação, monitoramento e comunicação entre máquinas. Os exemplos a seguir destacam algumas das maneiras que a Internet das coisas pode ser utilizada:

- Reconhecimento facial:
  - Mobii, desenvolvido pela Ford e Intel, é um projeto que faz com que ao entrar no carro, uma câmera reconhece o rosto do motorista, a fim de oferecer informações sobre o seu cotidiano, recomendar músicas de sua preferência e receber orientações do mapa. Caso o motorista não seja reconhecido, o sistema tira foto do condutor e envia ao smartphone do dono, que por sua vez, pode definir as permissões desse motorista como pode ser visto na Figura 2 (CALDWELL, 2014).

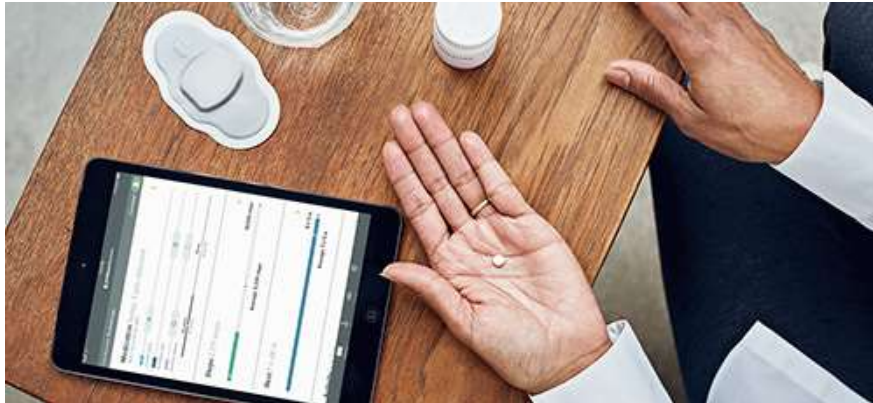
Figura 2: Representação do sistema Mobii, onde tem-se o carro com a câmera de reconhecimento facial e o painel de controle do sistema.



Fonte: (Ford, 2016)

- *Wearable Computer* (Computação vestível). É uma tecnologia que redefine a interação entre máquinas e humanos, fazendo com que os usuários tenham dispositivos diretamente ligados a eles. Alguns exemplos podem ser vistos a seguir:
  - Proteus é um sistema que é ativado ao tomar uma medicação com um sensor ingerível. Uma vez que esse sensor chega ao estômago, ele transmite um sinal para um receptor localizado no tronco do paciente. Esse receptor também contém sensores que monitoram parâmetros fisiológicos como a frequência cardíaca, respiração e movimento corporal. A partir daí um sinal é enviado para o smartphone do paciente e, em seguida, para a nuvem Proteus, onde com a permissão do paciente, os profissionais de saúde podem acessá-lo através do seu portal. O paciente poderá, então, ser aconselhado a ajustar as dosagens ou alterar o medicamento utilizado conforme Figura 3 (PROTEUS DIGITAL HEALTH, 2016).

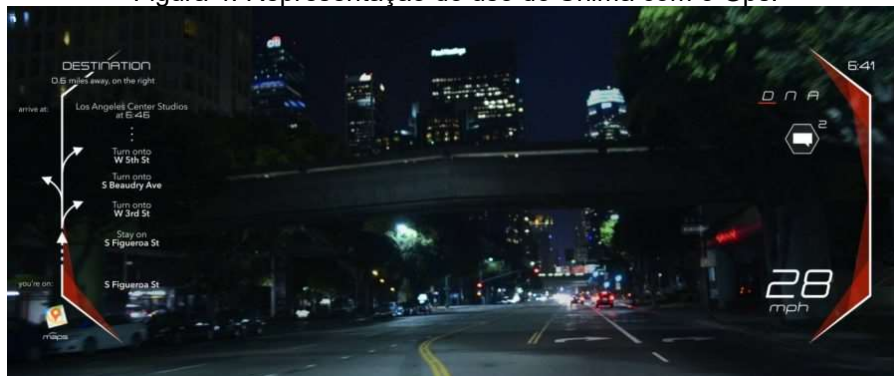
Figura 3: Representação do Sistema Proteus, comprimido sensor, sistema receptor e smartphone com o sistema de assistência médica.



Fonte: (PROTEUS DIGITAL HEALTH, 2016)

- Shima, desenvolvido pela Laforge Optical, é um óculos que possibilita a interação dos usuários com diversos conteúdos em realidade aumentada. Esses óculos apresentam as informações em torno do campo de visão do usuário, onde se é possível, a partir de toque nas suas hastes, controlar um reprodutor de mídia, por exemplo, verificar notificações de redes sociais, verificar o clima, e utilizar em conjunto com o GPS (como pode ser visto na Figura 4). O Shima é controlado por um aplicativo instalado em seu smartphone que trabalha com iOS, Android e Windows Phone (LAFORGE, 2016). Na Figura 4 podemos ver a representação do campo de visão com a utilização do Shima.

Figura 4: Representação do uso do Shima com o Gps.



Fonte: (LAFORGE, 2016)

- Outro exemplo é o projeto Jacquard, que permite que o usuário interaja com um celular, a partir de toques em um tecido. Com ele é possível aumentar o volume, passar, pausar a música, sem tirar o celular do bolso, apenas tocando na sua roupa, ilustrado na Figura 5 (PROJECT JACQUARD, 2016).



Figura 5: O toque no tecido da roupa é transmitido sem fio a dispositivos para controlar funções desses dispositivos.



Fonte: (PROJECT JACQUARD, 2016).

- Eletrodomésticos:

- *Family Hub* é um refrigerador que oferece ao usuário novas maneiras de gerenciar e verificar o que está faltando na geladeira, uma vez que o sistema tira foto todas as vezes que a porta da geladeira é fechada. É possível também deixar notas, lembretes ou fotos no display que podem ser visto na Figura 6 (FAMILY HUB REFRIGERATOR, 2016).

Figura 6: Refrigerador Family Hub e o aplicativo de celular.



Fonte: (FAMILY HUB REFRIGERATOR, 2016).

- Cafeteiras que possibilitam que o usuário seja capaz de controlar o que acontece com seu café remotamente, como por exemplo, ser acordado apenas quando o café estiver pronto, receber alertas quando a cafeteira precisar ser recarregada, ou fazer café de tempos em tempos, a partir de aplicativos de celulares como mostrado na Figura 7. (SMARTER, 2016);

Figura 7: Cafeteira Smarter.



Fonte: (SMARTER, 2016)

- Smart TVs que oferecem vários níveis de conectividade, onde pode ser acessado redes sociais, sites diversos, fazer vídeo chamadas, utilizar o serviço de stream de música e vídeo, entre outros, como ilustradas na Figura 8.

Figura 8: Smart Tv Samsung.

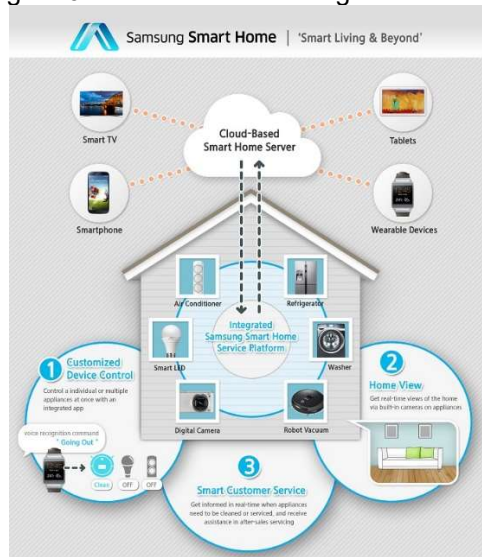


Fonte: (SAMSUNG, 2016a).

- *Smart Home* são casas que permitem que o usuário gerencie os eletrodomésticos inteligentes através do celular, como em um sistema feito pela Samsung por exemplo. Os usuários podem usar as configurações personalizadas em seus dispositivos móveis ou Smart TV para monitorar ou controlar eletrodomésticos, ligar o ar condicionado ou ativar a iluminação, por exemplo. A partir de um aplicativo esse sistema permite que os usuários controlem um ou vários dispositivos simultaneamente, não importando a

localização dos dispositivos e do usuário, desde que estejam conectados à internet conforme a Figura 9.

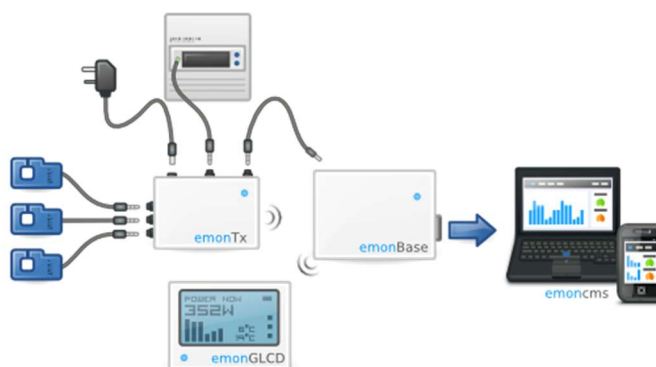
Figura 9: Sistema da Samsung Smart Home.



Fonte: (SAMSUNG, 2016b)

- *OpenEnergyMonitor* trata-se de um projeto para o desenvolvimento de ferramentas de monitoramento de energia *open-source* que permitem ao usuário verificar em forma de relatórios, os gastos energéticos de sua residência, ajudando assim o relacionamento com a sua própria utilização de energia como pode ser observado na Figura 10. (OPENENERGYMONITOR, 2016a).

Figura 10: Representação do sistema da OpenEnergyMonitor.



Fonte: (OPENENERGYMONITOR, 2016a)

Como visto, a internet das coisas pode ser empregada em diversas áreas, mas nada melhor do que trazer comodidade aliada com a tecnologia para dentro de nossas

casas. Assim, começaremos a falar de um conceito importante, que começou antes da IoT, e hoje é muito utilizado aliado a esse conceito, que é a Automação Residencial.

## 2.2 Automação residencial

A automação residencial, também chamada de Domótica, é um conjunto de serviços fornecidos por ferramentas tecnológicas que buscam satisfazer as necessidades básicas de segurança, comunicação, gestão energética e conforto em uma residência (MURATORI; DAL BÓ, 2001). Ela também pode ser definida como um sistema que proporciona alguma funcionalidade ou mensagem ao ambiente residencial sem a necessidade de intervenção humana direta. (RILEY, 2012).

De acordo com Sena (2016) e Teza (2002), atualmente a automação residencial é dividida em três níveis quanto à sua intensidade de interação com o usuário, que são:

- Sistemas autônomos ou *stand-alone*: Sistemas que conseguem operar de forma independente sem a intervenção de um controlador centralizado, pois são projetados somente para agir de acordo com um ajuste predefinido.
- Sistemas integrados: Sistemas que tem uma central inteligente programada através de um microprocessador, onde são inseridas funções de liga/desliga, de comutações e regulações. Esse tipo de sistema possui uma arquitetura centralizada, ou seja, um controlador envia e recebe as informações dos sensores e atuadores conectados a ele segundo a Figura 11, mas tem uma importante limitação, pelo fato do sistema funcionar apenas da forma com que o seu fabricante pretende.

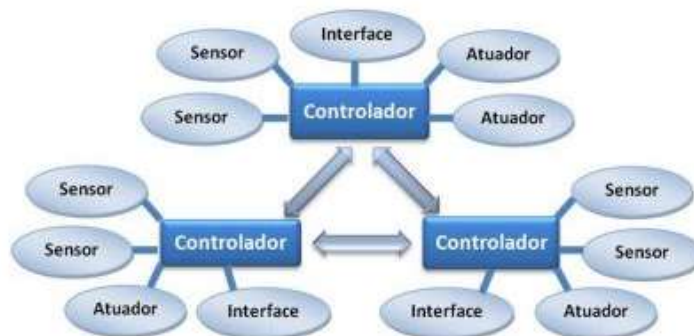
Figura 11: A arquitetura centralizada dos sistemas integrados.



Fonte: <https://repositorio.ufrn.br/jspui/bitstream/123456789/15219/1/DaniseSS>

- **Sistemas complexos:** Também chamado de casa inteligente, é um sistema onde o projetista e o usuário delineiam instruções específicas que atendam aos seus requisitos. Assim, o sistema torna-se gerenciador ao invés de ser apenas um controlador remoto, e há uma integração total dos sistemas domésticos. Esse tipo de sistema possui arquitetura descentralizada, ou seja, existem vários controladores interconectados, havendo comunicação entre unidade controladora e os sensores ou atuadores como a Figura 12.

Figura 12: A arquitetura descentralizada do Sistema complexo.



Fonte: <https://repositorio.ufrn.br/jspui/bitstream/123456789/15219/1/DaniseSS.pdf>

Esse modelo de interação é utilizado na IoT, mas, além dos controladores serem descentralizados eles ainda são conectados à internet. Sendo assim, as instruções de delineamento do funcionamento dos controladores podem ser definidas pelo usuário remotamente.

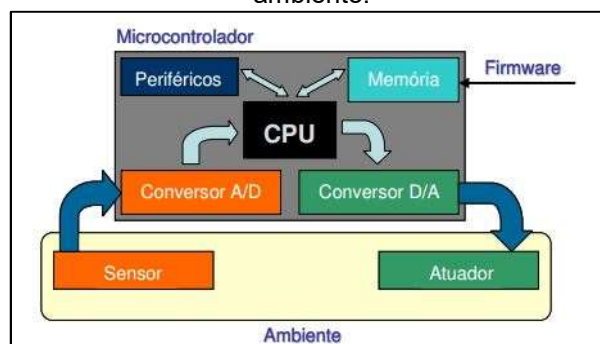
A transformação de casas em lugares capazes de serem seguros e terem economia nos seus custos é uma vantagem da automação residencial. O que antes era tido como algo futurístico, hoje é tido como presente e a cada dia mais se torna acessível a todas as pessoas, pois com a crescente dos hardwares e softwares *open-source*, a construção de sistemas de automação residencial se torna mais barata. Além disso, a inserção de objetos à internet faz com que a automação residencial se torne mais poderosa, podendo haver maiores possibilidades de automação de tarefas (TEZA, 2002; BOLZANI, 2004).

### 2.3 Microcontroladores, Sensores e Atuadores.

Microcontroladores são pequenos sistemas computacionais que englobam em um microchip, interfaces de entrada/saída, digitais e analógicas, e periféricos tais quais, memória RAM, memória FLASH, interfaces de comunicação serial, conversores analógicos/digitais (ADC) e temporizadores/contadores. Basicamente, o seu propósito é executar tarefas específicas a partir de programas escritos para eles chamados de firmware, armazenados em sua memória (CHASE, 2016; NICOLOSI, 2000).

Um microcontrolador tem a capacidade de realizar a leitura de sinais externos via sensores, executar programas com as tarefas a serem feitas, processar os sinais e enviar para atuadores os resultados esperados. Uma ilustração do diagrama básico de um sistema embarcado com um microcontrolador monitorando o ambiente pode ser visto na Figura 13.

Figura 13: Diagrama básico de um sistema embarcado dotado de um microcontrolador monitorando o ambiente.



Fonte: <http://www.lyfreitas.com.br/ant/pdf/Embarcados.pdf>

Os sensores e atuadores são componentes eletromecânicos que são utilizados na automação em geral. Em sumo, são a interface da residência inteligente e o ambiente. Basicamente eles permitem que o sistema “enxergue” e “manipule” esse ambiente em que estão. Para que o sistema “enxergue” o que está acontecendo no ambiente, são utilizados sensores que transformam parâmetros físicos recebidos do ambiente (temperatura, humidade, luminosidade e outros) em informação, sendo possível que o microcontrolador a partir de uma análise, saiba se é necessário tomar algumas decisões ou verificar se uma ação enviada foi executada com sucesso ou não. Para “manipular” esse ambiente, são utilizados os atuadores, dispositivos que a partir de impulsos elétricos recebidos são convertidos em energia mecânica, onde essa potência mecânica é enviada pelos atuadores aos elos fazendo com que os mesmos se movimentem. Como por exemplo, motores de passo que variam a posição angular para abrir ou fechar persianas, ou solenóides, que variam a posição linear, para o acionamento de uma bomba de água, ou um relé, que varia o seu chaveamento para ligar ou desligar uma tomada. (BOLZANI, 2004).

Existem uma grande infinidade de sensores, atuadores e microcontroladores no mercado que possibilitam o monitoramento de inúmeras grandezas físicas e a realização dos mais diversos eventos. Um dos microcontroladores mais utilizados atualmente é o Arduino.

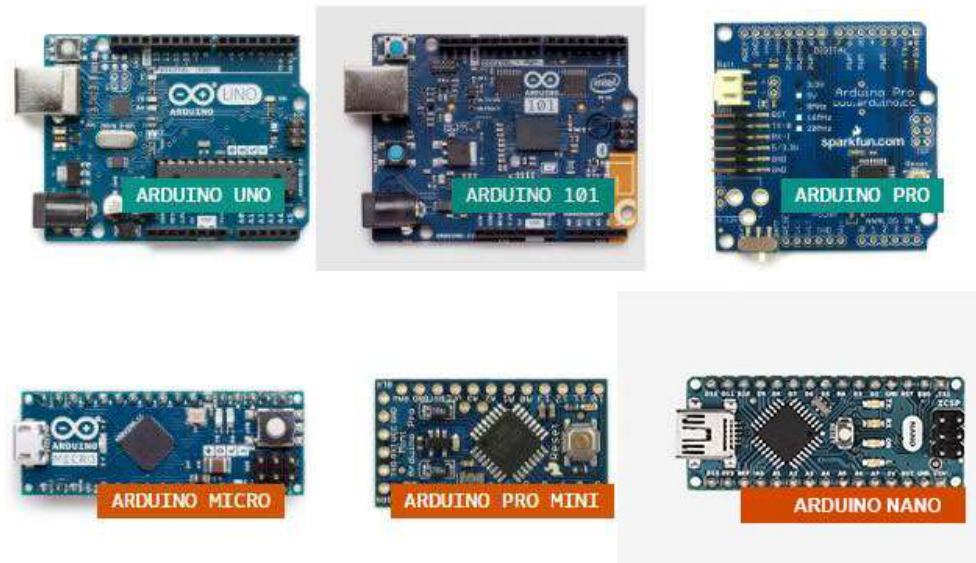
### **2.3.1 Plataforma Arduino**

O Arduino é uma plataforma de desenvolvimento de protótipos eletrônicos *open-source* que nos últimos anos tem sido o “cérebro” de milhares de projetos, desde os mais simples, de uso no nosso cotidiano, aos mais ousados, como instrumentos científicos, por exemplo. A sua vantagem sobre as outras plataformas de desenvolvimento de microcontroladores é a facilidade de sua utilização e sua ampla comunidade na internet, fazendo com que tenha uma grande troca de informações sobre essa plataforma (MCROBERTS, 2011).

Em termos práticos, o Arduino é um pequeno microcontrolador que pode ser programado para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele, como atuadores e sensores (MCROBERTS, 2011).

Existem várias versões do Arduino, todas baseadas em um microprocessador de 8 bits ATMEL AVR, como os Arduino de “entrada”, que são Arduinos para pessoas com conhecimentos básicos, ideais para quem quer aprender a mexer com eletrônica e programação. São eles: Uno, 101, PRO, Micro, Mini e Nano representados pela Figura 14.

Figura 14: Representação dos Arduinos Uno, 101, Pro, Micro e Mini.



Fonte: (ARDUINO, 2016b)

Já para quem precisa desenvolver projetos maiores, com maiores funcionalidades ou performances mais rápidas, tem-se o Arduino Mega e Zero, representados pela Figura 15.

Figura 15: Representação dos Arduinos Mega e Zero.

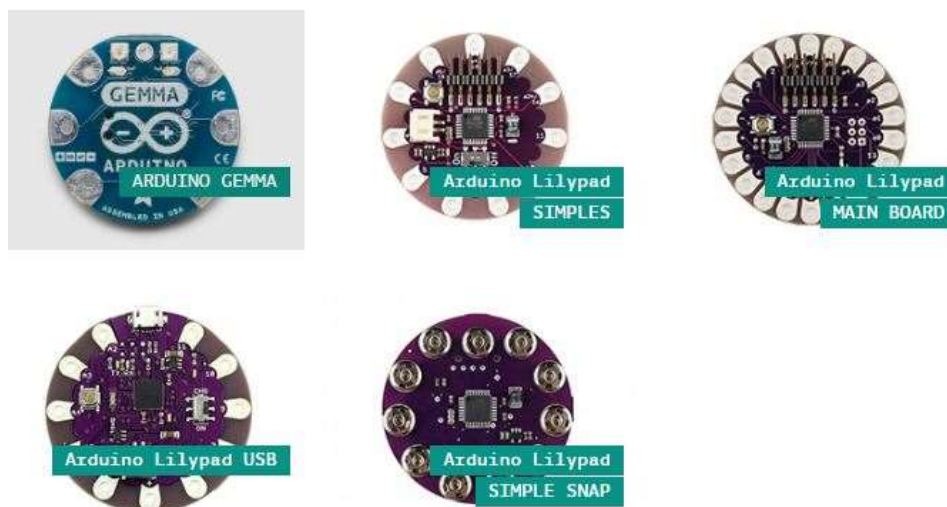


Fonte: (ARDUINO, 2016b).

E Arduinos para tecnologias vestíveis, Gemma, Lilypad Simple, Lilypad Main Board, Lilypad USB, e Lilypad Simple Snap, representados pela Figura 16.



Figura 16: Representação dos Arduinos vestíveis: Gemma, Lilypad Simples, Lilypad Main Board, Lilypad USB, e Lilypad Simple Snap.



Fonte: (ARDUINO, 2016b)

### 2.3.2 Plataforma ESP

A comunicação sem fios não é novidade para ninguém, e com a onda da internet das coisas o protocolo IEEE 802.11 começou a ser utilizado para fazer a comunicação entre os periféricos. Com isso, a empresa Espressif lançou em 2014 o microcontrolador ESP8266, que une o baixo custo a um tamanho reduzido. Desde então, foram lançados diversas variantes dessa placa, diferenciando a sua quantidade de GPIO disponíveis, interfaces I2C, SPI, e PWM, o seu tamanho e a sua capacidade de comunicação via ponte Serial-WiFi - recebe comandos via Serial e interagem com a rede WiFi usando o protocolo TCP/IP- e modo *standalone*- microcontroladores por meio da WiFi. Todos eles trabalham com 3.3V em sua alimentação, possuem uma CPU 32-bit RISC Tensilica Xtensa LX106 operando a 80MHz, podendo operar até 160 MHz, tem 64 KB de memória RAM, flash de 512KB a 4 MB, suporta o padrão IEEE 802.11b/g/n e os protocolos de segurança WEP, WPA, WPA2. A sua programação pode ser feita por comandos AT, em C, C++, ou utilizado a linguagem LUA. (CURVELLO, 2016b; FILIPPOV, 2016; FRITZEN, 2016; THOMSEN, 2016b).

A primeira grande diferença entre o Arduino e o ESP8266 é a comunicação wireless. A placa Arduino não possui um protocolo de comunicação sem fio próprio logo, caso seja necessária a comunicação via HTTP com o Arduino, você terá que acoplar uma Shield no Arduino, Wifi-Shield ou Ethernet-Shield, gerando mais custos ao projeto.

Dessa forma, os módulos com chip ESP8266 estão se popularizando rapidamente, sendo um ótima alternativa para os projetos de IoT. A figura 17 apresenta algumas das variantes desse módulo.

Figura 17: Variantes do ESP8266

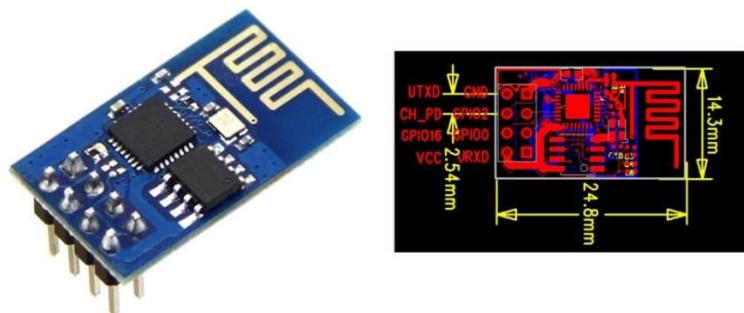


Fonte: (THOMSEN, 2016b)

Os ESP's mais comumente utilizados são:

- ESP-01 – 8 conectores, sendo, duas GPIOs que podem ser controladas via programação, Rx, Tx, CHPD, GND, VCC como pode ser visto na Figura 18, ele serve basicamente como um módulo para o Arduino:

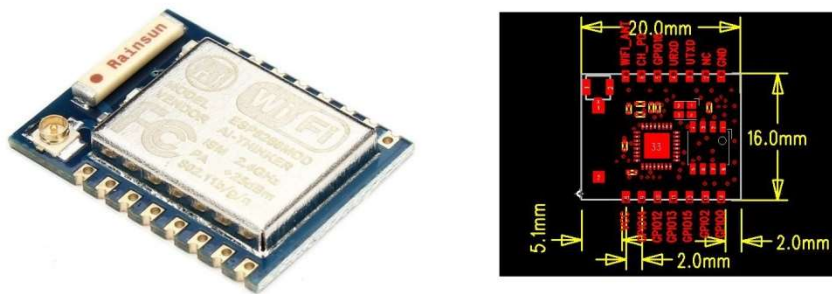
Figura 18: ESP-01 e seu circuito.



Fonte: (THOMSEN, 2016b)

- ESP-07 – 16 conectores, conta com uma antena de cerâmica embutida, além de um conector U-FI para antena externa. Esse módulo possui 9 GPIOs, que podem funcionar com pinos I2C, SPI e PWM. Na Figura 19 são mostrados a placa ESP-07 e o seu circuito.

Figura 19: ESP-07 e seu circuito.



Fonte: (THOMSEN, 2016b)

- ESP-12E – Semelhante ao ESP-07, como pode ser comparado visualizando as Figuras 19 e 20. Ele conta apenas com a antena interna, possui 22 pinos, sendo 11 para GPIO. Muito utilizado em outros módulos ESP8266, como o NodeMcu.

Figura 20: ESP-12E e seu circuito.



Fonte: (THOMSEN, 2016b)

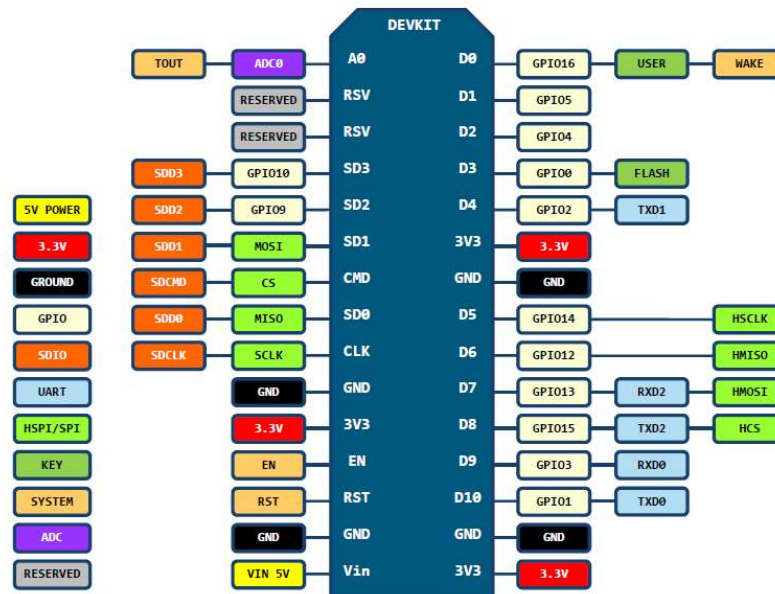
A gravação de firmware nos módulos ESP não é a coisa mais simples, sendo necessário um conversor USB/Serial (FTDI), para que os dados possam ser passados do computador ao módulo, além de usar 3.3V, o que não é uma tensão muito fácil de obter.

## 2.4 NodeMCU ESP-12E

O Módulo Wifi ESP8266 NodeMCU ESP-12E é uma das melhores placas da família ESP8266. Esta plataforma utiliza em sua construção o conversor TTL-Serial e um regulador de tensão 3.3V, onde pode ser alimentada via 5V no seu conector micro USB. Possui onze pinos digitais (todos, exceto D0 possuem interrupção, PWM, I2C e One Wire) e um pino analógico com resolução de 10 bits, onde os pinos digitais e analógicos trabalham em nível lógico de 3.3V. O Diagrama de pinagem dessa placa pode ser visualizado na Figura 21 (NODEMCU, 2016; HUANG, 2016).

Figura 21: Pinagem Node MCU V3

PIN DEFINITION



D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

Fonte: <https://github.com/nodemcu/nodemcu-devkit-v1.0>

A programação deste módulo pode ser feita em C, através da SDK da Espressif, na linguagem LUA e também pode ser utilizada C++ na IDE do Arduino utilizando a comunicação serial do seu micro USB ou via OTA (Over The Air), ou seja, através do WiFi você pode enviar os códigos para a placa. Abaixo, na Figura 21 temos a ilustração da placa NodeMCU ESP-12E.

Figura 22: NodeMCU ESP-12E.



Fonte: <https://github.com/nodemcu/nodemcu-devkit-v1.0>

Em um cenário em que tentamos deixar tudo conectado à internet agregando funcionalidades de controle de alguma aplicação, há um aumento expressivo na complexidade da mesma, caso seja necessário tratar requisições TCP/UDP e eventos diversos utilizando apenas uma plataforma. Dessa forma fica muito bagunçado, ainda

mais se for necessário fazer uma página Web, há uma grande mistura de código C, da aplicação, com código HTML, de exibição da página. A partir daí, há uma melhora na organização do projeto a delegação de funções (CURVELLO, 2016a).

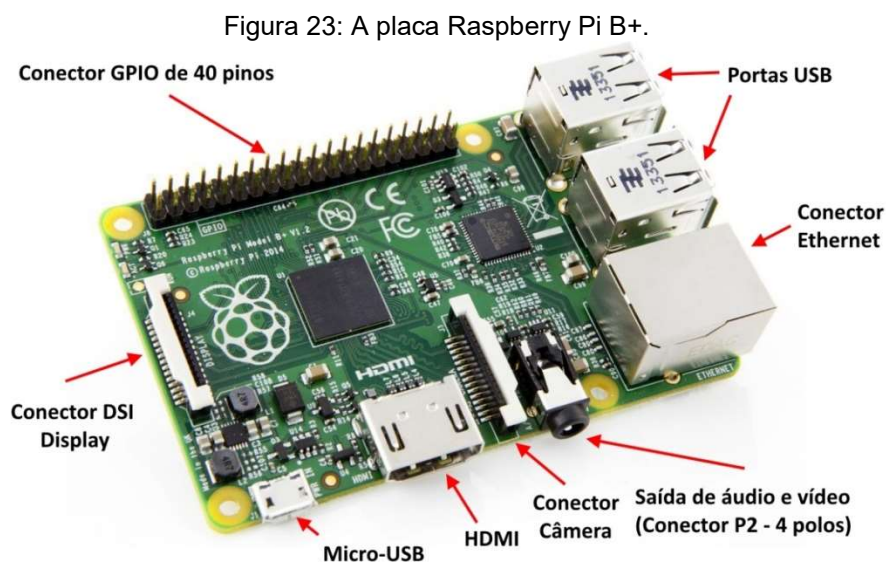
Portanto, além de utilizar microcontroladores, é necessário fazer uso de um microcomputador, como o Raspberry Pi, para tratar da parte alto nível, como requisições Web, exibição de páginas, consultas e gerenciamento do BD devido à sua capacidade de armazenamento e processamento. Já o microcontrolador tratará das atividades de baixo nível, tais como leitura de sensores, controle de atuadores.

## **2.5 Plataforma Raspberry Pi**

A origem da plataforma Raspberry Pi está atrelada a um grupo de cientistas da computação da Universidade de Cambridge liderados por Eben Upton no ano de 2005, com o objetivo de produzir um microcomputador barato, voltado ao ensino de programação a candidatos a uma vaga no curso de Ciência da Computação de Cambridge, já que eles notaram um declínio no nível de conhecimento de programação dos seus candidatos. Com o passar do tempo os cientistas perceberam o potencial do projeto, e então criaram a Raspberry Pi Foundation, e projeto deixou de ser destinado apenas aos ingressantes, passando a ser uma ferramenta educacional, podendo ser utilizada por crianças do mundo inteiro. Apesar disso, antes de ser lançada, por volta de 2011, a exposição na mídia e o interesse foi tão grande que se percebeu que essa placa não serviria apenas para amadores e o ensino de programação a crianças, mas também como uma plataforma alternativa para desenvolvedores de todo o mundo (NETTO, 2013; DENNIS, 2013).

Logo no início de 2012 o Raspberry Pi foi lançado para o público, com sua primeira geração disponível inicialmente em dois modelos. O modelo A com preço de 25 US\$, e o modelo B a 35 US\$. Os modelos A + e B + foram lançados um ano depois. Já a segunda geração do modelo B foi lançada em fevereiro de 2015, e a terceira; Raspberry Pi 3, modelo B, em fevereiro de 2016. Todas elas com preços entre 20 e 35 US\$ (DENNIS, 2013).

O Raspberry Pi versão B+ ilustrado na Figura 23, é equipado por 512 Mb de memória, entrada para armazenamento Micro SD, processador ARM single-core de 700 Mhz, 4 portas USB 2.0, conector ethernet e 40 conectores GPIO, que significa entrada e saída de uso geral em tradução livre. Sendo possível, assim como o Arduino, a conexão de sensores, displays e outros componentes utilizando o conector GPIO (RASPBERY PI, 2016).



Fonte: <http://www.arduinoecia.com.br/2014/10/o-que-e-o-raspberry-pi-e-onde-comprar.html>

A área de uso desse computador é a mais diversa possível, podendo ser usado além da área educacional, como um computador, central multimídia, servidor, ou para a programação de diversas linguagens, entre elas o Python, Java, C/C++. Por oferecer portas GPIO é possível que seja usado como controlador de um robô, sistemas embarcados ou automação (NETTO, 2013).

## 2.6 Protocolos de comunicação

Desde o princípio dos tempos a comunicação faz parte da sociedade, sendo uma ferramenta de troca de conhecimento e desenvolvimento social. O processo de comunicação consiste em transmitir uma informação a um ou vários receptores por meio de um canal de comunicação (meio onde circula a mensagem), que por sua vez decodifica a informação, interpreta e, eventualmente, responde ao emissor. A mensagem trocada é codificada em um sistema de sinais definidos, podendo ser

gestos, sons, linguagens naturais ou códigos e outras formas de comunicação. No meio digital, existem diversos protocolos que fazem com que padrões de comunicação sejam seguidos para que haja a troca de informações. O método de comunicação do computador com um dispositivo eletrônico varia de acordo com a tecnologia empregada no processo. A seguir serão descritos alguns protocolos de comunicação e a sua utilização.

### **2.6.1 Protocolo de Comunicação Serial**

A comunicação serial é bastante utilizada pela indústria eletroeletrônica por ser o padrão de comunicação entre hardwares mais simples, sendo na maioria das vezes reconhecida pelo padrão RS-232 (Recommended Standard). Sua principal característica está no fato de estabelecer comunicação entre dispositivos com a utilização de apenas 3 fios: Rx (recebimento de dados), Tx (transmissão de dados) e GND (comum)(NUMAJIRI, 2003). Segundo Andrade e Oliveira (2006), para se ter uma comunicação entre transmissor e receptor ambos devem concordar em determinados aspectos, tais quais:

- Bits por segundo (Boud Rate): Velocidade de transmissão serial de bits.
- Bits de dados (Character Length): Quantidade de bits de um caractere. Normalmente esse campo tem o padrão de 8 bits.
- Paridade (Parity): bit configurado em 0 ou 1, para assegurar que o número total de bits 1 no campo de dados é par ou ímpar.
- Bit de parada (Start and Stop Bits): bit de flag que determina início e fim de um bloco.
- Controle de fluxo (Flow Control): pode ser feito por hardware e software. A negociação por hardware usa linhas de controle para transmitir o sinal e receber as condições. O DSR(Data Set Ready), DTR(Data Terminal Ready), CTS(Clear to Send) e RTS(Request to Send) são os padrões de negociação por hardware da comunicação serial. Há também o software para controle de fluxo dos sinais (XON e XOFF) que habilita ou desabilita a transmissão.

Dessa forma, torna-se simples e prático criar o processo de comunicação entre o computador e o dispositivo microcontrolador. Além dos computadores e

microcontroladores, vários módulos e sensores também utilizam desta comunicação para transmitir dados.

### **2.6.2 Protocolo HTTP**

O protocolo de comunicação HTTP é o protocolo de transferência utilizado em toda a Word Wide Web. Ele é usado na Internet para acessar informações seguindo a estratégia requisição/resposta no paradigma cliente/servidor. Essa é uma arquitetura onde o processamento da informação é dividido em módulos ou processos distintos. Um processo é responsável pela manutenção da informação (Servidor), enquanto o outro é responsável pela obtenção dos dados (Cliente). Em uma rede de computadores existem uma ou mais máquinas que atuam como servidores, disponibilizando recursos para as demais máquinas, as quais atuam como clientes (BATTISTI, 2001).

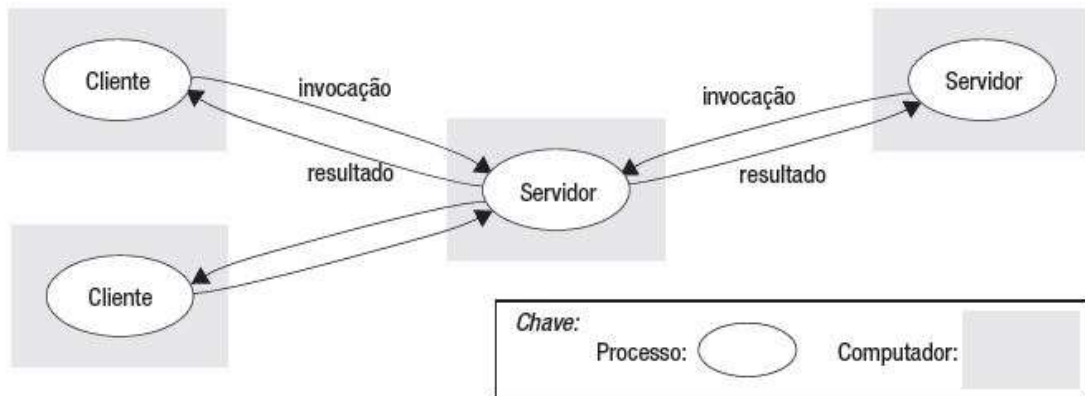
Segundo Coulouris et al. (2013), esse protocolo é implementado sobre o TCP. Em sua versão original, cada interação cliente-servidor consiste nas seguintes etapas:

- O cliente solicita uma conexão com o servidor na porta HTTP padrão ou em uma porta especificada no URL;
- O cliente envia uma mensagem de requisição para o servidor;
- O servidor envia uma mensagem de resposta para o cliente;
- A conexão é fechada.

A Figura 24 mostra a estrutura da arquitetura usada neste protocolo, onde os processos assumem os papéis de clientes ou servidores. Em particular, os processos clientes interagem com processos servidores, localizados possivelmente em distintos computadores hospedeiros para acessar os recursos compartilhados, tais quais, bancos de dados, aplicações ou páginas estáticas que estes gerenciam.



Figura 24: Arquitetura de comunicação cliente-servidor.



Fonte: (COULOURIS, 2013).

Diferentemente das máquinas servidoras e dos computadores, os dispositivos usados na IoT possuem poder computacional restrito, o que limita a utilização do protocolo HTTP nesses elementos. Para resolver esse problema, foram desenvolvidos diversos protocolos na camada de aplicação especificamente para recuperar informações de dispositivos com baixo poder computacional destacando-se o protocolo MQTT.

### 2.6.3 Protocolo MQTT

Desenvolvido por Andy Stanford-Clark e Arlen Nipper, com suas origens na IBM e Eurotech, o MQTT, Message Queueing Telemetry Transport, é um protocolo de comunicação M2M (Machine-to-Machine) *open-source*, projetado para ser simples, leve e de fácil implementação. Trata-se de um protocolo de mensagens baseado na arquitetura publish/subscribe, cuja proposta é atuar principalmente, mas não exclusivamente, em ambientes onde há limitações computacionais, tais quais, baixa largura de banda, alta latência e dispositivos embarcados com recursos limitados de memória e processamento. (IBM; EUROTECH, 2017; BATALLA et al., 2017). Segundo Brown (2017), o MQTT é perfeito para o uso em dispositivos embarcados porque é um protocolo que:

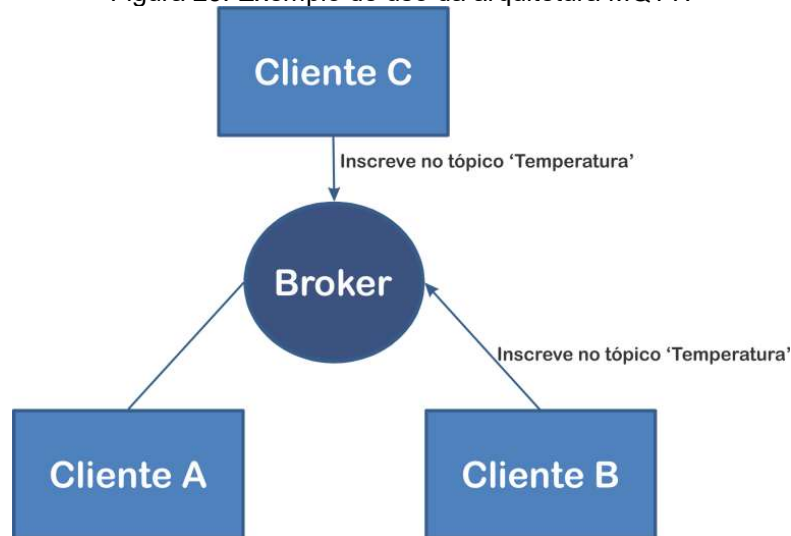
- É assíncrono, com vários níveis diferentes de qualidade de serviço, o que é importante nos casos em que as conexões de Internet não são confiáveis;

- Envia mensagens curtas e precisas que o tornam útil para situações de largura de banda baixa;
- Não requer muito software para implementar um cliente, o que o torna excelente para dispositivos como com memória limitada;

### 2.6.3.1 Arquitetura do protocolo MQTT

Seguindo o modelo da arquitetura publish/subscribe, que é uma arquitetura que tem o princípio de publicação de mensagens e subscrição de tópicos, vários clientes se conectam a um servidor, chamado de broker, e se inscrevem em tópicos em que eles estão interessados. Os clientes podem também se conectar ao broker e publicar mensagens nos tópicos (LIGHT, 2017). Vamos utilizar o exemplo adaptado de Jeffrey (2017) para um melhor entendimento: Imaginando uma rede com três clientes e um broker central, todos os três clientes abrem conexões TCP com o broker, sendo os clientes B e C se inscrevendo no tópico: “Temperatura” conforme a Figura 25.

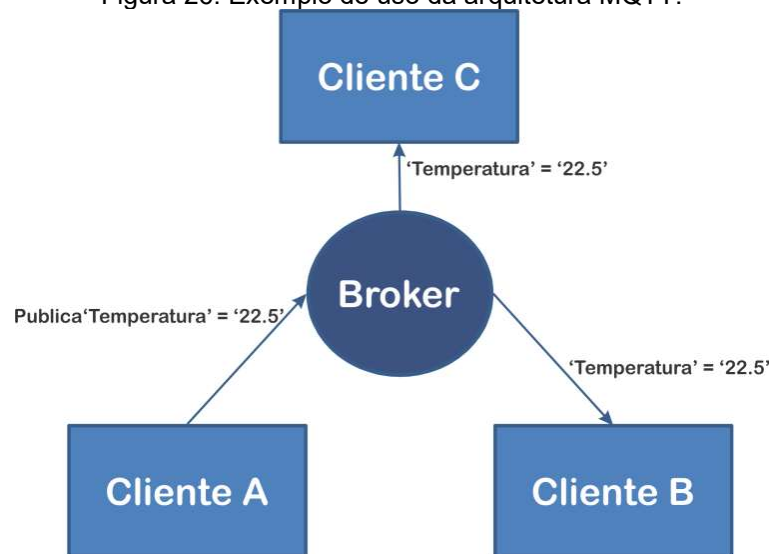
Figura 25: Exemplo de uso da arquitetura MQTT.



Fonte: Adaptado de (JAFNEY 2014).

Após essa conexão, o Cliente A publica o valor 22.5 no tópico Temperatura e o Broker encaminha a mensagem para todos os clientes inscritos neste tópico, como pode ser visualizado na Figura 26.

Figura 26: Exemplo de uso da arquitetura MQTT.



Fonte: Adaptado de (JAFNEY 2014).

Os tópicos são tratados como uma hierarquia, usando a barra (/) como separadora. Dessa forma, é permitido o uso de tópicos comuns, assim como um sistema de arquivos, como, por exemplo, casa01/temperatura/cozinha casa02/temperatura/sala.

### 2.6.3.2 Qualidade de serviço

O MQTT define três níveis de qualidade de serviço (QoS), ela é que define quanto o broker e o cliente tentará garantir que uma mensagem seja recebida. As mensagens podem ser enviadas a qualquer nível de QoS, e os clientes podem tentar se inscrever em tópicos em qualquer nível de QoS. Isso significa que o cliente escolhe a QoS máxima que receberá. Os níveis mais elevados de QoS são mais confiáveis, mas envolvem maior latência e têm maiores requisitos de largura de banda (LIGHT, 2017). Os níveis são os seguintes:

- 0: O broker / cliente enviará a mensagem uma vez, sem confirmação;
- 1: O corretor / cliente enviará a mensagem pelo menos uma vez, com confirmação necessária;
- 2: O corretor / cliente enviará a mensagem exatamente uma vez usando um handshake de quatro passos;

## 2.7 Desenvolvimento Web

Em uma pesquisa feita em 2015 organizada pelo Comitê Gestor da Internet no Brasil (CGI.br), Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (Cetic.br), e pelo Núcleo de Informação e Coordenação do Ponto BR (NIC.br) mostrou que cerca de 58% da população do Brasil usa a internet, sendo essa proporção 5% superior à registrada no levantamento de 2014. Esse mesmo estudo mostrou também que o telefone celular é o dispositivo utilizado para o acesso individual da internet pela maioria dos usuários, cerca de 89%, seguido pelo computador de mesa (40%), computador portátil ou notebook (39%), tablet (19%), televisão (13%), e videogame (8%). De acordo com o levantamento, 56% da população brasileira usaram a internet no telefone celular nos três meses antes da pesquisa. A proporção era de 47% em 2014, e de 31% em 2013. (BOCCHINI, 2016), (CETIC.br, 2016).

A multiplicidade de dispositivos no mercado promoveu um grande problema aos desenvolvedores Web, que por muito tempo compartilharam a ideia que a Web teria que ter páginas de tamanho fixo, sempre fazendo telas apenas para as larguras de telas mais comuns à época. Como resposta a esse problema, foi desenvolvido um conjunto de técnicas para adaptar os sites a qualquer dispositivo, chamado de Design Responsivo (LOPES, 2013).

O termo Web Design Responsivo surgiu em 2010 em um artigo feito por Ethan Marcotte no blog “A List Apart”, sendo a palavra ‘Responsivo’ vinda da arquitetura, a qual se refere a técnicas para construções e materiais de adaptarem ao ambiente e as pessoas que interagem com ele. Segundo Marcotte(2010):

Ao invés de criar designs desconectados para cada um do crescente número de dispositivos web, nós podemos tratá-los como faces da mesma experiência. Podemos criar para uma experiência de visualização ideal, mas embutir tecnologias padronizadas nos nossos designs para fazê-los não apenas mais flexíveis, mas mais adaptados para a mídia que os renderiza.

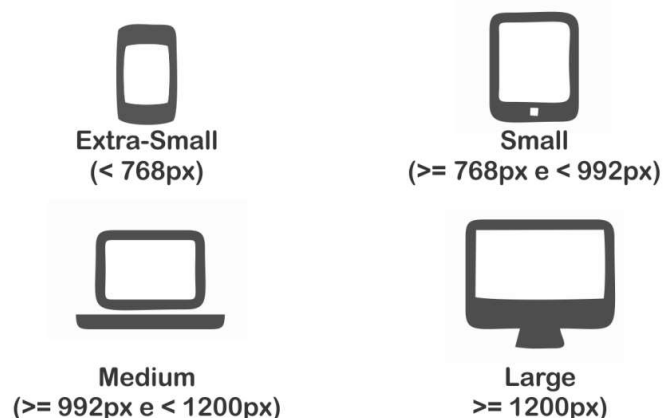
Antes do Design Responsivo, era comum a criação de uma ou mais versões desktop e mobile do mesmo site, o que dificultava a manutenção do conteúdo. Ainda segundo Marcotte (2010) o princípio de Web Design Responsivo tem como chave ser um design flexível e adaptável, que se ajuste às características do navegador, do

dispositivo e o contexto do usuário por meio de tecnologias padronizadas (HyperText Markup Language [HTML] e Cascading Style Sheets [CSS]). Sendo os seus três pilares, layout fluido, uso de media queries e imagens flexíveis. Seguindo esses princípios e chaves, foram criados diversos frameworks, entre eles, se destacam o Bootstrap e o Materialize.

Criado em 2010 por um design e um desenvolvedor da rede social Twitter, Mark Otto e Jacob Thorton respectivamente, o Bootstrap é um dos mais populares frameworks JavaScript, HTML e CSS open source do mundo para o desenvolvimento de sites e aplicações Web Responsivo. O Bootstrap, segue a premissa de desenvolvimento “Mobile First”, que se trata de fazer o design e implementar o código para mobile primeiro e, depois, ajustar para o desktop. Com isso fez-se necessário a configuração de breakpoints em sua arquitetura, que são os pontos de quebra do layout fluido de uma página, onde uma reestruturação maior é necessária, fazendo assim, o reposicionamento do conteúdo na página. Por padrão, o menor breakpoint é 768px, que é a largura máxima de tela em dispositivos extra pequenos, tais quais telefones. O segundo breakpoint, 992px, para dispositivos pequenos com a largura entre 768 e 992, tais quais tablets, o terceiro e último breakpoint é 1200px, entre 992 e 1200 de largura para dispositivos médios como notebooks, e para dispositivos grandes, a partir de 1200, como desktops, como pode ser visto na Figura 27. Dessa forma, utilizando o padrão Bootstrap, é possível definir até quatro layouts de organização a partir da largura da sua viewport (BOOTSTRAP, 2016), (SILVA, 2015).

Figura 27: Tamanho de telas suportadas no sistema de grids do Bootstrap.

## Breakpoints Disponíveis



Fonte: O autor

### 3. METODOLOGIA

O presente trabalho caracteriza-se como um estudo do tipo bibliográfico exploratório com vista ao desenvolvimento experimental, ou seja, a implementação de ferramentas e algoritmos tomando como base artigos, livros e impressos diversos sobre o tema (GIL, 2010).

Na maioria dos trabalhos acadêmicos, a pesquisa bibliográfica se faz necessária para fornecer uma fundamentação teórica ao trabalho e o estado da arte do tema proposto. Esse tipo de metodologia confere a possibilidade de investigar de forma mais ampla uma série de fenômenos que não estão ao alcance do pesquisador (GIL, 2010). O desenvolvimento experimental é definido como:

Um trabalho sistemático, que utiliza conhecimentos derivados da pesquisa ou experiência prática com vistas à produção de novos materiais, equipamentos, políticas e comportamentos, ou à instalação ou melhoria de novos sistemas e serviços (GIL, 2010, p. 27).

#### 3.1 Planejamento

No início do processo de desenvolvimento de um sistema é extremamente importante analisar cada processo e modelar como o sistema deve se comportar, sempre obedecendo aos requisitos levantados.

A partir de reuniões com os orientadores deste trabalho foram levantadas as seguintes questões de pesquisa:

- Como é feita a leitura da energia elétrica?
- Qual *hardware* disponível melhor se adequa para execução das tarefas propostas?
- Quais equipamentos, componentes eletrônicos e ferramentas que melhor se adaptam ao projeto?
- Quais softwares se adequam ao escopo do projeto, sendo eles open-source?

Seguindo essas questões de pesquisa, desenvolveu-se o modelo de sistema especificado nas seguintes seções.

## **3.2 Modelagem do Sistema**

O foco deste trabalho é o desenvolvimento de um sistema que é composto por Hardware e Software. Na escolha das tecnologias foi considerado utilizar hardware e software open-source e de baixo custo, ao mesmo tempo em que essas ferramentas sejam condizentes com as necessidades do projeto.

Antes de iniciar o desenvolvimento do sistema, foi feita toda a análise necessária, onde foram levantados todos os requisitos funcionais e não funcionais, além de criar um diagrama de caso de uso seguido dos modelos conceituais, lógicos e físicos do banco de dados. Na sequência o sistema será apresentado e detalhado cada um dos itens citados acima.

### **3.2.1 Apresentação do sistema**

O ÉdaSuaConta possibilita a aquisição de dados de leitura de um sensor de corrente através da plataforma Arduino e apresentação desses dados a partir de um servidor web rodando em um Raspberry Pi. O sensor fica conectado a uma placa Arduino e os dados são enviados para o Raspberry Pi que fica dedicado a apresentá-los em um sítio e armazená-los em banco de dados. O usuário interage com o sistema por meio de uma página web, onde é possível a visualização desses dados atualizados a cada segundo na tela de consumo instantâneo do sistema, e a visualização de relatórios de leituras e gráficos do histórico das leituras realizadas. O sistema permite também que o usuário ligue e desligue componentes remotamente, possuindo ainda área administrativa para cadastro, exclusão e alteração de usuários.

### **3.2.2 Requisitos funcionais**

RF01- O sistema deve permitir o cadastro de usuários.

RF02- O sistema deve permitir a alteração e exclusão de usuários.

RF03- O sistema deve permitir a geração de relatórios de dados coletados dos sensores.

RF04- O sistema deve permitir aos usuários do sistema consultar dados dos sensores instantaneamente.

RF05- O sistema deve permitir o controle de tomadas, ligá-las ou desligá-las.

RF06- O sistema deve obter os dados do sensor e salvar no banco de dados.

### **3.2.3 Requisitos não funcionais**

NF01 – O sistema deve possuir um mecanismo de segurança para evitar que pessoas não autorizadas tenham acesso ao sistema ou os dados do sistema.

NF02– O sistema deve possuir dois tipos de usuários: usuário e administrador.

NF03– O sistema deve ser acessado via página web.

NF04– Somente o administrador e os usuários cadastrados e autenticados podem operar o sistema.

NF05– Somente o administrador pode cadastrar, alterar, excluir e consultar usuários.

NF06- Os dados obtidos dos sensores devem ser armazenados em banco de dados.

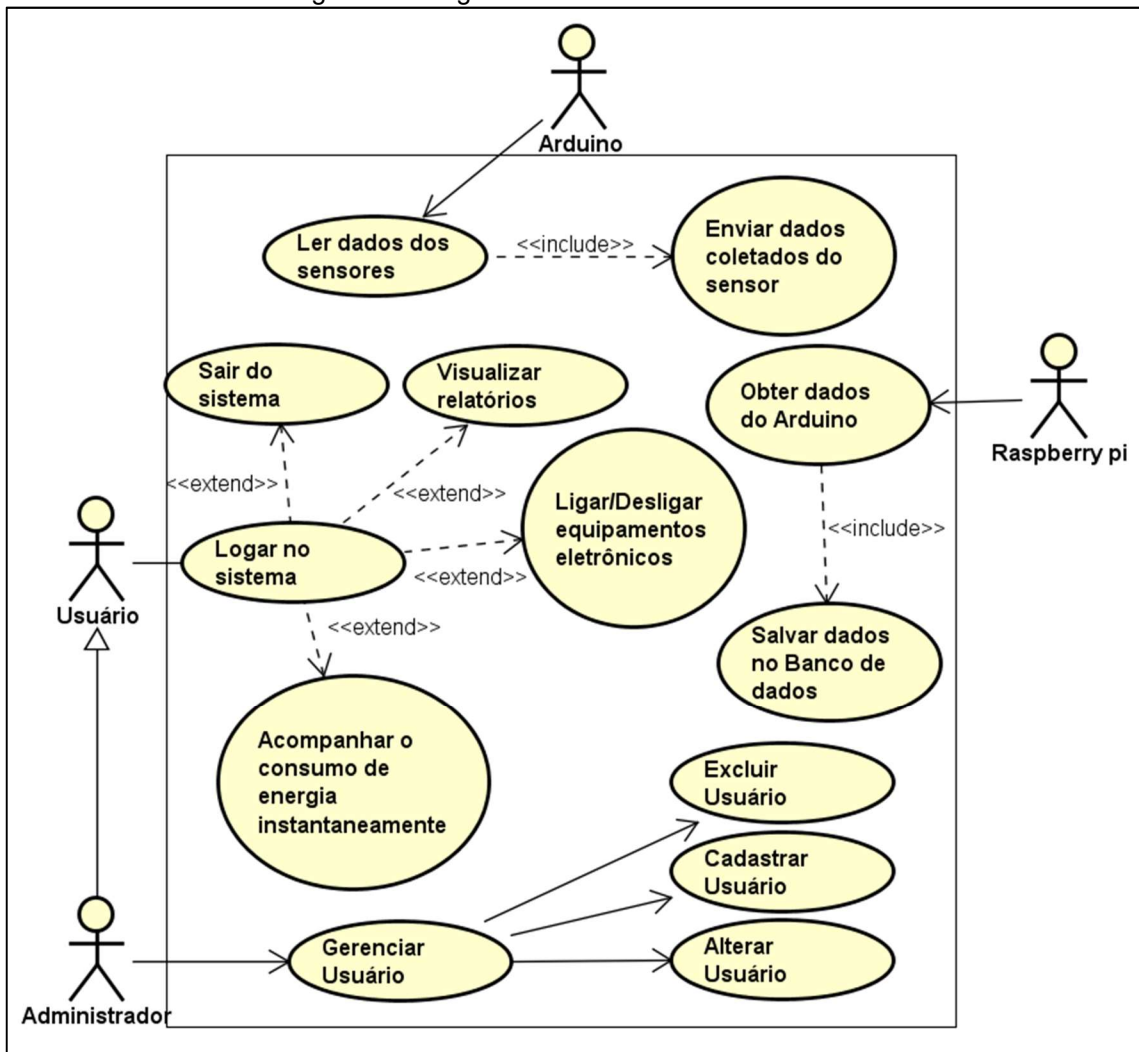
NF07- A interface do aplicativo deverá se ajustar aos diferentes tamanhos de tela de dispositivos como desktops, tablets e smartphones.

NF08- O sistema deve ser feito com hardware e software open-source.

Dos requisitos funcionais e não funcionais apresentados acima derivou-se o seguinte caso de uso descrito na Figura 28.



Figura 28: Diagrama de caso de uso do sistema



Fonte: O Autor.

### 3.2.4 Atores do Sistema

Tabela 1: Diagrama de caso de uso do sistema

Nome	Descrição
<b>Usuário</b>	Representa os moradores da residência que acessam o site do sistema de gerenciamento da casa.
<b>Administrador</b>	Representa o administrador que acessa o site do sistema de gerenciamento da casa.
<b>Arduino</b>	Representa a placa Arduino, que faz a leitura de sensores e envia os dados obtidos.
<b>Raspberry Pi</b>	Representa a placa Raspberry Pi, que é o servidor web, sistema de banco de dados e unidade controladora do sistema.

Fonte: O autor.

### 3.2.5 Casos de Uso

- **Logar no sistema:** O usuário (por herança o Administrador atua sobre esse caso de uso) efetua o login na página web do sistema informando o seu usuário e senha, e a partir daí, será encaminhado a página inicial do sistema.
- **Sair do sistema:** todo usuário que conseguir logar no sistema terá acesso a um botão para sair do mesmo, onde, após confirmação, será levado de volta à página inicial.
- **Visualizar relatórios:** Caso de uso onde o usuário visualiza os dados de leitura do sensor a partir dos dados gravados no banco de dados.
- **Acompanhar o consumo de energia instantaneamente:** Descreve as operações para um usuário visualizar os dados da leitura dos sensores instantaneamente.

- **Ligar/Desligar equipamentos eletrônicos:** Esse caso de uso descreve as operações para um usuário controlar equipamentos eletrônicos conectados a relés no sistema de automação, permitindo ligar ou desligá-los via página web.
- **Gerenciar usuário:** Caso de uso que descreve as operações para um Administrador do sistema visualizar os usuários e realizar o gerenciamento (Cadastrar, excluir e alterar) dos mesmos no sistema web.
- **Cadastrar usuário:** Onde o administrador do sistema cadastra um novo usuário à plataforma.
- **Excluir usuário:** Onde o Administrador do sistema exclui um usuário da plataforma.
- **Alterar usuário:** Onde um Administrador do sistema pode alterar as informações sobre o usuário do sistema.
- **Ler dados dos sensores:** Caso de uso que descreve as operações para o microcontrolador (Arduino) fazer a leitura do sensor SCT-013.
- **Enviar dados dos sensores:** Onde o Arduino manda para o servidor (Raspberry Pi) os dados lidos do sensor via USB.
- **Obter dados dos sensores:** Caso de uso que descreve as operações para o servidor (Raspberry Pi) obter os dados de leitura dos sensores enviados pelo Arduino
- **Salvar no banco de dados:** Após o Raspberry Pi obter os dados de leitura dos sensores, salva-os no banco de dados.

### 3.2.6 Modelo do Banco de Dados

Uma vez que todos os requisitos foram levantados e analisados, o próximo passo foi a criação de um esquema conceitual, lógico e físico para o banco de dados. O esquema que se segue é bem simples, contando com apenas 3 tabelas, as quais não se relacionam. Foi pensado em um modelo simples para um melhor entendimento e funcionamento no hardware disponível.

Figura 29: Modelo Conceitual do Banco de dados.

energy.usuarios	energy.medidas	energy.medidasHora
id : int(10) unsigned	horario : datetime	ano : varchar(4)
nome : varchar(100)	corrente : float	mes : varchar(2)
usuario : varchar(50)	potencia : float	dia : varchar(2)
senha : varchar(50)	id : int(11)	hora : varchar(2)
tipo : tinyint(4)		id : varchar(10)
		potencia : float

Fonte: O Autor.

Um modelo lógico de um BD relacional deve definir quais as tabelas que o banco contém e, para cada tabela, quais os nomes das colunas. O modelo lógico para o BD em questão é o seguinte:

A tabela medidas têm como atributos:

- id: Chave primária de identificação de registro, atributo auto incrementado;
- horário: contém a data e hora do momento em que o sensor fez a leitura da corrente;
- corrente: Valor lido pelo sensor;
- potencia: valor lido pelo sensor multiplicado por 220 (pela fórmula potência aparente=Tensão\*Corrente).

A tabela medidashora foi necessária para dar granularidade ao sistema, pelo motivo de em apenas uma hora o sistema ter mais de 3.000 registros, colocando isso em um dia, totalizando cerca de 86.400 registros em um só dia. Visto que o Raspberry Pi estava demorando muito para fazer as consultas, houve a necessidade de granular esses registros em horas. Ou seja, de hora em hora é rodado um script para salvar a soma da potência por 36000, que resulta na quantidade de kw gasta na última hora. Essa tabela tem os seguintes atributos:

- id: Chave primária de identificação de registro, sendo o seu registro feito da seguinte forma: (HoraDiaMesAno);
- ano: ano em que a leitura foi feita pelo sensor
- mes: mês em que a leitura foi feita pelo sensor;
- dia: dia em que a leitura foi feita pelo sensor;
- potencia: soma da potência consumida na última hora da tabela medidas por 36000, que resulta na quantidade de kw gasta na última hora.

A tabela usuários tem como atributos:

- id: Chave primária de identificação do usuário, atributo auto incrementado;
- nome: Nome do usuário;
- usuário: alias do nome do usuário, utilizado para o acesso ao sistema
- senha: senha do usuário;
- tipo: como o sistema tem dois tipos de usuário: Administrador e Usuário, então faz-se necessário o atributo tipo para distinguir o tipo de usuário. 1 para Usuário comum e 2 para Administrador.

O Modelo físico do banco de dados é derivado a partir do modelo lógico, sendo representado pelos comandos do MySql nas Tabelas 2, 3, 4 e 5. O MySql escolhido para esse projeto por se tratar de um sistema de gerenciamento de banco de dados (SGBD) relacional que é robusto e escalável, multi plataforma (Mac OS, Linux e Windows). Além de ser um software de código aberto, sendo ele um dos mais utilizados no mercado.

A Tabela 2 contém os comandos para a criação do banco de dados 'energy'.

Tabela 2: Quadro com o comando de criação do banco de dados 'energy'

```
CREATE DATABASE IF NOT EXISTS `energy` DEFAULT CHARACTER SET  
utf8 COLLATE utf8_general_ci;  
  
USE `energy`;
```

Fonte: O Autor.

A Tabela 3 contém os comandos para a criação da tabela 'medidas' com os atributos horário, corrente, potencia e id.

Tabela 3: Quadro com o comando de criação da tabela 'medidas'

```
CREATE TABLE IF NOT EXISTS `medidas` (  
  
  `horario` datetime NOT NULL,  
  
  `corrente` float NOT NULL,  
  
  `potencia` float NOT NULL,  
  
  `id` int(11) NOT NULL AUTO_INCREMENT,
```

```
PRIMARY KEY (`id`)  
)ENGINE=MyISAM AUTO_INCREMENT=344320 DEFAULT CHARSET=latin1;
```

Fonte: O Autor.

A Tabela 4 contém os comandos para a criação da tabela 'medidasHora' com os atributos ano, mês, dia, hora id e potencia.

Tabela 4: Quadro com o comando de criação da tabela 'medidasHora'

```
CREATE TABLE IF NOT EXISTS `medidasHora` (  
  `ano` varchar(4) NOT NULL,  
  `mes` varchar(2) NOT NULL,  
  `dia` varchar(2) NOT NULL,  
  `hora` varchar(2) NOT NULL,  
  `id` varchar(10) NOT NULL,  
  `potencia` float NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

Fonte: O Autor.

A Tabela 5 contém os comandos para a criação da tabela 'medidas' com os atributos id, nome, usuário, senha e tipo.

Tabela 5: Quadro com o comando de criação da tabela 'usuarios'

```
CREATE TABLE IF NOT EXISTS `usuarios` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `nome` varchar(100) NOT NULL,  
  `usuario` varchar(50) NOT NULL,  
  `senha` varchar(50) NOT NULL,
```

```
`tipo` tinyint(4) NOT NULL DEFAULT '1',  
PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

Fonte: O Autor.

### **3.3 Metodologia de Testes**

Durante toda a fase de implementação, foram realizados testes de integração à medida que o protótipo era construído. Ao final de cada teste, outros eram realizados para a verificação de conformidade do mesmo, sendo sempre considerados os requisitos. À medida que eram detectados erros, os mesmos eram corrigidos até que entrassem em total conformidade com os requisitos do sistema. Alguns testes são mostrados na seção 5.

## **4. FERRAMENTAS UTILIZADAS**

Este Capítulo apresenta uma descrição do hardware e software utilizados na implementação do sistema, sendo divididos entre Hardware e Software para um melhor entendimento.

### **4.1 Hardware**

Para o Hardware, foi escolhido o microcomputador Raspberry Pi B+ para ser o servidor web e fazer todo o processamento do banco dados. O Arduino foi escolhido por ser utilizado em outros projetos que fazem o uso do sensor de corrente SCT013-000 para a leitura dos dados da corrente e também por conseguir se comunicar com a serial do Raspberry Pi. O módulo WiFi ESP8266 NodeMCU ESP-12E, foi escolhido

para receber os comandos via WiFi e controlar o módulo de Relés. Como o Raspberry Pi B+ não tem embutido uma placa WiFi, foi necessário a obtenção de um adaptador WiFi USB.

#### **4.1.1 Preço do Hardware Utilizado**

Tendo como base o requisito de ser um sistema de baixo custo, necessitou-se de uma pesquisa de preços dos hardwares a serem utilizados. A pesquisa de preço foi feita no dia 20 de maio de 2017, no site nacional do MercadoLivre (<http://www.mercadolivre.com.br/>) e nas lojas virtuais internacionais AliExpress(<http://www.aliexpress.com>), Amazon (<https://www.amazon.com/>) e Adafruit (<https://www.adafruit.com/>). O padrão de preço adotado foi a partir do menor preço encontrado, sendo que para as lojas internacionais o preço foi dado em dólares por se tratar da moeda de compra em tais sites. Além disso, levou-se em consideração o preço do frete. A Tabela 6 mostra os dados da pesquisa.



Tabela 6: Preço dos Hardwares utilizados.

	<b>Mercado Livre</b>	<b>Aliexpress</b>	<b>Adafruit</b>	<b>Amazon</b>
<b>Raspberry Pi B+</b>		>\$35	\$29.9	>\$35
<b>Raspberry Pi 3 B</b>	> R\$150	>\$37	\$39.95	>\$43
<b>Arduino Nano</b>	> R\$26	>\$3	-	>\$5
<b>Arduino Mega</b>	> R\$45	>\$6	45\$	>\$10
<b>NodeMCU</b>	> R\$35	>\$7.5	-	>\$7.99
<b>SCT013-000</b>	> R\$45	>\$14.5	-	>\$15
<b>Relé 8 Canais</b>	> R\$35	>\$6	-	>\$10
<b>Adaptador WiFi</b>	> R\$20	>\$5	-	>\$5

Fonte: O Autor

O preço do Raspberry Pi 3B foi pesquisado por se tratar da placa mais atual do Raspberry Pi, sendo assim, a mais encontrada para venda em tais sites, já que a placa B+ está obsoleta.

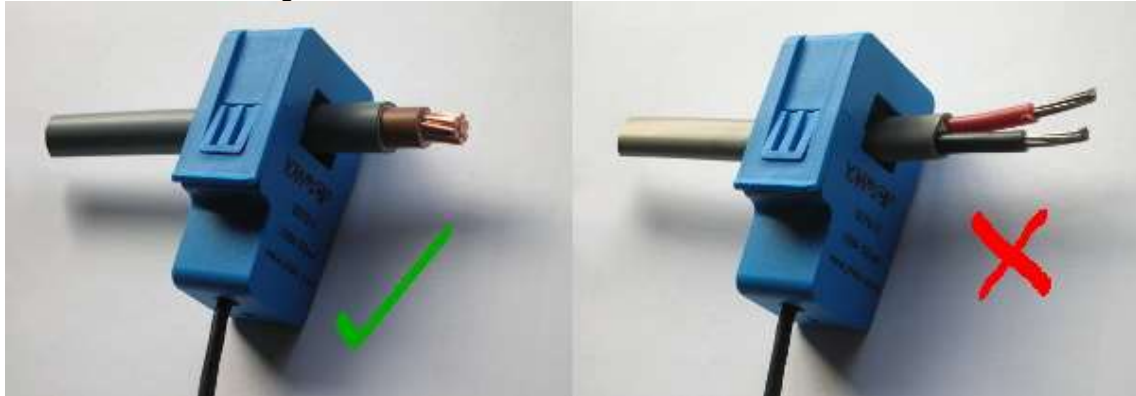
Para que fosse possível a comunicação entre o sensor de corrente e o Arduino, foi necessária a prototipação de um circuito, que pode ser vista na próxima seção.

#### 4.1.2 Prototipação do circuito para a leitura do sensor de corrente

Os transformadores de corrente (CT's) são sensores que conseguem medir a corrente alternada, sendo úteis para a medição de consumo ou geração de eletricidade. O sensor utilizado foi o SCT013-000, e sua maior vantagem é a sua possibilidade de medição de corrente alternada sem a necessidade de interromper o circuito para realizar a medição. Nesse caso é necessário apenas “envolver” um dos fios de energia, não ambos, como pode ser visto na Figura 30. Isso acontece pelo motivo de que um fio que transporta uma corrente elétrica por ele gera um campo magnético em torno de si, dessa forma se esse sensor for envolvido nesses dois fios ele irá medir a soma das correntes em cada um dos condutores. No caso de um cabo de núcleo duplo

ou de núcleo duplo e terra, a mesma corrente estará fluindo em direções opostas, sendo a sua soma igual à zero. (OPENENERGYMONITOR, 2016b).

Figura 30: Forma correta de envolver o sensor no fio.



Fonte: (OPENENERGYMONITOR, 2016b).

Em seu datasheet, (Anexo 1), é informado que a saída de dados deste sensor é do tipo de variação de corrente, onde o sinal precisa ser convertido em um sinal de tensão com um resistor de carga para que possa ser interpretado pelo Arduino.

Seguindo os passos de como calcular o resistor de carga adequado em OpenEnergyMonitor(2016c), derivou-se a seguinte fórmula para o cálculo do resistor de carga:

Resistência de carga ideal (ohms) =  $(AREF * N^{\circ} \text{ de voltas do CT}) / (2\sqrt{2} * \text{corrente de pico primária})$ .

$$\text{Resistência de carga ideal} = (2,5 \text{ V} * 2000) / 2\sqrt{2} * 141,4 = 35,4 \Omega.$$

Explicando melhor esse cálculo, sendo a faixa de corrente do sensor de 0A a 100A, teremos a corrente máxima de 100A a ser lida por tal. Foi convertido a corrente **RMS** (raiz do valor quadrático médio) máxima na corrente máxima multiplicado por  $\sqrt{2}$ .

$$\text{Corrente de pico primária} = \text{Corrente RMS} \times \sqrt{2} = 100 \text{ A} \times 1,414 = 141,4\text{A}$$

Em seguida, foi dividida a corrente de pico pelo número de voltas na CT para dar a corrente de pico na bobina secundária.

Verificando o datasheet novamente, verificou-se que o YHDC SCT-013-000 CT tem 2000 voltas, então a corrente de pico secundária será:

$$\text{Corrente de pico secundária} = \text{Corrente de pico primária} / \text{N}^\circ \text{ de voltas} = 141,4\text{A} / 2000 = 0,0707\text{A}$$

Para melhorar a resolução da medição, a tensão entre a resistência de carga na corrente de pico deve ser igual a metade da tensão de referência analógica do Arduino (AREF / 2). Como a tensão de referência no Arduino é de 5 V, teremos:

$$\text{Resistência de carga ideal} = (\text{AREF} / 2) / \text{Corrente de pico secundária} = 2,5\text{V} / 0,0707\text{ A} = 35,4\ \Omega$$

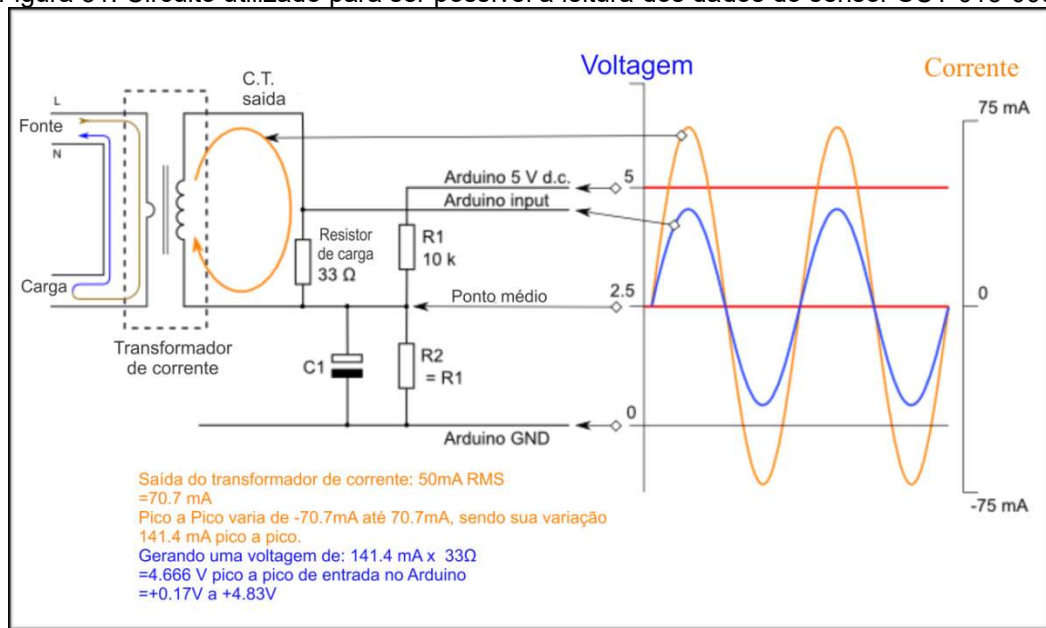
Como 35  $\Omega$  não é um valor de resistência comum, e os valores mais comuns próximos a este são 39  $\Omega$  e 33  $\Omega$ , foi escolhido fazer uso do menor valor para que a corrente de carga máxima não crie uma tensão maior do que o AREF. Aqui estão os mesmos cálculos acima em uma forma mais compacta:

$$\text{Resistência de carga (ohms)} = (\text{AREF} * \text{N}^\circ \text{ de voltas do CT}) / (2\sqrt{2} * \text{corrente de pico primária}).$$

$$\text{Resistor de carga} = (2,5\text{ V} * 2000) / 2\sqrt{2} * 141,4 = 35,4\ \Omega.$$

Se você fosse conectar um dos fios do CT ao solo e medir a tensão do segundo fio, em relação ao solo, a tensão variaria de positivo para negativo em relação à terra. No entanto, as entradas analógicas Arduino requerem uma tensão positiva. Conectando o cabo do CT conectado à terra, a uma fonte com a metade da tensão de alimentação, a tensão de saída do CT vai agora oscilar acima e abaixo de 2,5 V permanecendo assim positivo, como pode ser verificado na Figura 31.

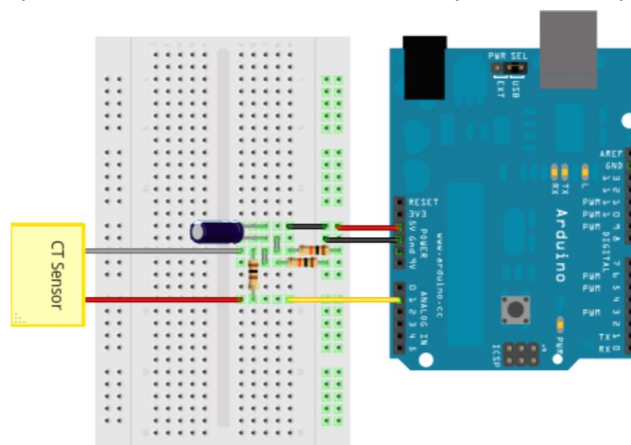
Figura 31: Circuito utilizado para ser possível a leitura dos dados do sensor SCT 013-000.



Fonte: Adaptado de OpenEnergyMonitor (2016c).

Os resistores R1 e R2 no diagrama de circuito acima são um divisor de tensão que fornece a fonte de 2,5 V. O capacitor C1 tem uma baixa reatância - algumas centenas de ohms - e fornece um caminho para a corrente alternada para contornar o resistor. Foi utilizado um capacitor com o valor de 10  $\mu\text{F}$ . Em uma protoboard (placa de ensaio para montagem de circuitos), o circuito ficará como na Figura 32:

Figura 32: Circuito resultante para o sensor de corrente, que produz um sinal proporcional à corrente da rede, que converte o sinal em um formato que o Arduino pode usar.



Fonte: (OPENENERGYMONITOR, 2016d).

Neste trabalho foi construído um monitor de energia simples que mede a corrente, mas no servidor é usado um valor fixo, assumindo a tensão 220V (tensão utilizada em

boa parte dos municípios do Brasil) e calcula a potência aparente nos gráficos do sistema. Embora não tão preciso quanto um monitor que mede a tensão e a corrente, este é um método comumente utilizado em monitores de energia de uma residência inteira, estando comercialmente disponível por razões de simplicidade e custo (OPENENERGYMONITOR, 2016d).

Feitos os testes de leitura com o monitor de corrente a partir de uma protoboard, verificou-se um ruído de leitura no circuito, mais informações na seção 5.3. Para resolver esse problema, necessitou-se a construção de um circuito permanente para que o projeto passasse a ser mais compacto e esse ruído diminuísse.

#### **4.1.3 Construção da PCB Do Circuito Para a Leitura da Corrente**

As placas de circuito impresso, ou PCB, são feitas a partir de uma base não condutiva, como fenolite ou fibra de vidro que são cobertas por uma camada condutora, geralmente cobre. Para a construção desse tipo, a placa passa por um processo subtrativo, onde toda a superfície condutiva, exceto a superfície que fará as conexões dos componentes eletrônicos é retirado. Para que isso ocorra, o projeto da placa é desenhado sobre a camada de cobre usando uma substância resistente, como uma caneta especial ou impressão a laser em papel fotográfico (glossy) na superfície que deve ser preservada no processo de corrosão. Após esse processo, a placa já com a máscara aplicada, é mergulhada uma substância corrosiva, conforme a Figura 33, que subtrairá o cobre não protegido e deixará apenas as áreas cobertas, que não são afetadas, deixando as trilhas desenhadas na placa. Por fim, basta remover a camada de proteção do cobre, furar a placa para fazer a soldagem dos componentes e usar (LEMOS, 2016).

Seguindo os passos referidos acima, foram necessários os seguintes materiais:

- Placa “virgem” de fenolite;
- Percloroeto de Ferro;
- Perfurador de placa;
- Recipiente, que não seja metálico, para colocar a solução corrosiva de percloroeto de ferro;
- Diagrama do circuito da placa;

- Esponja de aço.

A placa de fenolite e a solução de percloroato de ferro foram compradas em uma loja de eletrônica e pode ser facilmente encontrada na internet também, o preço gasto no percloroato e do fenolite não passou de 20 reais.

Figura 33: Placa de fenolite e percloroato de ferro em uma vasilha de plástico.

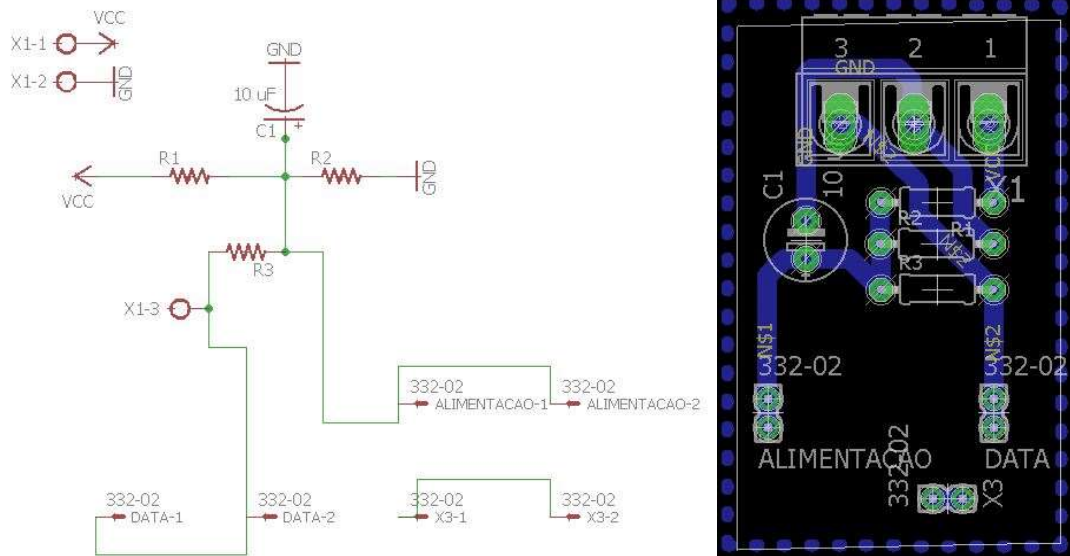


Fonte: O Autor.

#### 4.3.1 Software para a Confeção do Design da Placa

O software utilizado para esse fim foi o Eagle 8.2.1, software com licença de download e uso para projetos não comerciais, que é muito utilizado para o desenho de placas PCB, ou seja, placa de circuito impresso. Nas Figuras 34a e 34b temos o projeto do schematic (Esquema de representação do circuito) e o design da placa feita no Eagle.

Figura 34: a: Schematic do circuito do leitor de corrente. b: Design da placa a ser impressa.



Fonte: O Autor.

#### 4.3.2 Impressão e Montagem da Placa

Após a construção do esquema e do design, foi feita a impressão do design da placa (Figura 35a) em uma folha de papel de foto em uma impressora a Laser . Feita a impressão, os pedaços da folha destinadas a placa foram cortados e colados na placa de fenolite (Figura 35b), que foi mergulhada por alguns minutos na solução de percloroeto de ferro, fazendo assim a corrosão das áreas que não tinham tinta, como pode ser visto na Figura 35c.

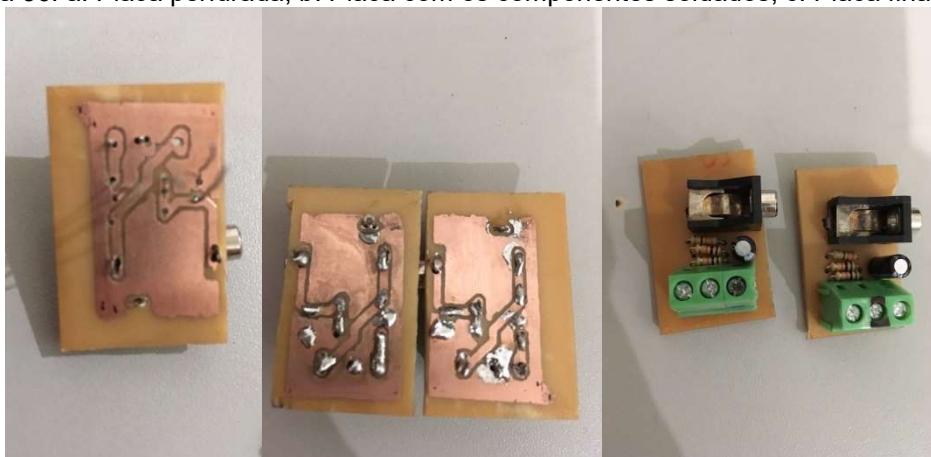
Figura 35: a: folha com o design da placa; b: Design impresso e colado à placa de fenolite; c: placa de fenolite após a corrosão à esquerda antes de ser lavada e depois de ser lavada à direita.



Fonte: O Autor

Feito isso, foram realizados furos na placa para a colocação dos componentes, e em seguida foi feita a soldagem de tais chegando ao produto final como pode ser visto repectivamente nas Figuras 36a, 36b e 36c.

Figura 36: a: Placa perfurada; b: Placa com os componentes soldados; c: Placa finalizada.



Fonte: O Autor.

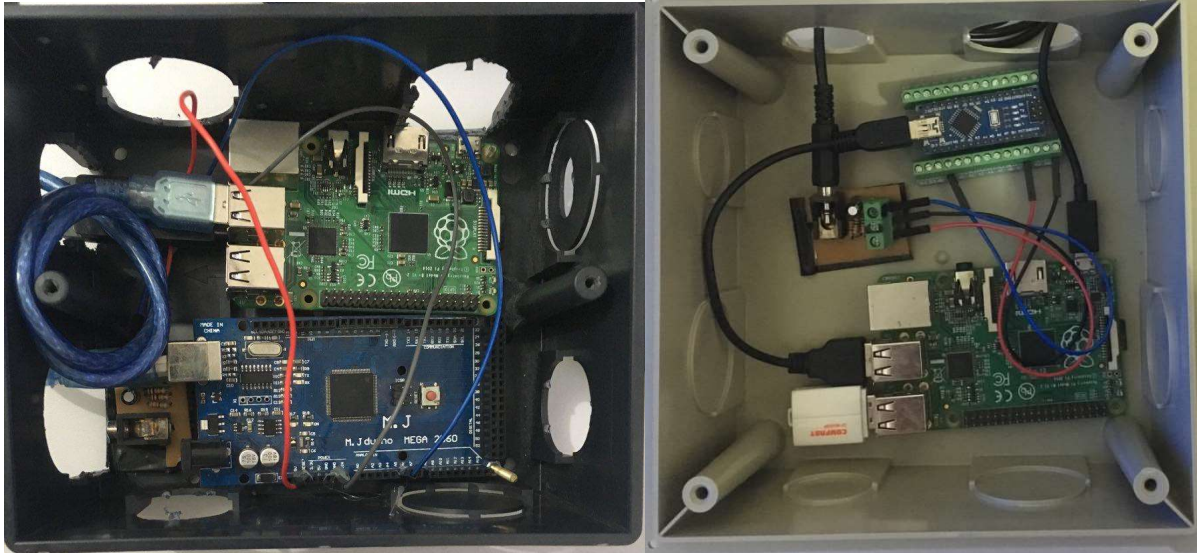
#### 4.1.4 Montagem do protótipo

Com os hardwares na seção 4.1 foi construído dois protótipos do sistema para atuarem em locais diferentes, um em uma residência com o Arduino Mega e o outro no laboratório LInDALVA(Laboratório de Inteligência em Dispositivos de Arquitetura Livre e Veículos Autônomos) com o Arduino Nano. O protótipo que ficava no laboratório servia para testes. Na construção dos dois protótipos foi utilizando uma



caixa de passar fios de 15x15 cm que pode ser encontrada em qualquer loja de materiais de construção.

Figura 37: a: Protótipo do sistema colocado na residência. b: Protótipo do sistema colocado no laboratório.

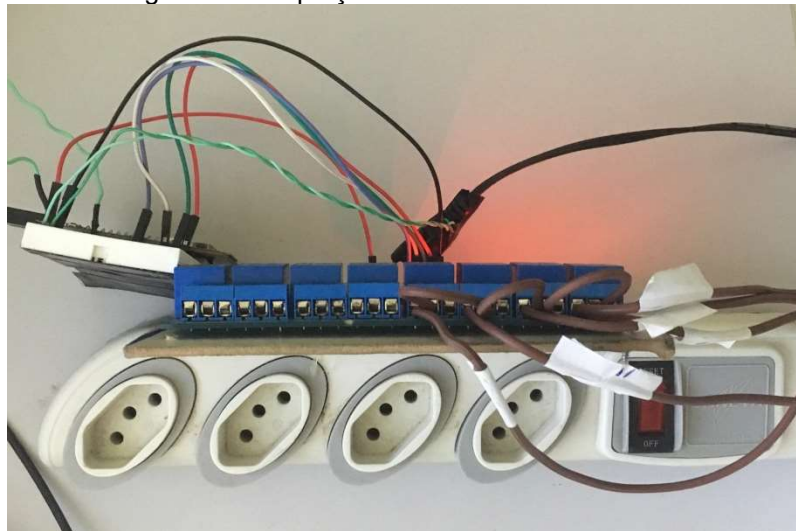


Fonte: O Autor.

Como pode ser observado nas Figuras 37a e 37b, cada um dos dois sistemas tem: 1 Raspberry Pi B+, 1 Arduino, 1 Adaptador WiFi USB, uma placa PCB, 3 jumpers que conectam o Arduino à placa PCB, 1 cabo USB para a conexão entre o Raspberry Pi e o Arduino.

Para a montagem do protótipo de controle, ligar/desligar dispositivos, necessitou-se fazer a adaptação dos relés a um filtro de linha, como ilustrado na Figura 38. Esse filtro de linha foi obtido como sucata e modificado para que cada entrada tomada correspondesse a um relé.

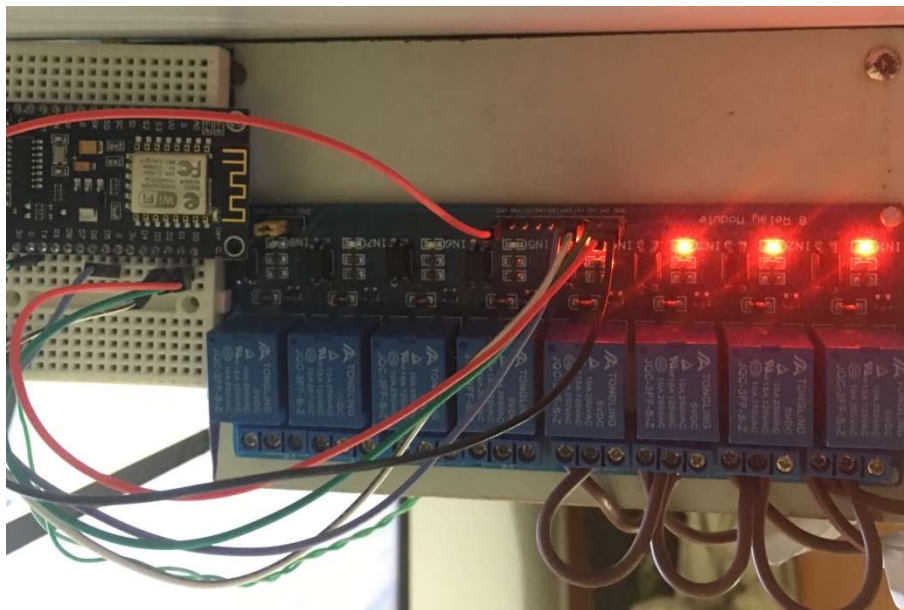
Figura 38: Adaptação do filtro de linha aos relés



Fonte: O Autor

Como o filtro de linha tem apenas 4 tomadas, foram utilizados apenas 4 dos 8 canais do módulo de relés. Após essa adaptação, o NodeMCU foi conectado aos relés por meio de jumpers como observado na Figura 39.

Figura 39: NodeMCU conectado ao módulo de relés



Fonte: O Autor

Quanto a alimentação, o NodeMCU e o Relé são alimentados separadamente por uma fonte de 5V.

#### 4.1.4 Instalação do protótipo do Leitor de Consumo

A título de ilustração, a seguir só será detalhado a instalação do protótipo na residência. A caixa de distribuição da residência, mostrada na Figura 40, conta com 6 disjuntores: Geral, Chuveiro 1, Chuveiro 2, Cozinha, Tomadas, Iluminação.

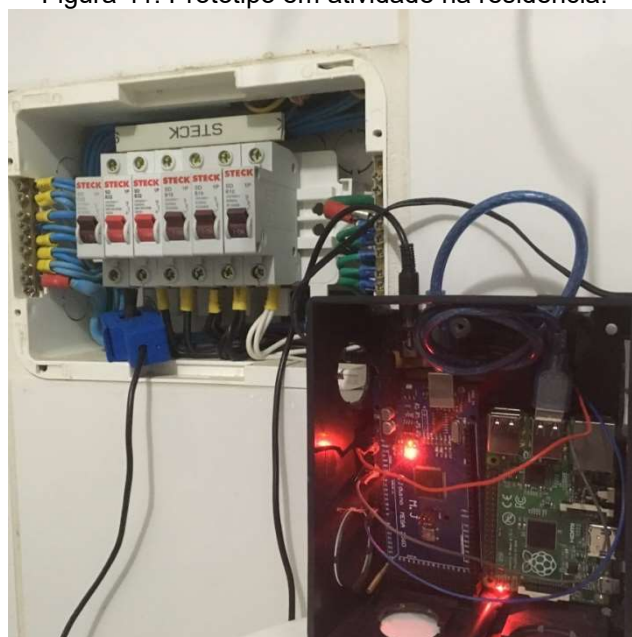
Figura 40: Caixa de disjuntores da residência.



Fonte: O Autor.

Dessa forma, para verificar a corrente que está sendo consumida na residência foi necessário colocar o sensor SCT013-00 na chave Geral, que é a fase monofásica. A conexão do sensor com a fase do disjuntor é mostrada na Figura 41.

Figura 41: Protótipo em atividade na residência.



Fonte: O Autor.

Para finalizar a instalação, foi necessário fixar a caixa do protótipo na parede e colocar as devidas tampas do protótipo e da caixa de distribuição, como pode ser observado na Figura 42.

Figura 42: Protótipo instalado na residência.



Fonte: O Autor.

## 4.2 Software

Para a utilização do Raspberry Pi como servidor web, foi necessária a preparação do ambiente com instalação de programas e a configuração de rotinas. Para um melhor entendimento, esse capítulo foi dividido entre ferramentas para o desenvolvimento da interface, chamadas de Front-End, e das ferramentas que trabalham por trás da interface, chamadas de Back-End.

### 4.2.1 Ferramentas Utilizadas no Front-End

Para o desenvolvimento do Front-End do projeto, foi utilizado o tema SB Admin 2, baseado no Bootstrap v3.3.7, que contém uma vasta documentação de tabelas,

gráficos, ícones, botões, fontes e ótimos plug-ins jQuery, aumentando a sua funcionalidade e capacidade de interação entre o navegador e o usuário. Apesar de ter incluído em sua documentação dois plug-ins jQuery: Flot Charts e Morris.js, foi utilizado o API Chart.js para a plotagem dos gráficos, por conter 8 tipos de gráficos, ter uma ótima renderização e performance no HTML 5 utilizando o elemento Canvas, que juntamente com o JavaScript permite a criação de uma interface para o desenho de pixels na tela.

Para o gerenciamento de usuários, foi utilizada um plugin que utiliza o jQuery, chamado de DataTables, que se trata de uma ferramenta para a organização de dados em tabelas em HTML. Esse plugin foi utilizado por facilitar o desenvolvimento a vida do desenvolvedor permitindo organizar e apresentar dados no formato de tabela, tais quais, pesquisa, paginação, ordenação, filtro, etc.

Ainda seguindo a premissa do Requisito não funcional 07, o tema SB Admin, o plugin DataTables e o API Chartjs são softwares com a licença open-source MIT, a qual permite que o código fonte desses softwares possam ser baixados, editados, modificados, distribuídos e até vendidos, sendo exigido apenas o aviso de licença e direitos autorais.

#### **4.2.2 Ferramentas Utilizadas no Back-End**

O Back-End nada mais é que o software do lado do servidor, a base utilizada para o servidor web desse sistema é feita a partir do pacote LAMP (Linux, Apache, MySQL e PHP ) no Raspberry Pi, que é uma combinação de softwares livre e de código aberto para a construção de aplicações de Back-End, unindo alta disponibilidade, alto desempenho, segurança e compatibilidade com o Raspberry Pi.

O servidor web Apache versão 2.4.10 ficou responsável por processar e atender as solicitações HTTP feitas pelos usuários, o MySQL 5.5.54 ficou responsável pelo gerenciamento do banco de dados. A Linguagem utilizada para fazer os scripts de acesso ao banco de dados foi o PHP.

Além do Servidor Web Apache, foi necessário a implementação de outros dois servidores, um utilizando NodeJs, por conta da necessidade do desenvolvimento de

uma página para a plotagem dos dados recebidos em tempo real, e um servidor Python para fazer a comunicação com o Broker via MQTT. Um resumo de atividades será mostrado mais abaixo sobre as tarefas desses dois servidores:

O servidor NodeJs:

- Faz a conexão com o banco de dados;
- Cria uma web app juntamente com o framework Express;
- Permite estabelecer uma conexão em tempo real utilizando o Socket.io, que é uma biblioteca que oferece uma API de JavaScript simples baseada em eventos que permitem a comunicação entre o servidor e o cliente sem esforço e em tempo real;
- Estabelece a conexão com o Arduino através da biblioteca mySerial, definindo a porta serial que o Arduino está conectado, e o seu baud rate (taxa de transmissão de dados);
- Salva no banco de dados todos os dados que chegam via serial;
- Abre uma conexão HTTP na porta 8080 do servidor.

Resumindo: Conecta-se ao Arduino, salva os dados recebidos via serial e envia esses dados para a porta 8080 que é escutada pela página de interface gráfica do consumo instantâneo, dessa forma, o usuário tem acesso à plotagem do gráfico em tempo real.

O servidor Python é responsável por:

- Ser um servidor web responsável pela renderização da página de controle via framework web flask, hospedada na porta 8181;
- Criar um Cliente MQTT para receber o status das GPIOs do NodeMCU;
- Conectar-se ao Broker Mqtt, publicar e assinar aos tópicos específicos de controle dos dispositivos.

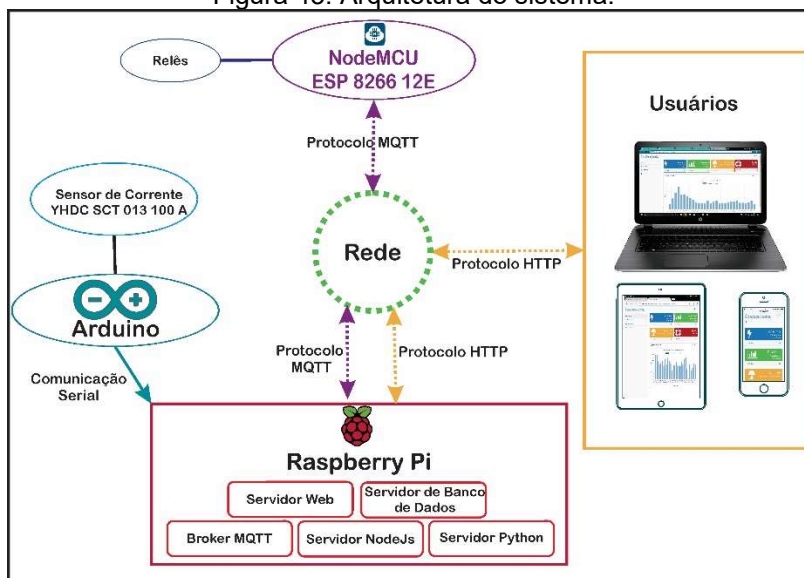
De maneira resumida, a página de controle (ligar/desligar dispositivos) é renderizada por esse servidor, pois, pelo fato de necessitar do conhecimento do status das GPIO's do NodeMCU, foi preciso criar um cliente MQTT nesse servidor. Dessa forma, é possível que ele verifique quais dispositivos estão ligados ou desligados, e consegue fazer o retorno desses dados de status à página de controle.

### 4.3 Arquitetura do Sistema

Utilizando a abordagem de arquitetura cliente-servidor, o Back-End foi desenvolvido para disponibilizar o acesso aos recursos do sistema. Na sua conexão com a rede, o Servidor Web no Raspberry Pi fornece os serviços do sistema por protocolo HTTP numa página web para o Cliente, que é usuário do sistema. O usuário faz a requisição de uma página do sistema acessando o site por um navegador. Chegando essa requisição no servidor web, ele retorna a página que foi pedida pelo usuário. Na aba de consumo instantâneo, a página fica escutando a porta 8080, que é a porta que o servidor NodeJs está, para a plotagem do gráfico em tempo real. A arquitetura completa do sistema está ilustrada na Figura 43.

Para um usuário ligar/desligar algum dispositivo, ele deverá acessar a página de controle disponível na porta 8181 onde está localizado o servidor Python, e mudar o status do dispositivo. Essa comunicação entre o usuário e o servidor é feita via protocolo HTTP, mas para que o Raspberry Pi se comunique com o NodeMCU é passada via protocolo MQTT a informação modificada. Dessa forma, o NodeMCU atua sobre o relé e retorna para o servidor a porta modificada.

Figura 43: Arquitetura do sistema.



Fonte: O Autor.

Já a comunicação feita entre o Arduino e o Raspberry Pi é feita via comunicação serial via USB. A cada segundo o Arduino lê o dado do sensor e envia os dados para o Raspberry Pi, que por sua vez, salva no banco de dados por meio do servidor NodeJs.

## **5. CENÁRIOS, TESTES, REPAROS E RESULTADOS.**

Os cenários foram criados para validar o projeto e verificar se o mesmo atende às necessidades estabelecidas inicialmente. Cada um deles foi testado com o objetivo de verificar a resposta produzida pelo sistema.

### **5.1 Falta de alimentação e reinício automático do sistema**

Por se tratar de uma placa que de fábrica não vem com a bateria integrada, o Raspberry Pi desliga caso a fonte de alimentação seja cortada. Dessa forma, em caso de falta de energia em uma casa o sistema de leitura e salvamento de consumo sofreria uma pausa, sendo necessária a intervenção do administrador ou usuário para que o servidor NodeJs voltasse a funcionar. Para resolver essa questão foi criada uma rotina via cron, que se trata de um serviço do Linux que é carregado durante o processo de boot do sistema, e disponibiliza o agendamento de execução de comandos e processos. Fez-se necessário ainda, utilizar o módulo forever que faz com que scripts continuem em execução mesmo que tenha ocorrido algum erro. Após uma pausa o script retoma sua execução normalmente. O comando abaixo foi criado com o crontab para resolver essas questões:

```
@reboot /usr/bin/sudo -u pi -H /usr/local/bin/forever start /var/www/energia/servidor/server.js
```

Esse script faz com que toda vez que a placa inicie no processo de boot do sistema, seja chamado o processo forever dando início ao servidor server.js, que é o servidor que faz a leitura e o salvamento dos dados do consumo energético. Após a inserção dessa rotina, o sistema de leitura funcionou normalmente, levando em torno de dois minutos para voltar a sua atividade a cada reinício.

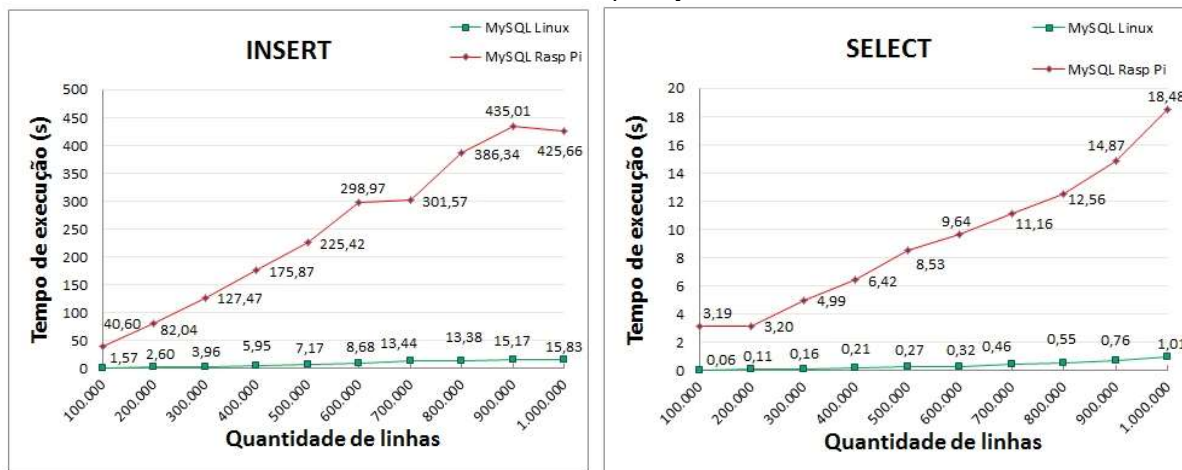
### **5.2 Demora na pesquisa e inserção do banco de dados – Criação de rotina de inserção a cada hora**

Apesar de ser uma placa muito boa, o Raspberry Pi não conta com um recurso computacional muito elevado para a consulta de banco de dados grandes. Alguns testes foram feitos por Pichiliani (2016) utilizando o Raspberry Pi B e um computador



com um processador bem mais robusto. A Figura 44 mostra alguns resultados dos testes feitos.

Figura 44: Tempo de execução do comando insert e select em um banco MySql em um computador e em um Raspberry Pi.



Fonte: (PICHILIANI, 2016)

Durante o processo inicial das consultas ao banco de dados para gerar os relatórios na página web, foi definido que a consulta seria realizada na tabela “medidas” que continha os dados de segundo em segundo, com cerca de 86.400 registros inseridos na tabela, como foi descrito na Seção da 3.2.6. Por ser uma tabela com muitos dados, a plotagem do gráfico de histórico de leituras era gerado com um atraso de cerca de um minuto para retornar a consulta para a página web. Verificando isso, fez-se necessário a criação de uma nova tabela no banco de dados, chamada “medidasHora”, que granulava os dados da tabela “medidas” em horas, fazendo a soma de todos os dados de potência na hora referida, dividindo esse valor por 3600000, e salvando o consumo na tabela “medidasHora” em kwh. Para fazer isso automaticamente foi preciso utilizar novamente o cron, agora com um comando que fazia com que a cada hora, esse serviço execute um script php que faz a consulta à base de dados “medidas”, e insira na tabela “medidasHora” os respectivos valores de potencia total consumida na hora, dia, mês, e ano. Dessa forma a consulta ficou mais rápida, pois em um mês de trinta dias, ao invés de 2.592.000 registros, tem-se apenas 720, retornando assim o gráfico instantaneamente para o usuário.

### 5.3 Erro de leitura do leitor de consumo utilizando protoboard – Construção de uma PCB

Questões comuns levantadas a quem utiliza o sensor SCT013-000 são do tipo:

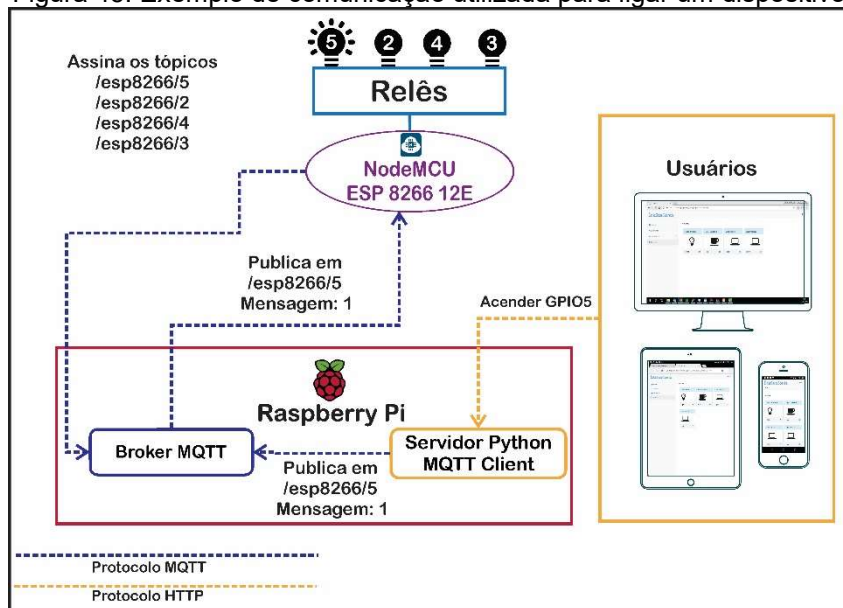
“por que há leitura de x Ampères quando não há nada conectado a energia?”. Como pode ser visto em OpenEnergyMonitor(2017), o conversor analógico-digital (ADC) no chip Arduino não é perfeito. De maneira resumida, é possível obter alguma leitura errônea quando não há corrente passando pelos fios. Além do conversor, outro problema que pode ser encontrado é o de ruídos digitais gerados pelo próprio Arduino, ou ruído externo captado pelas protoboards ou a fiação. Nesse caso, optou-se pela construção de um circuito impresso para diminuir o erro nas leituras feitas em uma protoboard, além de diminuir o espaço usado para a construção do circuito, como descrito na sessão seção 4.1.3.

#### **5.4 Acúmulo de funções ao Arduino - NodeMCU e o protocolo MQTT**

No projeto inicial, o Arduino acumulava as funções de ler o sensor e atuar sobre o sistema de controle, mas pelo fato dele estar “amarrado” geograficamente ao Raspberry Pi fez com que se optasse pela compra de um outro microcontrolador, com uma interface de comunicação WiFi, que no caso foi NodeMCU, para controlar os relés. Dessa forma, teve que ser incluído um novo protocolo de comunicação ao sistema, o MQTT, para fazer a comunicação entre o Raspberry Pi e o NodeMCU. Para tal, necessitou-se a instalação do pacote Mosquitto no Raspberry Pi, para que ele funcionasse como Broker.

Seguindo o modelo da arquitetura publish/subscribe, explicada na seção 2.6.3.1, o Raspberry Pi é o Broker e um servidor web Python cliente MQTT, além disso, temos também o NodeMCU como cliente MQTT. Quando um usuário deseja ligar um dispositivo, ele acessa a aba de controle e clica sobre o ícone do dispositivo. Ao clicar, ele manda uma requisição ao servidor web Python, que por ser um cliente MQTT, publica o valor “1” no tópico “esp8266” no sub tópico do dispositivo a ser ligado, no caso a lâmpada que está na GPIO5 do NodeMCU tem o valor 5, ou seja, manda uma requisição na forma “esp8266/5/1” a partir daí, o Broker encaminha a mensagem para todos os clientes que estão inscritos nesse tópico, que no caso é o NodeMCU. O NodeMCU por sua vez, muda o valor da GPIO 5 de 0 para 1, fazendo com que o relé feche a chave e a lâmpada ligue. A Figura 45 mostra este exemplo citado acima.

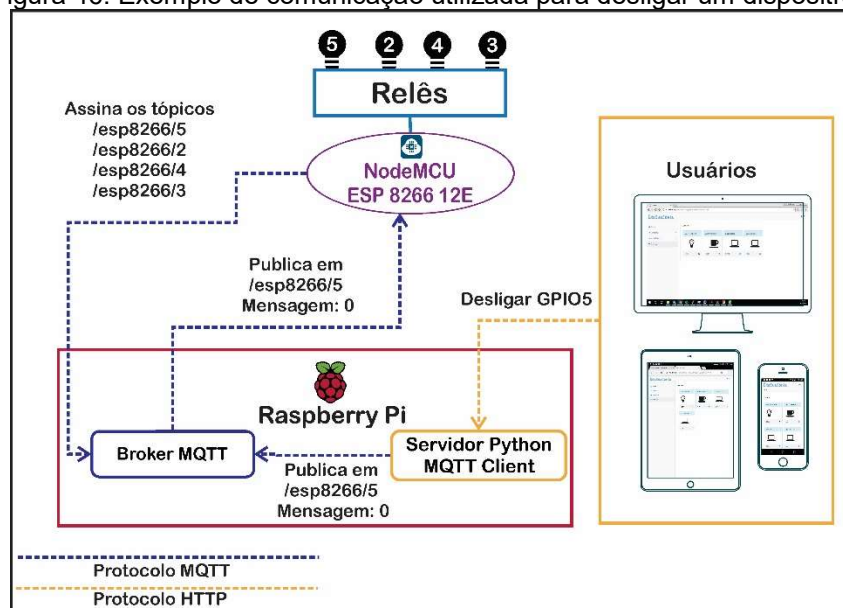
Figura 45: Exemplo de comunicação utilizada para ligar um dispositivo.



Fonte: O Autor.

Caso o usuário queira desligar o mesmo dispositivo, ele clicará no ícone que informa o status do dispositivo (ligado ou desligado), e ao clicar ele manda novamente uma requisição ao servidor web Python, o qual publica o valor "0" no tópico "esp8266" no sub tópico 5, ou seja, manda uma requisição na forma "esp8266/5/0". A partir daí, o Broker encaminha a mensagem, e o NodeMCU por sua vez, muda o valor da GPIO 5 de 1 para 0, fazendo com que a o relé abra a chave e lâmpada desligue. A Figura 46 demonstra o exemplo da comunicação utilizada para desligar um dispositivo.

Figura 46: Exemplo de comunicação utilizada para desligar um dispositivo.

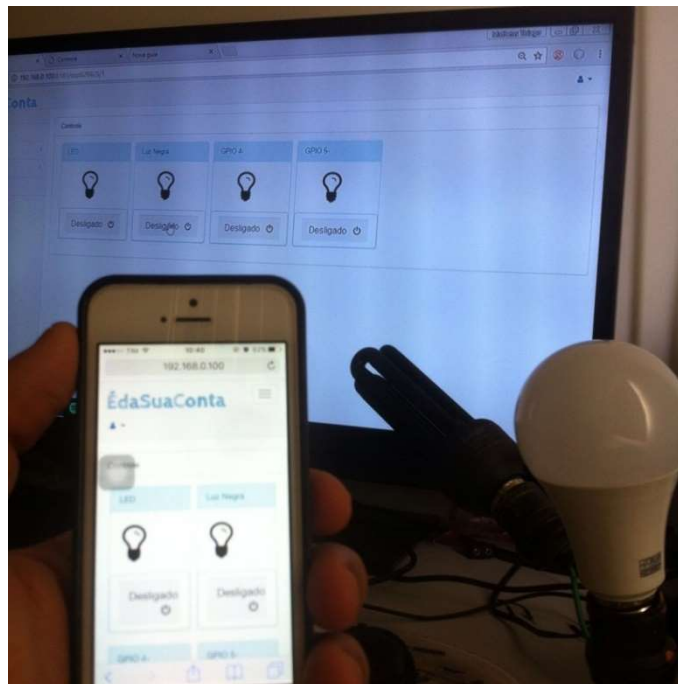


Fonte: O Autor

### 5.4.1 Teste de acionamento com duas lâmpadas

Com o protótipo de controle montado como na seção 4.1.4, foram conectados duas lâmpadas ao filtro de linha e foi acessada a página de controle (Ligar/Desligar dispositivos). A Figura 47 mostra o momento em que a página foi acessada por um celular e um computador, nela é possível verificar que os status dos dispositivos estavam corretos.

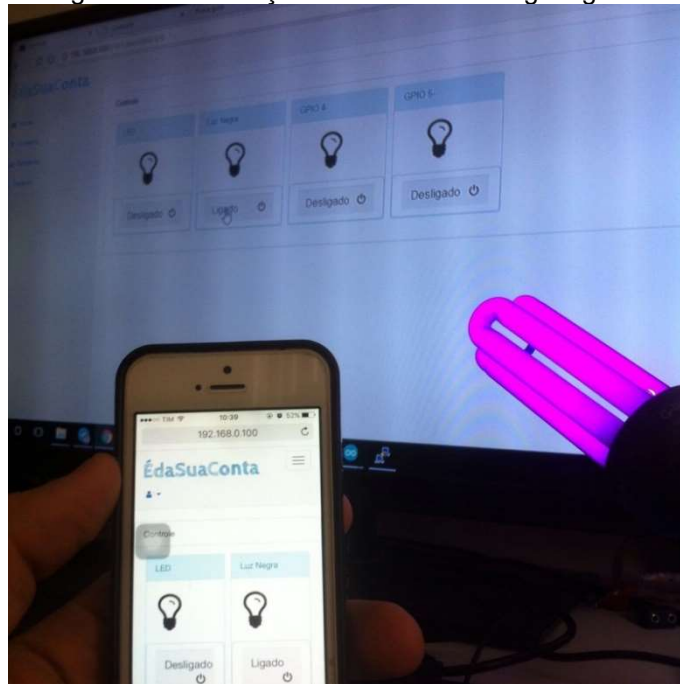
Figura 47: Verificação do status dos Relés



Fonte: O Autor.

Depois disso foi clicado sobre o segundo ícone que tinha o título de “Luz Negra”. Ao clicar, imediatamente a luz negra ligou-se e o status foi repassado corretamente ao computador e ao celular, como pode ser visto na Figura 48.

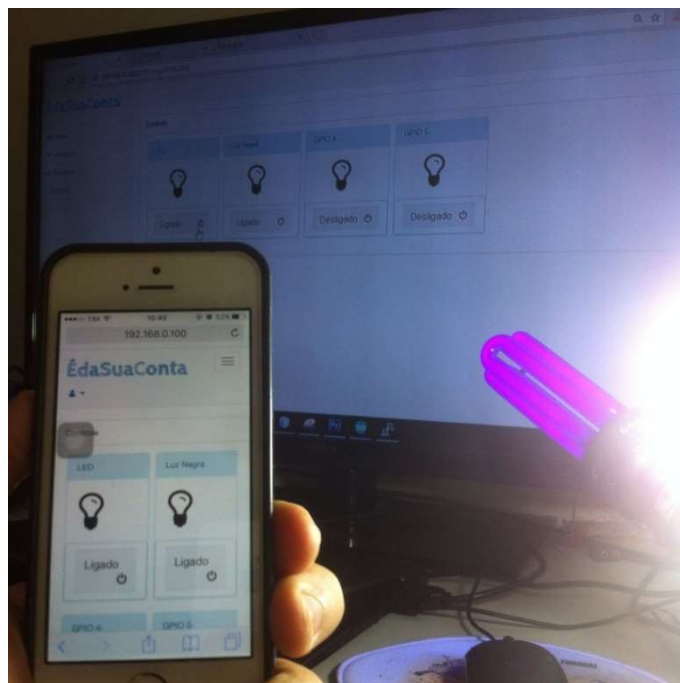
Figura 48: Verificação do status a luz negra ligada.



Fonte: O Autor.

A seguir, foi clicado sob o primeiro ícone “LED”, a resposta também foi imediata e o status também foi passado às páginas corretamente. Como pode ser visto na Figura 49, as duas lâmpadas foram acesas e o status foi passado corretamente às páginas web no computador e no celular.

Figura 49: Teste de acionamento de duas lâmpadas



Fonte: O Autor

Nos testes com a página de controle “ligar/desligar dispositivos”, o sistema funcionou como previsto, tendo uma resposta imediata de acionamento e os status das portas sendo passados corretamente a página de controle de dispositivos para o usuário.

## 5.5 Teste de Leitura

O ÉdaSuaConta foi colocado em testes de leituras em dois locais; em uma residência com o Arduino Mega e no laboratório LInDALVA(Laboratório de Inteligência em Dispositivos de Arquitetura Livre e Veículos Autônomos) com o Arduino Nano. Ambos cumpriram bem a sua função, não havendo diferenças na velocidade de consulta dos dados do sensor.

O sistema colocado no laboratório servia para testes em geral, seja ele de interface, de hardware ou de software. À medida que esses testes deram certo, as modificações eram repassadas ao sistema que estava na residência. Na residência o sistema foi colocado em atividade do dia 11/03 ao dia 14/05, dando um total de 65 dias. Para a verificação da veracidade do consumo, foram utilizadas as contas de energia dos meses de março e abril. A primeira conta de energia fez a leitura no intervalo dos dias 08/03 a 07/04, como pode ser visto na Figura 50.

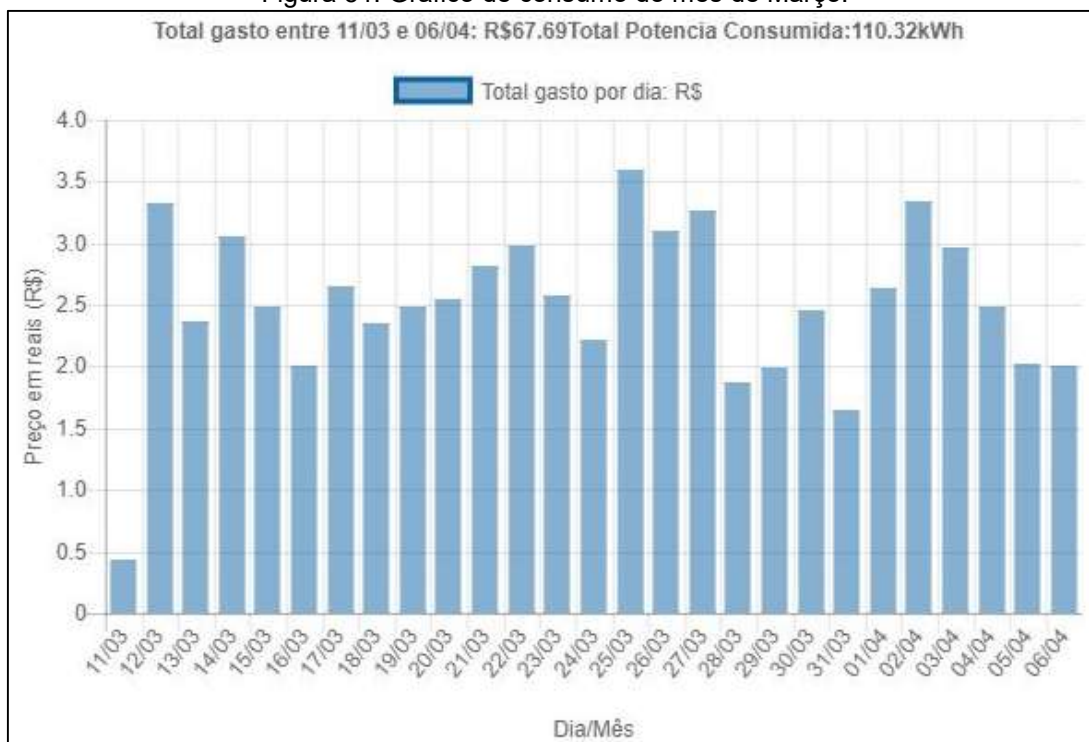
Figura 50: Nota fiscal do consumo do mês de março

DESCRIÇÃO DA NOTA FISCAL									
DESCRIÇÃO				QUANTIDADE	PREÇO	VALOR (R\$)			
Consumo Ativo(kWh)				114,00	0,61342401	69,93			
Acréscimo Bandeira AMARELA						2,57			
Acréscimo Bandeira VERMELHA						1,16			
Contribuição Iluminação Pública						4,48			
TOTAL DA FATURA						78,14			
DEMONSTRATIVO DE CONSUMO DESTA NOTA FISCAL									
NÚMERO DO MEDIDOR	TIPO DA FUNÇÃO	ANTERIOR		ATUAL		Nº DIAS	CONSTANTE	AJUSTE	CONSUMO kWh
		DATA	LEITURA	DATA	LEITURA				
	CAT	08/03/2017	4.929,00	07/04/2017	5.043,00	30	1,00000	0,00	114,00

Fonte: [servicos.coelba.com.br](http://servicos.coelba.com.br)

Como o sistema só foi instalado no dia 11/03 às 23 horas, não foi possível realizar as leituras corretas do dia 08/03 ao dia 11/03. Ainda assim, o resultado foi muito próximo à leitura presente na conta de energia. A Figura 50 informa que o consumo em kWh foi de 114 kWh em todo o mês, já o gráfico de consumo, retirado da página de relatórios do sistema, mostrado na Figura 51, informa que esse valor foi de 110.32 kWh.

Figura 51: Gráfico de consumo do mês de Março.



Fonte: O Autor.

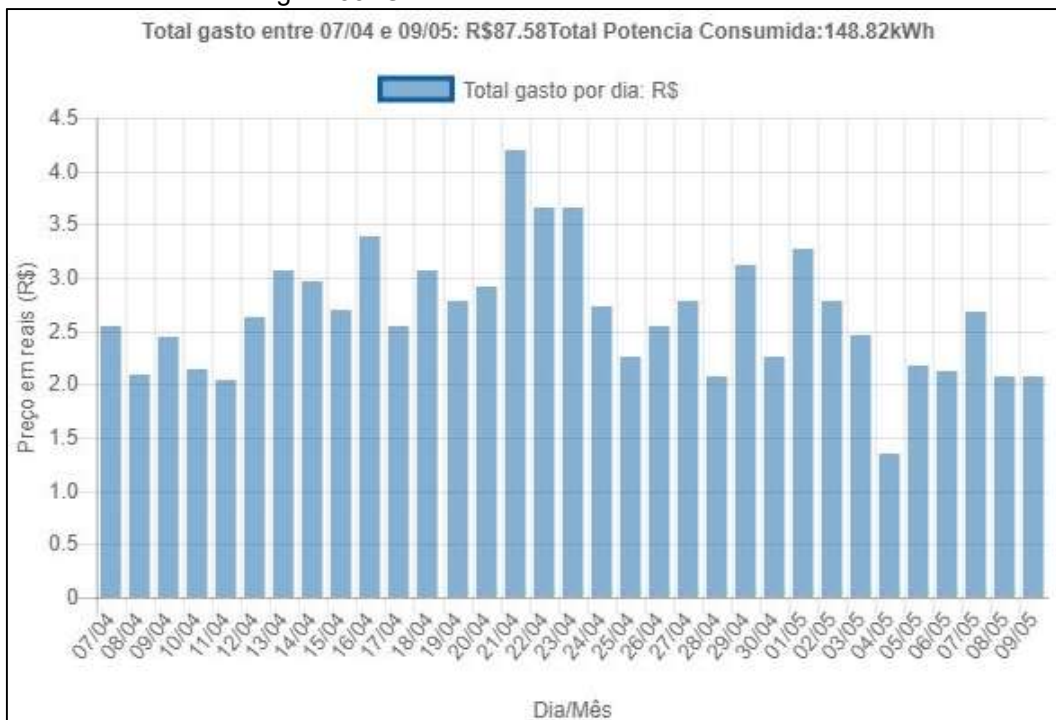
No mês de Abril foi verificado o consumo durante todo o mês, podendo observar se o sistema estava condizente com o gasto mostrado na conta de energia. Como pode ser verificado ao comparar as Figuras 52 e 53, o valor do leitor de consumo e o valor apresentado na nota fiscal são bastante próximos, com menos de 1% de variação.

Figura 52: Nota fiscal do consumo do mês de Abril.

DESCRIÇÃO DA NOTA FISCAL									
DESCRIÇÃO				QUANTIDADE	PREÇO	VALOR (R\$)			
Consumo Ativo(kWh)				147,00	0,58847192	86,50			
Acréscimo Bandeira VERMELHA						6,66			
Contribuição Iluminação Pública						5,54			
<b>TOTAL DA FATURA</b>						<b>98,70</b>			
DEMONSTRATIVO DE CONSUMO DESTA NOTA FISCAL									
NÚMERO DO MEDIDOR	TIPO DA FUNÇÃO	ANTERIOR		ATUAL		Nº DIAS	CONSTANTE	AJUSTE	CONSUMO kWh
		DATA	LEITURA	DATA	LEITURA				
	CAT	07/04/2017	5.043,00	09/05/2017	5.190,00	32	1,00000	0,00	147,00

Fonte: servicos.coelba.com.br.

Figura 53: Gráfico de consumo do mês de Abril.



Fonte: O Autor.

Como pode ser visto nas comparações entre as Figuras 50, 51, 52 e 53, o sistema mostrou-se bastante eficaz quanto o sistema de leitura de consumo.

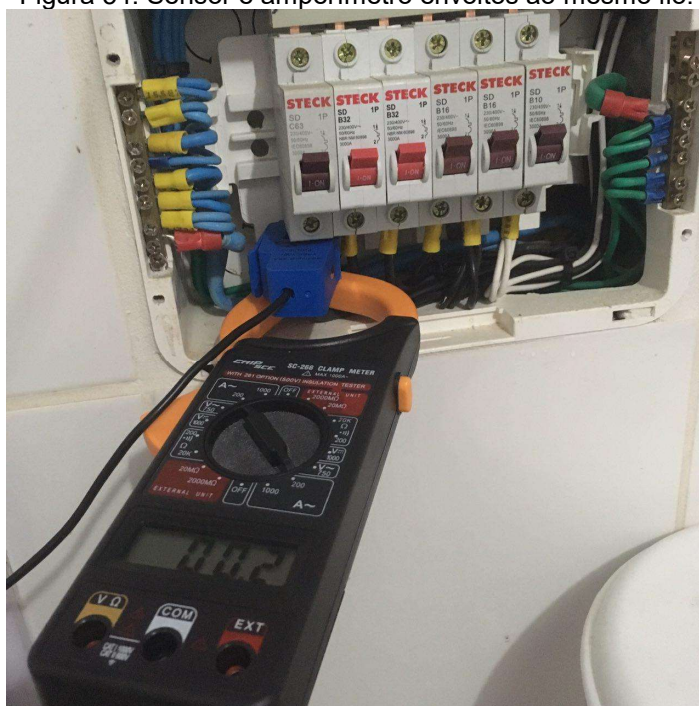


## 5.6 Testes utilizando monitor de consumo instantâneo

Para verificar a conformidade da velocidade da plotagem dos dados no gráfico e da verificação dos valores lidos pelo sensor, foram feitos testes de velocidade de retorno à página e testes utilizando um amperímetro para a comparação entre os dados lidos pelo sensor e o amperímetro.

O alicate amperímetro SC-266, assim como o sensor SCT013-000, é um equipamento não invasivo, dessa forma, basta envolver a garra dele ao fio para verificar a corrente que está passando. Nos testes 5.6.2 e 5.6.3 tanto o SCT013-000 quanto o SC-266 foram envoltos ao fio da fase monofásico da casa.

Figura 54: Sensor e amperímetro envoltos ao mesmo fio.



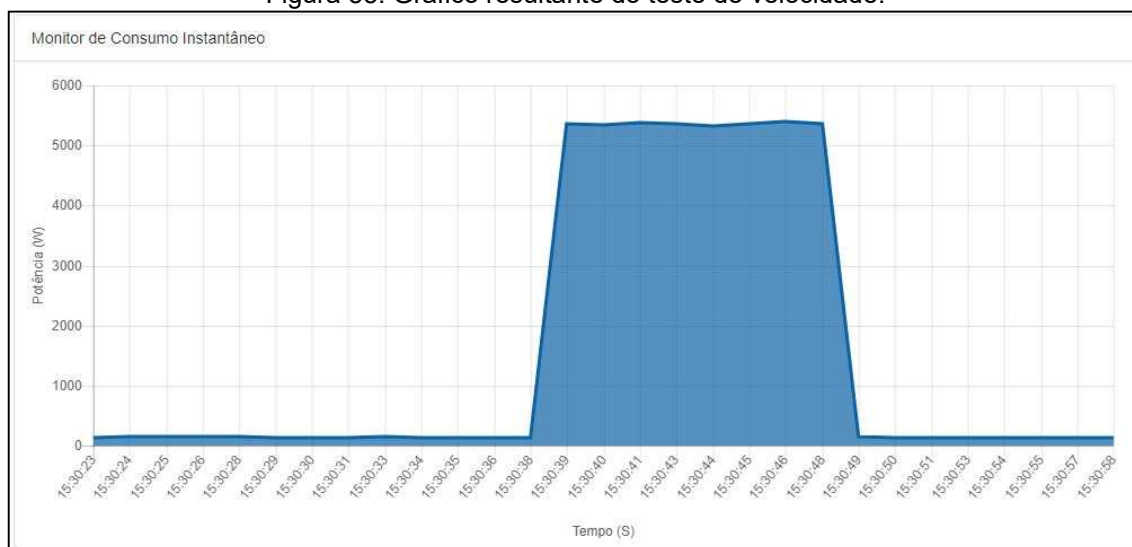
Fonte: O Autor.

Para ser possível a comparação dos dados do monitor de consumo com o amperímetro, necessitou-se a construção de uma nova página de consumo instantâneo, dessa vez mostrando ao usuário apenas os dados da corrente, ou seja, o dado cru lido pelo Arduino, sem a multiplicação pela tensão (220), que é a forma utilizada para o cálculo da potência aparente consumida (Potência Aparente=Tensão\*Corrente).

### 5.6.1 Teste de Velocidade

Para a verificação do tempo de resposta do monitor de consumo instantâneo, foi aberta a página de consumo e um chuveiro de 5500W foi ligado por 20 segundos, como ilustrado na Figura 55.

Figura 55: Gráfico resultante do teste de velocidade.



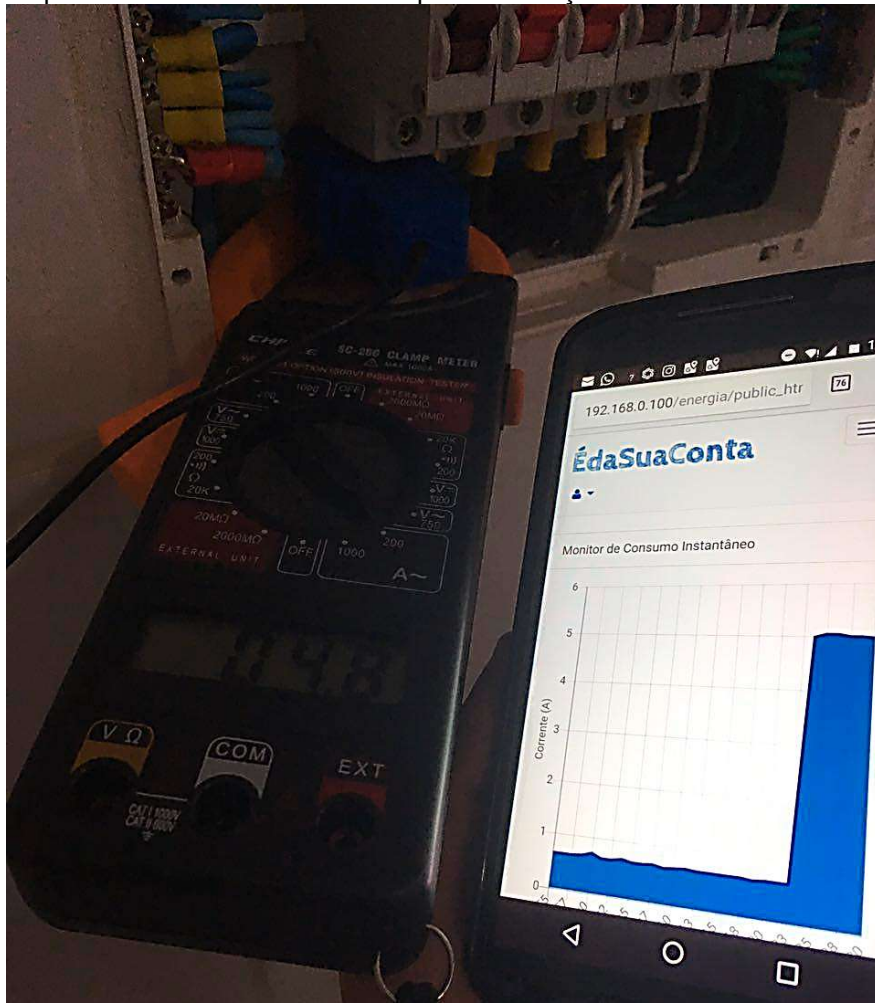
Fonte: O Autor.

A resposta dada pela página foi de um segundo para a plotagem de cada um dos dados que chegavam do Arduino para o gráfico, que é o tempo definido no sketch do Arduino para fazer a leitura do sensor. Sendo assim, eficaz quanto à velocidade de plotagem dos dados.

### 5.6.2 Teste com uma sanduicheira

Esse teste foi feito com uma sanduicheira que em sua etiqueta marcava 880W de potência, ou seja, aproximadamente 4A de corrente. Como o sensor e o amperímetro estavam conectados a fase geral da casa, há um consumo por conta de outros dispositivos ligados na residência, cerca de 0.8A. A figura 56 mostra o dado do amperímetro em 4.8A enquanto o gráfico do sistema converge também para um dado bem próximo a este, mostrando-se eficaz quanto à leitura.

Figura 56: Amperímetro e Celular lado a lado para verificação dos dados de leitura do consumo.

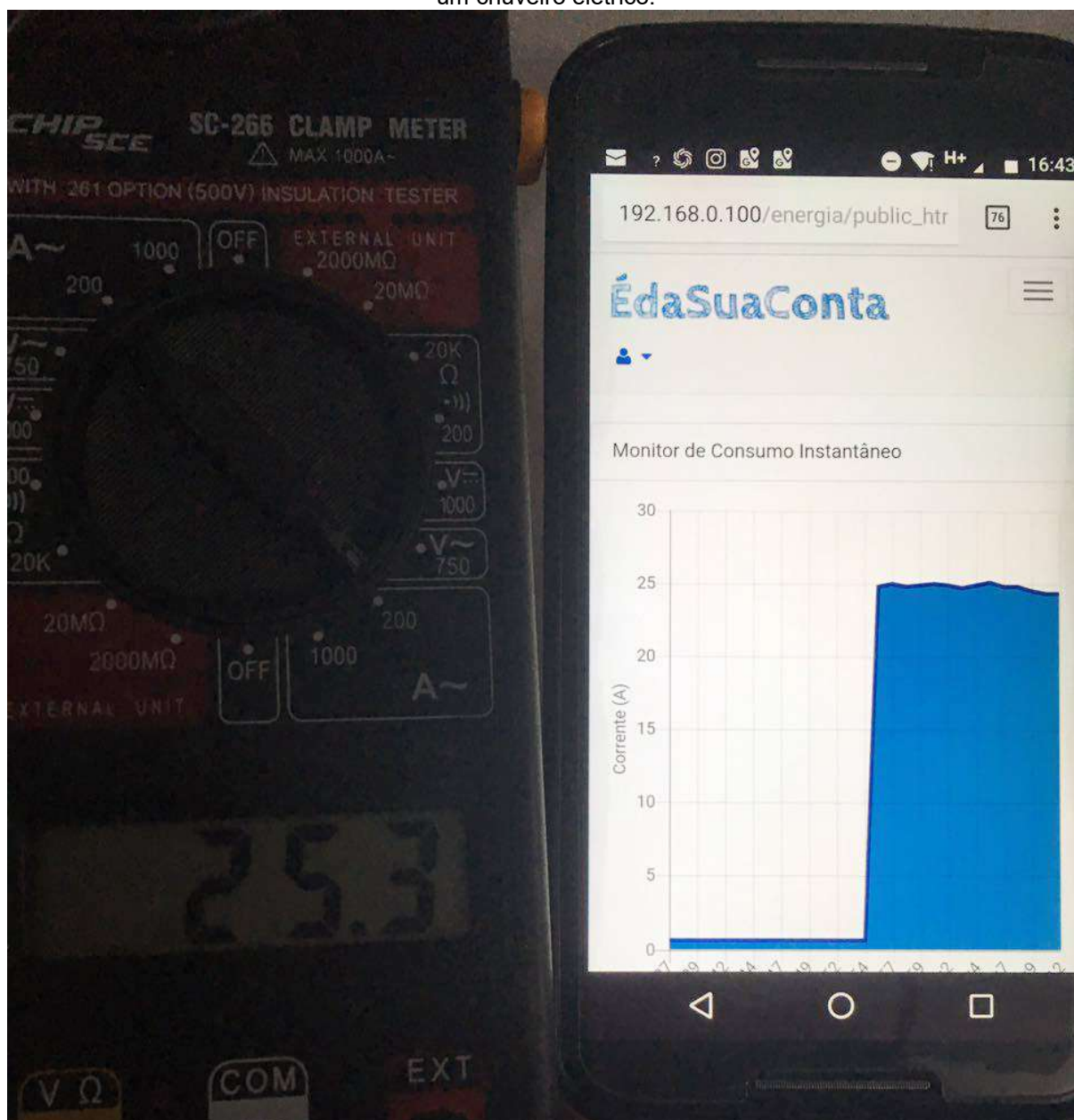


Fonte: O Autor.

### 5.6.3 Teste com um chuveiro elétrico

O teste com um chuveiro elétrico foi feito nos mesmos moldes citados acima. Como pode ser observado na Figura 57, o sistema teve uma leve diferença de cerca de 0.3A quando comparado a leitura do amperímetro.

Figura 57: Amperímetro e Celular lado a lado para verificação dos dados de leitura do consumo de um chuveiro elétrico.



Fonte: O Autor.

Apesar dessa variação entre o amperímetro e os dados do sistema, ainda assim o sistema mostra uma boa eficácia nas leituras.

#### 5.6.4 Teste sem equipamentos conectados

Como citado na Seção 5.3, quando o sensor SCT013-000 é utilizado juntamente com o Arduino pode se ter uma leitura mesmo sem nenhuma corrente passando no fio.

Dessa forma, desconectando o sensor do fio da fase obtiveram-se os valores de cerca de 0.4A.

## **5.7 Diferentes tamanhos de telas para acessar o sistema**

Pelo fato de se ter vários dispositivos com tamanho de telas diferentes no mercado como pode ser visto na pesquisa mostrada na seção 2.7. Fez necessária a construção de um Web Site com aplicações de técnicas de design responsivo para que o usuário possa acessá-lo de qualquer dispositivo que tenha um navegador e com qualquer sistema operacional.

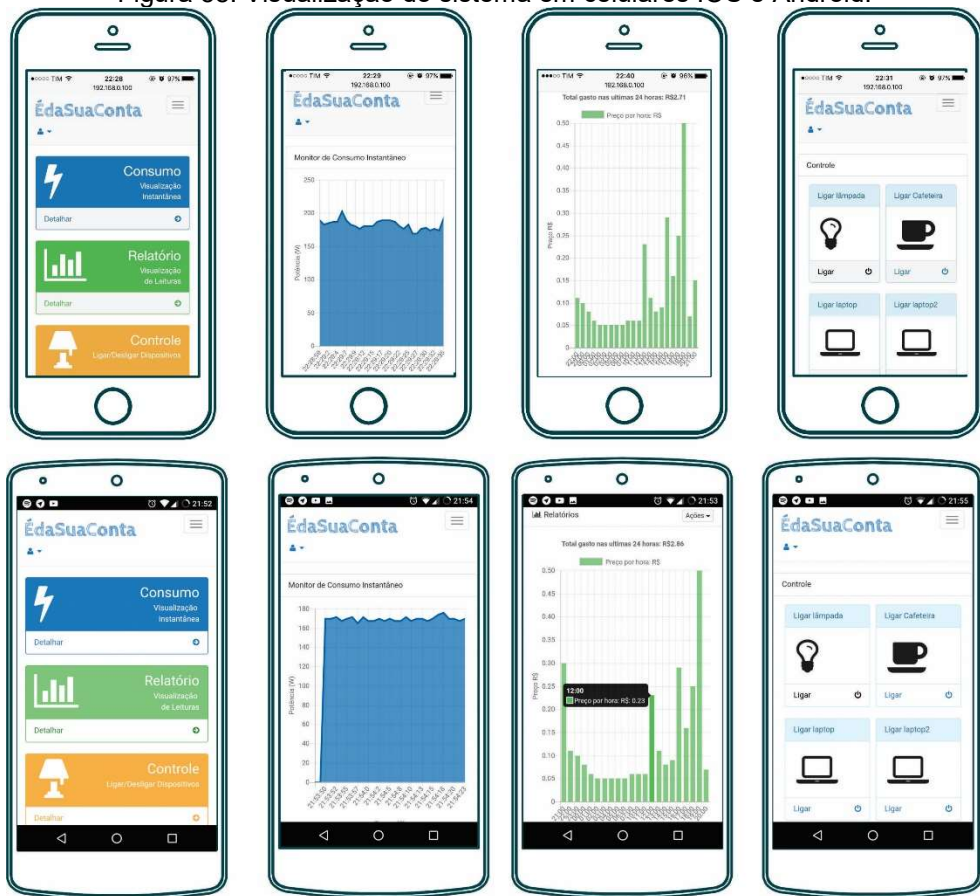
### **5.7.1 Interfaces do sistema**

Algumas imagens foram feitas a título de ilustrações reais das interfaces finais do sistema web. As Figuras 58, 59 e 60 mostram as seguintes telas:

- Tela inicial do sistema, mostrada logo após o usuário logar no mesmo;
- Tela de visualização do consumo instantâneo;
- Tela de visualização de relatórios;
- Tela para ligar/desligar dispositivos.

A Figura 58 mostra a visualização das telas em dispositivos extra pequenos, de largura máxima de tela de 768px com os sistemas IOS e Android nos navegadores Safari e Google Chrome.

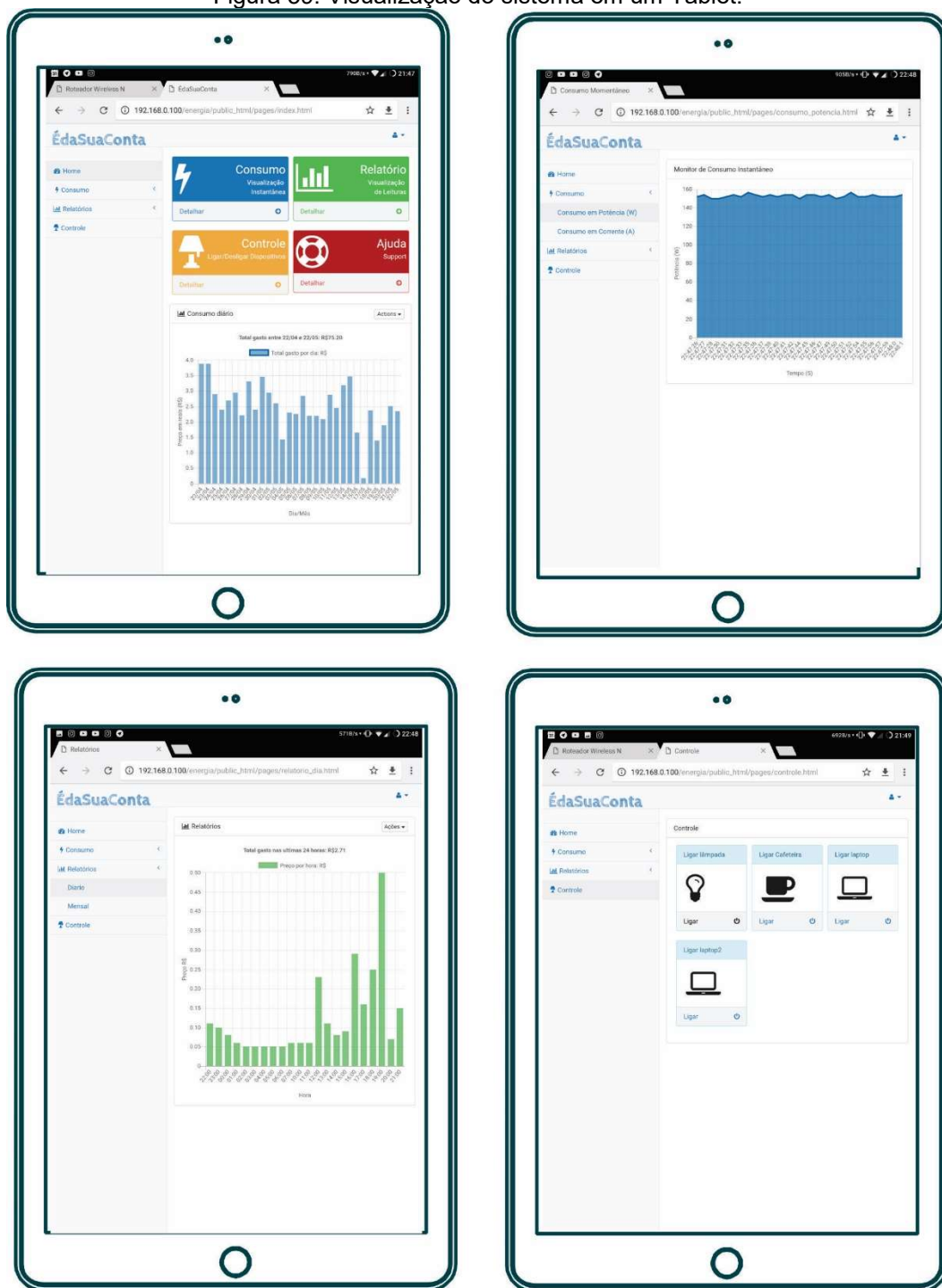
Figura 58: Visualização do sistema em celulares IOS e Android.



Fonte: O Autor.

A Figura 59 mostra a visualização das telas em dispositivos pequenos e médios, com largura de tela entre 768px e 1200px com o sistema Android no navegador Google Chrome.

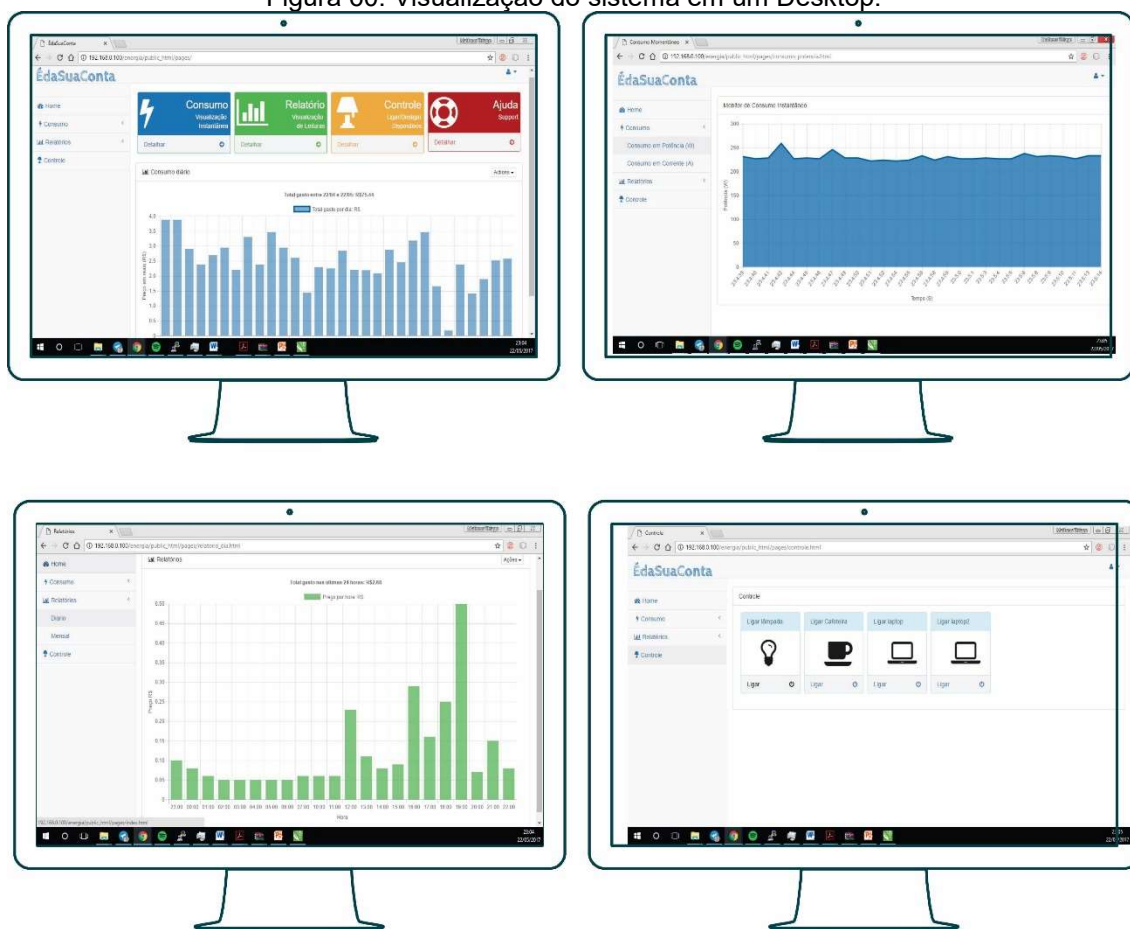
Figura 59: Visualização do sistema em um Tablet.



Fonte: O Autor.

A Figura 60 mostra a visualização das telas em dispositivos médios e grandes, com a largura de tela maior que 1200px com o sistema Windows 10 e navegador Google Chrome.

Figura 60: Visualização do sistema em um Desktop.



Fonte: O Autor.

Além dos sistemas e navegadores citados acima, o sistema foi testado no Sistema Operacional Ubuntu 14.04 em um desktop utilizando os navegadores Google Chrome, Mozilla Firefox e Opera. No Windows 10 ele foi testado com os navegadores Google Chrome, Microsoft Edge, Mozilla Firefox e Opera. Nesses testes não houveram mudanças na visualização das páginas. Dessa forma, foi observado um resultado satisfatório na aplicação das técnicas de design responsivo como: Mobile First, meta tag viewport, layouts fluídos, medidas flexíveis e media queries. Além do que, com o site da aplicação responsivo, as informações do site podem ser acessadas de maneira otimizada por qualquer dispositivo, com qualquer sistema operacional, bastando ter um navegador, e em qualquer resolução de tela dando maior conforto ao usuário e aperfeiçoando a sua usabilidade.



## 6. CONCLUSÃO

Os temas de automação residencial e IoT nos permitem a construção de várias implementações e soluções envolvendo o dia-a-dia de uma residência. A exemplo, quem não desejaria poder ligar ou deligar os eletrônicos e saber com exatidão e detalhamento o consumo de energia da sua própria casa remotamente?

Partindo desse princípio, construiu-se o ÉdaSuaConta, que é um sistema que auxilia a gestão e o controle dos gastos da energia residencial, tornando possível o controle de tomadas e a visualização detalhada dos gastos, com soluções utilizando hardware e software de baixo custo, sem perder a capacidade computacional.

Todos os hardwares e softwares utilizados no protótipo seguem o modelo de desenvolvimento de licenciamento livre open-source, o que torna o conhecimento sobre eles abertos, sendo possível a sua continuidade, modificação e expansão. O Hardware utilizado nesse protótipo teve o custo estimado de R\$ 400 (Quatrocentos reais). Por ser open-source, o Software não teve custos para obtenção de licenças de uso, alcançando uma redução ainda mais significativa quanto ao custo final da solução.

Após a construção do protótipo, foram aplicados testes que avaliaram o comportamento do sistema em determinadas situações. Os testes de leitura do consumo realizados na residência, por exemplo, foram bastante satisfatórios. Durante os dois meses em que o sistema atuou no local, o valor total do consumo do sistema foi muito próximo ao valor da nota fiscal da companhia de energia. Apesar de ter sido visualizado em outros testes que mesmo sem nada conectado à rede elétrica há uma leitura errônea na casa de 0.4A. Os testes feitos com o sistema de controle também foram satisfatórios.

Dessa forma, as contribuições desse trabalho consistiram em demonstrar a viabilidade de soluções de automação residencial de baixo custo, e em verificar detalhadamente os gastos quanto ao consumo energético auxiliando ao usuário ter uma melhor compreensão dos gastos permitindo o seu uso consciente. Destaca-se a vantagem do desenvolvimento das páginas web aplicando as técnicas de design responsivo, que tornaram possível acessar o sistema de maneira otimizada por qualquer dispositivo

que tenha um navegador, sempre se adaptando a largura de tela disposta. A interface do sistema auxiliou muito no processo de visualização dos dados de consumo, sendo possível verificar o gasto nas últimas 24 horas, e nos últimos 30 dias, além de verificar e modificar os status dos dispositivos conectados ao filtro de linha.

As principais dificuldades no desenvolvimento desse projeto foram:

- Multidisciplinaridade:
  - Para a construção do Software do Protótipo, foram necessários os princípios teóricos e práticos das disciplinas de Programação e Engenharia de Software, Redes de Computadores, Banco de Dados e Interação Humano-Computador;
  - Já na construção do Hardware, necessitou-se a utilização dos princípios teóricos e práticos das áreas de Arquitetura de Computadores, Robótica, Sistemas Embarcados, Física;
- Leituras incorretas em baixas Correntes do Sensor SCT013-000;
- Leitura apenas da corrente, de forma que a tensão fosse fixa, a leitura da potência aparente era estimada.

Dentro das propostas do início do projeto os objetivos gerais e específicos foram atingidos e obtiveram-se resultados muito promissores. Os testes práticos criaram expectativas otimistas para dar continuidade ao projeto com alguns aprimoramentos.

## 7. TRABALHOS FUTUROS

Nessa seção são mostradas algumas sugestões para o aperfeiçoamento do sistema proposto.

- Disponibilizar o acesso do sistema pela internet, não apenas em uma rede local;
- Utilizar apenas o NomeMCU para fazer a leitura dos sensores e atuar sobre os relés, enviando da informações pela internet;
- Utilizar um sensor de tensão para diminuir a aproximação do sistema;
- Testar outros sensores de corrente;
- Introduzir novos dispositivos ao projeto, como sensores de temperatura e umidade, sistema de alarme, sistema de som controlado pelo sistema e sistema de alarme residencial.
- Inserção de mais sensores ao quadro geral da residência para obter um maior detalhe quando ao consumo de energia como, por exemplo, apenas o total gasto pelo chuveiro elétrico;
- Adicionar funcionalidades ao controle, como uso de timers para o acionamento de relés, por tempo de operação ou dada uma certa hora.

## 8. REFERÊNCIAS

ANDRADE, Fernando Souza de; OLIVEIRA, André Schneider. **Sistemas Embarcados – Hardware e Firmware na Prática** 1ª. ed. São Paulo, SP: Érica, 2006.

ANEEL. BIG - Banco de Informações de Geração. **Agencia Nacional de Energia Elétrica**, 2016. Disponível em: <<http://www.aneel.gov.br/aplicacoes/capacidadebrasil/capacidadebrasil.cfm>>.

Acesso em: 19 de jul. 2016.

ARDUINO. O que é Arduino. **Arduino-introduction**, 2016. Disponível em: <[www.arduino.cc](http://www.arduino.cc)> Acesso em: 26 jul. 2016a.

ARDUINO. **Arduino Products**, 2016. Disponível em: <<https://www.arduino.cc/en/Main/Products>> Acesso em: 26 jul. 2016b.

ASHTON, Kevin. **That “Internet of Things” Thing**. RFID Journal, 22 jun. 2009. Disponível em: <<http://www.rfidjournal.com/articles/view?4986>>. Acesso em: 16 ago. 2016.

ATLAS DA ENERGIA ELÉTRICA DO BRASIL. **Parte II fontes renováveis, 4 Biomassa**, 2016. Disponível em: <[http://www2.aneel.gov.br/arquivos/pdf/atlas\\_par2\\_cap4.pdf](http://www2.aneel.gov.br/arquivos/pdf/atlas_par2_cap4.pdf)>. Acesso em 03 jul. 2016.

BATALLA, Jordi Mongay et al. **Beyond the Internet of Things: Everything Interconnected**. 1. ed. [S.I.]:Springer International Publishing, 2017. 210 p.

BATTISTI, Júlio. **SQL Server 2005: Administração e Desenvolvimento Curso Completo**. [S.I.]: Axcel Books do Brasil, 2005. 1 p.

BOCCHINI, Bruno. Pesquisa mostra que 58% da população brasileira usam a internet. Disponível em: <<http://agenciabrasil.ebc.com.br/pesquisa-e-inovacao/noticia/2016-09/pesquisa-mostra-que-58-da-populacao-brasileira-usam-internet>>. Acesso em: 22 nov. 2016.

BOLZANI, Caio A. M. **Desenvolvimento de um Simulador de Controle de Dispositivos Residenciais Inteligentes: Uma Introdução aos Sistemas**

Domóticos. 2004. 131 p. Escola Politécnica da Universidade de São Paulo, São Paulo, 2004.

BOOTSTRAP. **About**. Disponível em: <<http://getbootstrap.com/about/>>. Acesso em: 30 nov. 2016.

CALDWELL, Tracey. **Ford and Intel join to explore car biometrics**: Facial recognition. Biometric technology today, [S.l.], 19 jul. 2014. Elsevier, p. 3. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0969476514701298>>. Acesso em: 25 ago. 2016.

CASTOR, Michel. How I Built a Raspberry Pi Tablet. Disponível em: <<http://makezine.com/2014/01/07/how-i-built-a-raspberry-pi-tablet/>>. Acesso em: 12 set. 2016.

CEMIG: o que são as redes inteligentes de energia? Disponível em: <[http://www.cemig.com.br/pt-br/A\\_Cemig\\_e\\_o\\_Futuro/sustentabilidade/nossos\\_programas/Redes\\_Inteligentes/Paginas/as\\_redes\\_inteligentes.aspx](http://www.cemig.com.br/pt-br/A_Cemig_e_o_Futuro/sustentabilidade/nossos_programas/Redes_Inteligentes/Paginas/as_redes_inteligentes.aspx)>. Acesso em: 12 set. 2016.

CETIC.BR, . TIC Domicílios - 2015. Disponível em: <<http://cetic.br/pesquisa/domicilios/indicadores>>. Acesso em: 22 nov. 2016.

CHASE, Otavio. **Sistemas Embarcados**. Disponível em: <<http://www.lyfreitas.com.br/ant/pdf/Embarcados.pdf>>. Acesso em: 25 ago. 2016.

COULOURIS, George et al. **Sistemas distribuídos** : conceitos e projeto. 5. ed. Porto Alegre: Bookman, 2013. 1 p.

CURVELLO, André. **Sistema Web com Raspberry Pi, Arduino, USB, Lighttpd e PHP**. Disponível em: <<http://www.embarcados.com.br/sistema-web-com-raspberry-pi-e-arduino/>>. Acesso em: 01 set. 2016a.

CURVELLO, André. Apresentando o módulo ESP8266. Disponível em: <<https://www.embarcados.com.br/modulo-esp8266/>>. Acesso em: 28 out. 2016b.

DENNIS, Andrew K. **Raspberry Pi Home Automation with Arduino**: Automate your home with a set of exciting projects for the Raspberry Pi!. Birmingham, UK: Packt Publishing, 2013. 176 p.

EBC PORTAL: Entenda como funcionam as bandeiras tarifárias de energia. Disponível em: <<http://www.ebc.com.br/noticias/2015/02/entenda-como-funcionam-bandeiras-tarifarias-de-energia>>. Acesso em: 12 set. 2016.

EPE. **Consumo de Energia no Brasil: Análises Setoriais**. Empresa de Pesquisa Energética. Rio de Janeiro, p. 11,89. 2014. (NOTA TÉCNICA DEA 10/14).

EVANS, Dave. The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. [S.l.: s.n.], 2011. Disponível em: <[http://www.cisco.com/c/dam/en\\_us/about/ac79/docs/innov/IoT\\_IBSG\\_0411FINAL.pdf](http://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf)>. Acesso em: 24 ago. 2016.

FAMILY HUB REFRIGERATOR, **Family Hub Refrigerator**. Disponível em: <<http://www.samsung.com/us/explore/family-hub-refrigerator/>> Acesso em 15 ago. 2016.

FILIPPOV, Max. Esp8266-wiki: What is this ESP8266. Disponível em: <<https://github.com/esp8266/esp8266-wiki/wiki>>. Acesso em: 28 out. 2016.

FORD, **Ford and intel research demonstrates the future of in-car personalization and mobile interior imaging technology**. Disponível em: <<https://media.ford.com/content/fordmedia/fna/us/en/news/2014/06/25/ford-and-intel-research-demonstrates-the-future-of-in-car-person.html>> Acesso em 24 ago. 2016.

GIL, A. C. **Como elaborar projetos de pesquisa**. 5ª edição. São Paulo: Atlas, 2010.

HUANG, Rui. Nodemcu-devkit-v1.0. Disponível em: <<https://github.com/nodemcu/nodemcu-devkit-v1.0>>. Acesso em: 22 nov. 2016.

IBM; EUROTECH. MQTT V3.1 Protocol Specification. Disponível em: <<http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html#intro>>. Acesso em: 06 fev. 2017.

JAFFEY, Toby. MQTT and CoAP, IoT protocols. 2014. Disponível em: <[http://www.eclipse.org/community/eclipse\\_newsletter/2014/february/article2.php](http://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php)>. Acesso em 06 jan. 2017.

KONDO, M. M. Energia é fator determinante no desenvolvimento da sociedade. **Folha de S.Paulo**, São Paulo, 16 Fevereiro 2005. Disponível em:

<<http://vestibular.uol.com.br/resumo-das-disciplinas/geografia/energia-e-fator-determinante-no-desenvolvimento-da-sociedade.htm>>. Acesso em 10 set. 2016.

FRITZEN, Clovis. WiFi ESP8266: Diferentes modelos e suas funcionalidades. Disponível em: <<http://fritzenlab.com.br/2016/03/wifi-esp8266-diferentes-modelos-e-suas-funcionalidades/>>. Acesso em: 28 out. 2016.

LAFORGE, **Laforge Optical**. Disponível em: <<http://www.laforgeoptical.com/>> Acesso em 18 ago. 2016.

LE MOS, Marcelo. **Como fazer suas próprias PCBs** – placas de circuito impresso. Disponível em: <<http://blog.fazedores.com/como-fazer-suas-proprias-pcbs-placas-de-circuito-impresso/>>. Acesso em: 28 jul. 2016.

LIGHT, Roger. Mqtt: MQ Telemetry Transport. Disponível em: <<https://mosquitto.org/man/mqtt-7.html>>. Acesso em: 06 jan. 2017.

LOPES, Sergio. A Web Mobile: programe para um mundo de muitos dispositivos. São Paulo: Casa do Código, 2013. 28 p.

MARCOTTE, Ethan. Responsive Web Design. Disponível em: <<https://alistapart.com/article/responsive-web-design>>. Acesso em: 05 dez. 2016.

MCROBERTS, M. **Arduino Básico**. 1. Ed. São Paulo: Novatec 2011.

MEIO AMBIENTE TÉCNICO: Energia Hidrelétrica – Vantagens e desvantagens. Disponível em: <<http://meioambientetecnico.blogspot.com.br/2015/04/energia-hidreletrica-vantagens-e.html>>. Acesso em: 12 set. 2016.

MONK, S. **Raspberry Pi Cookbook**. O'Reilly Media. 2014.

MURATORI, José Roberto; DAL BÓ, Paulo Henrique. **Capítulo I - Automação residencial**: histórico, definições e conceitos. 2011. Disponível em: <[http://www.oseletrico.com.br/web/documentos/fasciculos/Ed62\\_fasc\\_automacao\\_capl.pdf](http://www.oseletrico.com.br/web/documentos/fasciculos/Ed62_fasc_automacao_capl.pdf)>. Acesso em: 25 set. 2016.

NETTO, Daniel. **O uso do Raspberry Pi pelos profissionais de eletrônica**. Saber Eletrônica: Industrial, São Paulo, ano 48, n. 468, p.12-17, mar./abr. 2013. Bimestral.

NICOLOSI, Denys E. C. **8051: Detalhado**. 6. ed. [S.l.]: Erica, 2000. 223 p.

NODEMCU. NodeMCU Documentation. Disponível em: <<http://nodemcu.readthedocs.io/en/master/>>. Acesso em: 22 nov. 2016.

NUMAJIRI, D. A. **Desenvolvimento de um sistema de controle de dispositivos via acesso remoto**. Lavras: UFLA, 2003.

OPENENERGYMONITOR. **About**. Disponível em: <<https://openenergymonitor.org/emon/sustainable-energy/>> Acesso em 05 jul. 2016a.

OPENENERGYMONITOR. **Current Transformer Installation**. Disponível em: <<https://learn.openenergymonitor.org/electricity-monitoring/ct-sensors/installation>>. Acesso em: 28 jul. 2016b.

OPENENERGYMONITOR. **CT Sensors** - Interfacing with an Arduino.

Disponível em: <<https://learn.openenergymonitor.org/electricity-monitoring/ct-sensors/interface-with-arduino>> Acesso em: 28 jul. 2016c.

OPENENERGYMONITOR. **How to Build an Arduino Energy Monitor - Measuring Mains Current Only**. Disponível em: <<https://learn.openenergymonitor.org/electricity-monitoring/ct-sensors/installation>>. Acesso em: 28 jul. 2016d.

PICHILIANI, Mauro. BANCO DE DADOS DESENVOLVIMENTO: Testando o MySQL no Raspberry Pi. Disponível em: <<https://imasters.com.br/banco-de-dados/testando-o-mysql-no-raspberry-pi/?trace=1519021197&source=single>>. Acesso em: 30 jan. 2017.

PIJUICE, A. **Raspberry Pi Portable Games Console**. Disponível em: <<http://www.instructables.com/id/Raspberry-Pi-Portable-Games-Console/>>. Acesso em: 12 set. 2016.

PIPSTA: The Raspberry Pi Printer With Big Ideas. Disponível em: <<http://www.raspberrypi-spy.co.uk/2015/02/pipsta-the-raspberry-pi-printer-with-big-ideas/>>. Acesso em: 12 set. 2016.

PORTAL BRASIL. **Potencial hidrelétrico brasileiro está entre os cinco maiores do mundo**. Disponível em: <<http://www.brasil.gov.br/infraestrutura/2011/12/potencial-hidreletrico-brasileiro-esta-entre-os-cinco-maiores-do-mundo>>. Acesso em: 12 set. 2016.



PORTAL DA ENERGIA. **Fontes de energia renováveis**. Disponível em: <<http://www.portal-energia.com/fontes-de-energia/>> Acesso: 12 set. 2016.

PROJECT JACQUARD, **Project Jacquard**. Disponível em: <<https://atap.google.com/jacquard/>>. Acesso em 15 ago. 2016.

PROTEUS DIGITAL HEALTH. Proteus Digital Health. Disponível em: <<http://www.proteus.com/>>. Acesso em 18 ago. 2016.

RASPBERRY Pi. **Raspberry Pi 1 model b+**. Disponível em: <<https://www.raspberrypi.org/products/model-b-plus/>>. Acesso em: 03 ago. 2016.

RILEY, Mike. **Programming your home: Automate with Arduino, Android, and your computer**. [S.I.]: The Pragmatic Programmers, 2012.

SAMSUNG. **Smart TV Ultra 4k 65" 3D**: SAMSUNG Brasil. Disponível em: <<http://www.samsung.com/br/consumer/tv-av/tv/uhd/UN65HU8500GXZD/>>. Acesso em: 15 ago. 2016a.

SAMSUNG. **Samsung Smart Things: For Your Connected Smart Home**. Disponível em: <<http://www.samsung.com/uk/smartthings/>>. Acesso em: 15 ago. 2016b.

SENA, Hamilton. **Automação Residencial e Internet das Coisas**. Disponível em: <<http://devidaasuaideia.com.br/hangout-automacao-residencial-e-internet-das-coisas/>>. Acesso em: 31 ago. 2016.

SIDDIQUI, O. **The Green Grid: Energy Savings and Carbon Emissions Reductions Enabled by a Smart Grid**. Electric Power Reserch Institute. Palo Alto, p. 1-1. 2008. (1016905).

SILVA, Mauricio Samy. Bootstrap 3.3.5: Aprenda a Usar o Framework Bootstrap para criar layouts CSS complexos e responsivos. . [S.I.]. 2015. 10 p. Disponível em: <[http://187.7.106.14/edecio/pi/livro\\_bootstrap.pdf](http://187.7.106.14/edecio/pi/livro_bootstrap.pdf)>. Acesso em: 21 fev. 2017.

SMARTER, **The conected kitchen- a new way to startyour day- Smarter**. Disponível em: <<http://smarter.am/>>. Acesso em 15 ago. 2016.

TANENBAUM, Andrew. S. **Redes de computadores**. 4. ed. Rio de Janeiro: Campus, 2003. 1 p.

THOMSEN, Adilson. **O que é Arduino?** Disponível em: <<http://blog.filipeflop.com/arduino/o-que-e-arduino.html>>Acesso em 07 Abr. 2016a.

THOMSEN, Adilson. QUAL MÓDULO ESP8266 COMPRAR? Disponível em: <<http://blog.filipeflop.com/wireless/qual-modulo-esp8266-comprar.html>>. Acesso em: 28 out. 2016b.

TEZA, Vanderlei Rabelo. **ALGUNS ASPECTOS SOBRE A AUTOMAÇÃO RESIDENCIAL - DOMÓTICA**. 2002. 23 f. Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2002. Disponível em: <<https://repositorio.ufsc.br/bitstream/handle/123456789/83015/212312.pdf>>. Acesso em: 17 ago. 2016.

VUJOVIC, Vladimir; MAKSIMOVIĆ, Mirjana. **Raspberry Pi as a Sensor Web node for home automation**. 44. ed. Tarrytown, NY, USA: Computers And Electrical Engineering, 2014. 18 p.

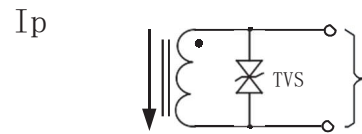
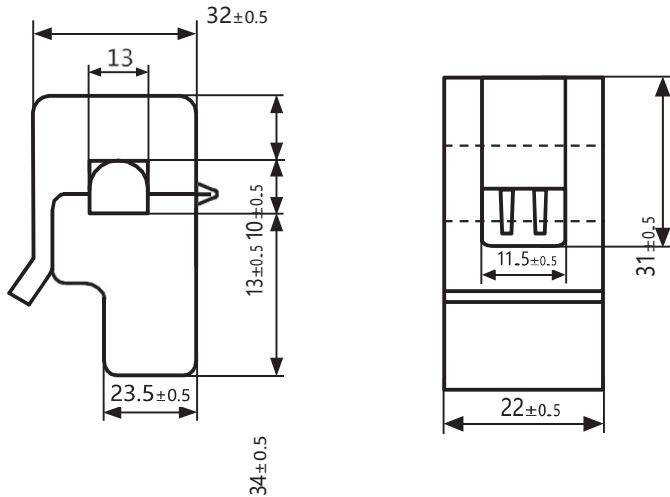
# ANEXO 1

Manufacture Model : SCT-013-000

Charateristics: open size:13mm×13mm  
 1m leading wire  
 Core material:Ferrite  
 Fire resistance property:in accordance with UL 94-V0  
 Dielectric strength: 1000V AC/1min 5mA (between shell and output)



Outline size diagram:(in mm)



TVS: Transient-voltage Suppressor

Current output type

Front View Diagram

Side View

Schematic

Typical table of technical parameters:

input current	output voltage	non-linearity	build-in sampling resistance (R <sub>L</sub> )
0-100A	0-50mV	±3%	Ω
turn ratio	resistance grade	work temperature	dielectric strength(between shell and output)
100A:0.05A	Grade B	-25°C~+70°C	1000V AC/1min 5mA

Customer Sign:

Beijing YaoHuadechang Electronic Co.,Ltd

Phone: 0355-7929499-803

Cell: 13693334514