

ALEX DO VALE DAS CHAGAS
IVAN STEINER TRUZZI
TOMAZ DE SOUZA SILVA

SMART PLUG

São Paulo
2015

ALEX DO VALE DAS CHAGAS

IVAN STEINER TRUZZI

TOMAZ DE SOUZA SILVA

SMART PLUG

Trabalho de conclusão de curso de Engenharia Elétrica com Ênfase em
Computação apresentado à Escola Politécnica da Universidade de São Paulo –
Departamento de Engenharia de Computação e Sistemas Digitais

Orientador: Prof. Dr. Marco Túlio Carvalho de Andrade

FICHA CATALOGRÁFICA

das Chagas, Alex do Vale; Truzzi, Ivan Steiner; Silva, Tomaz de Souza
Smart Plug / A.V. Chagas, I.S. Truzzi, T.S Silva - São Paulo, 2015.
127p.

Trabalho de Formatura - Escola Politécnica da Universidade de
São Paulo. Departamento de Engenharia de Computação e Sistemas
Digitais.

1. Engenharia de Computação 2. Automação Residencial 3. Smart Grid
4. Controle de Equipamentos. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

PENSAMENTOS

"A vida é muito curta para ser pequena"

Benjamin Disraeli

"Podemos apenas ver uma curta distância à frente,

mas podemos ver que há muito lá a ser feito"

Alan Turing

"Somos todos geniais. Mas se você julgar um peixe por sua capacidade de subir em árvores, ele passará sua vida inteira acreditando ser estúpido."

Autor desconhecido

DEDICATÓRIA

À minha família.

Alex do Vale das Chagas

À minha avó Luzia, e ao meu avô Vicente, por quem tenho a mais profunda
admiração.

Ivan Steiner Truzzi

Aos meus pais, meu irmão e meus amigos, pelo incentivo e pelo apoio constante.

Tomaz de Souza Silva

AGRADECIMENTOS

Agradecemos ao Professor Marco Túlio, pela sua orientação e por todas as suas valiosas ideias e conselhos na elaboração deste trabalho. Agradecemos aos nossos professores que nos repassaram seu conhecimento e foram de grande importância em nossa formação como engenheiros. Também gostaríamos de agradecer aos nossos amigos, que tornaram a nossa estadia na faculdade agradável e que nos ajudaram com seu comprometimento e amizade, nos momentos de estudo e trabalho. Em particular agradecemos ao nosso amigo Rafael, que sempre esteve disposto a nos ajudar com seu conhecimento, e nos deu contribuições importantes na realização deste trabalho. Finalmente queríamos fazer um agradecimento especial às nossas famílias, que nos deram todo o tipo de suporte, e que tornaram possível nossa passagem pela Escola Politécnica.

RESUMO

Neste trabalho foi construído um sistema de automação residencial que busca aumentar a eficiência do uso da energia elétrica, proteger os eletrodomésticos das flutuações de tensão na rede elétrica e garantir ao usuário maior conforto na interação com seus equipamentos, permitindo que ele os ligue/desligue remotamente, agende o acionamento destes e receba informações sobre o consumo de eletricidade de sua casa.

Foram definidos a especificação do sistema, as premissas adotadas e os requisitos. Também é analisado o contexto em que o sistema se insere, e as tecnologias já existentes. Foi concebida uma arquitetura em 4 níveis, considerando um hardware para obtenção de dados e execução de ações, um software web para tratar os dados, produzir relatórios, gráficos e enviar comandos, um software de comunicação entre estes (hardware e software web) e as formas pelas quais o usuário pode acessar o sistema através da interface criada. Foram explicadas as características destes níveis bem como a sua interação.

Um protótipo do sistema foi construído e submetido a um plano de testes para verificar sua funcionalidade. Também foi proposto um Plano de Negócios para verificar a viabilidade econômica do sistema como um produto. Finalmente foram expostas as conclusões finais do trabalho, considerando a abrangência do projeto, as dificuldades encontradas e os planos para aperfeiçoamentos futuros.

Palavras chave: Automação residencial, consumo de energia, interação remota, racionalização da eletricidade, Smart Grid, Smart Plug, Tomada Inteligente.

ABSTRACT

This Project involves a home automation system that aims to increase the efficiency on electrical energy use, to protect home appliances from tension fluctuations in the Power Grid and to grant the user a higher degree of convenience, allowing him to remotely interact with devices turning them on/off, scheduling their actuation and receiving reports on home power consumption.

It is defined the system specifications, hypothesis considered and project requirements. The context where the system is inserted and the existing technologies are also analyzed. An architecture in 4 levels was designed, considering a hardware for data acquirement and execution of commands, a web software to treat the data, to produce reports, graphs and send commands to the hardware, a communication software between them (hardware and web software), and the ways in which users may access the system through the created interface. The features of those levels were explained, just as their interaction.

A system prototype was built and then submitted to a tests plan to check its well functioning. It's also proposed a Business Plan to verify the economic viability of the system as a selling product. Finally it is exposed the work's final conclusions, considering the project's scope on engineering, difficulties found and plans for future improvements.

Keywords: Home automation, power consumption, remote interaction, electricity rationalization, Smart Grid, Smart Plug.

.

LISTA DE FIGURAS

Figura 1: Modelo DSP-W110 da D-Link [F1]	20
Figura 2: Modelo DSP-W215 da D-Link [F2]	21
Figura 3: WeMo Switch da empresa Belkin. [F3]	22
Figura 4: Triangulo de potência.....	24
Figura 5: Sinal da tensão de entrada (acima) e corrente de entrada deformada (abaixo) [F4].....	26
Figura 6: Harmônicos da corrente na figura 5 [F4].	27
Figura 7: Posicionamento das tecnologias sem fio [F5].....	28
Figura 8: Modelo de rede ZigBee [F5].....	31
Figura 9: Topologia adotada [F5].	32
Figura 10: Arquitetura de camadas do Sistema Smart Plug.....	38
Figura 11: Camada de Tomadas e seus componentes.....	39
Figura 12: Arquitetura lógica da central Doméstica.	39
Figura 13: Arquitetura lógica do Servidor Web.....	40
Figura 14: Formas pelas quais os usuários se conectam ao sistema Smart Plug.....	40
Figura 15: Arquitetura física do sistema.	41
Figura 16: Arquitetura geral do Sistema Smart Plug.....	42
Figura 17: Circuito do relé.....	43
Figura 18: Sensor de corrente e circuito [F6].....	44
Figura 19: Comportamento da tensão de saída dada a corrente de entrada.....	45
Figura 20: Lógica de funcionamento do programa que lê a corrente.....	45
Figura 21: Circuito do sensor de tensão.....	47
Figura 22: Componentes em blocos.....	47
Figura 23: Diagrama do circuito elétrico global.....	48
Figura 24: Circuito já com as trilhas.	49
Figura 25: Resultado a ser impresso para montagem do circuito.	50
Figura 26: Resultado do circuito impresso.....	50
Figura 27: Diferentes vistas do shield.	51
Figura 28: Diferentes vistas do shield integrado com Arduino.	52
Figura 29: Arduino Uno [F7].....	53
Figura 30: Módulo Xbee explorer usado (Sparkfun) [F8].....	53
Figura 31: Módulo Xbee [F9].....	54
Figura 32: Integração dos componentes na tomada – Todas as figuras utilizadas nessa composição, já foram referenciadas anteriormente.....	54
Figura 33: Integração dos componentes na central doméstica – Todas as figuras utilizadas nessa composição, já foram referenciadas anteriormente.....	55
Figura 34: Pico de Tensão pós religamento.	56
Figura 35: Onda sem pico de tensão.....	56
Figura 36: Funcionamento do software instalado no arduino.....	58
Figura 37: Esboço da arquitetura lógica da central doméstica.....	61
Figura 38: Divisão do programa 1 em duas threads.	63
Figura 39: Lógica de funcionamento da thread coleta_dados.....	64

Figura 40: Lógica de funcionamento da thread verifica_pendencias	66
Figura 41: Diagrama de navegação simplificado.....	71
Figura 42: Diagrama de Classes do software web.....	72
Figura 43: Gráfico da distribuição – Caso 1.....	74
Figura 44: Gráfico da distribuição – Caso 2.....	74
Figura 45: Método dos trapézios	76
Figura 46: Média total	79
Figura 47: Tela Inicial da Smart Plug	82
Figura 48: Tela de login ou Cadastro Inicial	82
Figura 49: Tela de Cadastro de Tomada.....	83
Figura 50: Tela de cômodos cadastrados	84
Figura 51: Tela de Monitoramento de Tomadas.....	84
Figura 52: Tela para criar agendamentos.....	85
Figura 53: Tela com todos os agendamentos realizados	86
Figura 54: Tela com o gráfico de consumo	86
Figura 55: Tela com o gráfico de potência	87
Figura 56: Mensagem de detecção de queda	88
Figura 57: Análise de SWOT da Smart Plug	103

SUMÁRIO

1.	Introdução à SMART PLUG	15
1.1.	Contexto e Motivação	15
1.2.	Objetivos	16
1.2.1.	Metas Intermediárias	16
1.3.	Descrição do Projeto	17
1.3.1.	Comodidade e Segurança	17
1.3.2.	Técnicas e Teorias	18
1.3.1.	Smart Grid	18
2.	Situação Atual do Mercado	20
2.1.	Modelo DSP-W110	20
2.2.	Modelo DSP-W215	21
2.3.	WeMo Switch	22
3.	Fundamentos Teóricos e Tecnologias	23
3.1.	Potência	23
3.1.1.	Potência em aparelhos	23
3.1.2.	Complicadores da medida de potência	25
3.1.3.	Medida adotada pelo grupo	28
3.2.	Comunicação sem fio	28
3.2.1.	Introdução ao ZigBee	29
3.2.2.	Relação com projeto	31
3.3.	Linguagem e Framework	32
3.3.1.	Linguagem Python	33
3.3.2.	Framework Django	34
3.4.	Arquitetura REST e RESTful Services APIs	35
4.	Arquitetura do Sistema SMART PLUG	37
4.1.	Requisitos do Sistema	37
4.2.	Arquitetura em 4 camadas	38
4.2.1.	Detalhes das camadas	39
4.2.	Arquitetura Física	40

5.	Projeto de hardware	43
5.1.	Relé	43
5.2.	Sensor de corrente	44
5.3.	Sensor de Tensão	46
5.4.	Interligação dos sensores	47
5.5.	Círculo Impresso.....	48
5.6.	Demais Componentes de Hardware	52
5.6.1.	Arduino	52
5.6.2.	Xbee Explorer	53
5.6.3.	Xbee.....	53
5.6.4.	Integração dos componentes na tomada	54
5.6.5.	Integração dos componentes na central.....	55
5.7.	Detector de queda.....	55
5.8.	Software da Tomada (Arduino)	57
6.	Central Doméstica	60
6.1.	Requisitos funcionais do Software da central doméstica	60
6.2.	Arquitetura lógica da central doméstica.....	60
6.3.	Composição da central doméstica	61
6.4.	Conceitos importantes	62
6.5.	Descrição dos Programas	63
6.5.1.	Programa p1_core	63
6.5.2.	Programa p2_send_data	68
6.5.3.	Programa p3_schedule.....	69
6.5.4.	Programa p4_send_state	69
7.	Software web	70
7.1.	Requisitos do software web	70
7.2.	Protótipo de navegação	71
7.3.	Diagrama de Classes	71
7.4.	Construção dos Gráficos de Consumo	73
7.4.1.	Método dos Retângulos	74
7.4.2.	Método dos Trapézios.....	76
7.4.3.	Método da Média Total.....	79
7.4.4.	Últimas considerações sobre os Gráficos de Consumo.....	80

7.5.	Detecção de queda de energia ou detecção de falha de comunicação via Software.....	80
7.6.	Funcionalidades do site mySmartPlug	81
7.6.1.	Cadastro de Usuário	82
7.6.2.	Envio de Convite.....	83
7.6.3.	Cadastro de Tomada	83
7.6.1.	Lista de Cômodos	83
7.6.2.	Monitoramento das tomadas	84
7.6.3.	Agendamentos	85
7.6.4.	Gráficos	86
7.6.5.	Produção e Envio de Relatorios	87
7.6.6.	Detecção de Queda	87
8.	Plano de testes e atingimento de metas.....	89
8.1.	Plano de testes proposto	89
8.2.	Atingimento das metas estabelecidas.....	92
8.2.1.	Tabela de Testes do Sistema Integrado	93
8.2.2.	Atingimento de metas.....	94
9.	Business Plan	95
9.1.	Ideia	95
9.2.	Mercado	96
9.3.	Modelo de Negócios.....	97
9.4.	Publico Alvo	97
9.5.	Precificação	97
9.6.	Expectativa de Faturamento	100
9.7.	Análise de SWOT	103
9.8.	Possibilidades Futuras	104
10.	Considerações Finais	105
10.1.	Contribuições	105
10.2.	Dificuldades	106
10.3.	Abrangencia do projeto.....	106
10.4.	Trabalhos Futuros.....	107
10.4.1.	Algoritmo estocástico para aquisição de dados.....	107
10.4.2.	Detecção de Perfil	108
10.4.3.	Controle do Stand-by	108

10.4.4. Criar um aplicativo para Android e para iOS	108
10.4.5. Adicionar uma porta USB	109
10.4.6. Customização de Relatório.....	109
10.4.7. Interação com a Smart Grid	109
10.5. Conclusão	110
11. Refeências	112
12. Referências de Figuras.....	115
13. Apêndice A – Definição de Requisitos.....	117
14. Apêndice B – Wire Frame.....	121
14.1. Wire Frame das telas iniciais	121
14.2. Wire Frame dos relatórios.....	124
15. Apêndice C – Informações sobre o CD	127

1. INTRODUÇÃO À SMART PLUG

Neste capítulo o leitor pode encontrar os acontecimentos que deram origem à ideia deste projeto. Estabelecerá seus objetivos e dará uma breve descrição de seu funcionamento.

1.1. Contexto e Motivação

Uma tendência mundial é a busca pela racionalização do uso dos recursos naturais, buscando uma economia sustentável que permita às sociedades terem um nível digno de vida, sem exaurir os recursos do planeta. Esta racionalização começa desde a maneira em que fabricamos a energia, até o modo como a consumimos.

Especificamente no caso do Brasil, recentemente temos passado por uma crise hídrica, que não só prejudica abastecimento de água nas residências, como também diminui a produção nas usinas hidroelétricas, encarece a tarifa de energia e nos coloca em risco de desabastecimento.

Aproveitando-se de um conceito cada vez mais difundido, a “Internet das Coisas”, é possível criar diversos aparelhos visando à diminuição do consumo nas residências, através de técnicas da Engenharia Elétrica.

Nesse contexto surge a ideia da Tomada Inteligente, que nomeamos como “*SMART PLUG*”, um dispositivo capaz de levar ao usuário informações relevantes sobre o consumo elétrico e proporcionar ao mesmo a tomada de decisões de consumo e tornar sua vida mais cômoda.

1.2. Objetivos

O objetivo do nosso projeto é criar um sistema que auxilie uma família a reduzir o seu consumo de energia elétrica e que gere comodidade aos seus usuários. Desse modo, constituindo-se em uma contribuição para o uso racional dos recursos naturais da sociedade, e se inserindo no moderno conceito de Internet das Coisas que provavelmente possibilitará um grande mercado nas próximas décadas.

Este objetivo será alcançado através do cumprimento de algumas metas intermediárias.

1.2.1. Metas Intermediárias

- Desligamento de equipamentos ociosos, redução drástica do consumo de energia pelo *Stand-By*, responsável por 20% do gasto de eletricidade nos eletrodomésticos [1][2][3][4]. Tornando essas medidas automáticas fazemos com que o usuário tenha menos trabalho.
- Programação de acionamento/desligamento de equipamentos à distância, evitando consumo desnecessário e permitindo ao usuário melhor controle sobre seus equipamentos.
- Medições e verificações de padrões de consumo, tornando possível a troca de equipamentos fora da especificação e possível venda de energia elétrica para a rede.
- Ao detectar uma queda de energia elétrica na rede e sua subsequente volta, a tomada deve manter o circuito aberto até que a voltagem esteja estável no valor normal de operação (110V ou 220V). Isso evita que o aparelho ligado à tomada queime.

1.3. Descrição do Projeto

O projeto visa à criação de uma Tomada Inteligente acoplada à tomada convencional, conferindo a esta novas funcionalidades.

A tomada deve ser capaz de monitorar o consumo de energia elétrica dos equipamentos que estejam acoplados a ela, bem como permitir o ligamento/desligamento remoto. O ligamento/desligamento deverá ser feito por meio de uma plataforma WEB. O usuário precisa ser identificado por Login nessa plataforma e assim poder acessar a funcionalidade. Também deve ser possível, pela mesma plataforma WEB, a visualização do histórico de consumo total dos dispositivos acoplados à tomada.

Como pode ser entendido pelo exposto acima, a tomada, algo comum nas residências, passará a se conectar com a internet, isso está incluso no conceito extremamente atual de internet das coisas, o qual prevê que cada vez mais objetos comuns do cotidiano de cada pessoa se conectem a internet, possibilitando não só uma interação maior com o usuário como também uma nova experiência.

1.3.1. Comodidade e Segurança

Com a tomada inteligente será possível otimizar a rotina dos usuários, imagine por exemplo, que uma pessoa que não dispõe de muito tempo para preparar seu almoço, nesse caso ela pode deixar a comida no forno e, antes de chegar em casa, acionar o eletrodoméstico de modo que a comida esteja pronta quando essa pessoa chegar em casa.

As funcionalidades de acionamento remoto e checagem de consumo podem ser usadas também em questões de segurança. Imagine, por exemplo, uma pessoa que saiu de casa e tenha deixado o ferro de passar roupas ligado, com a tomada inteligente ela poderá checar se de fato esqueceu o ferro ligado e em caso positivo efetuar o desligamento. Isso evitaria um possível incêndio.

Com o medidor de consumo acoplado à tomada é possível fazer um estudo posterior do consumo em “*stand by*” que os aparelhos costumam ter. Nesse caso, ao detectar

que o equipamento está em “*stand by*”, a tomada se desligaria, economizando energia. Vale ressaltar que essa funcionalidade não está no escopo do projeto, mas é uma possível extensão que pode ser feita no software da tomada em alguma possível versão futura desse projeto.

1.3.2. Técnicas e Teorias

O desenvolvimento do projeto permitirá a prática de várias teorias estudadas no curso. Primeiramente teremos de recordar a matéria de circuitos elétricos para o cálculo do consumo. Nesse ponto deve-se levar em conta não só a tensão e a corrente sendo consumidas, mas também a diferença de fase entre elas.

O estudo de ferramentas de prototipagem tais como Arduino e Raspberry PI também é de suma importância na formação de um engenheiro. Além disso, quando formos estudar os sensores a serem usados teremos contato com a teoria eletrônica estudada. Uma possível tarefa que teremos que realizar é a montagem de uma placa de circuito impresso que contenha os sensores usados, entretanto isso é um item secundário no escopo do projeto.

Do ponto de vista de software teremos a oportunidade de exercitar a teoria de engenharia de software criando uma arquitetura cliente servidor e fazendo programação web. Além disso, reforçaremos os conceitos teóricos estudados na disciplina de banco de dados relacionais.

1.3.1. Smart Grid

O projeto tange outro conceito importante, conhecido como “*Smart Grid*”. Com a tendência atual das coisas se tornarem elétricas (panelas, carros, trens...) e de que as casas do futuro produzam energia elétrica é uma questão de tempo para que o sistema de distribuição de energia fique mais dinâmico e inteligente, com os usuários interferindoativamente em seu funcionamento.

Nesse sistema espera-se que se crie um verdadeiro mercado em torno da energia elétrica. Imagine que alguém tenha um carro elétrico, se em um determinado dia essa pessoa não usar o carro ela pode carregar a bateria quando a cotação da

energia estiver mais baixa (normalmente de madrugada) e vender essa energia para a rede no horário de pico, lucrando com isso. Nesse cenário, a tomada inteligente poderia ser usada para o agendamento do horário de ligamento e desligamento do carro elétrico. Ou mesmo detectar que o carro esteja carregando (por meio do medidor de consumo) e desligar automaticamente ao detectar a alta cotação.

Além da diferença na cotação devido aos horários de pico de consumo, o que é caracterizado por uma variação em sua demanda, há também a previsão de que a tarifa da energia elétrica também irá variar de acordo com sua oferta. Isso ocorrerá pois as grandes usinas e centrais elétricas não serão as únicas geradoras de eletricidade, mas possivelmente cada residência, comércio ou indústria, serão unidades geradoras, que venderão sua energia excedente para a rede elétrica. As formas principais de geração nestas unidades serão solar, geradores por combustíveis fósseis ou mesmo eólica, assim um dia com muita incidência de energia solar ou com muitos ventos, pode provocar um aumento na oferta de energia, ao passo que, a ausência destes fatores pode levar a diminuição da oferta e aumento de preço. Neste contexto um sistema que ajuda a gerenciar este comércio é de grande utilidade para o usuário.

2. SITUAÇÃO ATUAL DO MERCADO

Em 2014 foram lançados alguns modelos de Tomadas Inteligentes no mercado. A empresa D-Link oferece dois modelos de Tomadas Inteligentes, sendo que ambos funcionam por meio de Wi-Fi e são controlados por smartphones e tablets (tanto os que usam sistema Androide como IOS). A empresa Belkin oferece um modelo apenas, semelhante ao modelo mais simples da D-Link. Abaixo estão listados os principais modelos e suas características.

2.1. Modelo DSP-W110

Na figura 1, pode-se ver as características físicas do modelo DSP-W110.



Figura 1: Modelo DSP-W110 da D-Link [F1]

Funcionalidades do modelo:

- Desligar e ligar dispositivos remotamente;
- Possibilita agendamento de horários;
- Usa conexão Wi-Fi.

2.2. Modelo DSP-W215

Na figura 2 pode-se ver as características físicas do modelo DSP-W215.



Figura 2: Modelo DSP-W215 da D-Link [F2]

Funcionalidades no modelo:

- Desligar e ligar dispositivos remotamente;
- Possibilita agendamento de horários;
- Usa conexão Wi-Fi;
- Possibilita o monitoramento de energia;
- Possui proteção térmica.

O modelo DSP-W215, além de implementar as funcionalidades do DSP-W110 também possui um sistema de cálculo da potência consumida e conta com um sensor térmico que alerta o usuário caso a temperatura na tomada ultrapasse um limiar também definido pelo usuário.

Consultando os sites que vedem as Tomadas e analisando as queixas dos consumidores foi levantado um conjunto dos principais problemas nos aparelhos.

Problemas:

- Muitos usuários criticaram o Firmware do dispositivo, relatando uma série de problemas ao usar o aparelho;
- Caso não seja usado com roteadores da mesma marca, a tomada pode apresentar problemas;

- Não permite o uso em 220V.

2.3. WeMo Switch

Na figura 3, pode-se ver as características físicas do modelo WeMo.



Figura 3: WeMo Switch da empresa Belkin. [F3]

Funcionalidades do modelo:

- Desligar e ligar dispositivos remotamente;
- Possibilita agendamento de horários;
- Usa conexão Wi-Fi.

Em todos os aparelhos vistos no mercado, o firmware e os aplicativos para smartphone e tablete foram severamente criticados. Os dispositivos também não custam barato, todos estão na faixa dos 50 dólares. Tanto os dispositivos da D-Link como o da Belkin ainda não estavam disponíveis no Brasil no dia em que foi feita a pesquisa (20/05/2015).

3. FUNDAMENTOS TEÓRICOS E TECNOLOGIAS

Este capítulo apresenta os fundamentos teóricos revisados para o desenvolvimento do sistema Smar Plug. Será dada ênfase aos estudos realizados para relembrar conceitos de medição de potência e estudos comparativos de tecnologia de comunicação sem fio.

No fim do capítulo será exposto um breve descritivo da linguagem Python, do framework Django e do modelo de arquitetura REST. Todos estes foram tecnologias utilizadas no desenvolvimento do sistema Smart Plug.

3.1. Potência

Nesse tópico serão explicados os tipos de potência existentes e a relação com o usuário final. Também serão vistas as complicações e conjunturas que levaram o grupo a adotar uma medida de potência diferente do modelo ideal, entretanto essa escolha será justificada. Será visto que essa escolha vai ao encontro das políticas e tendências mundiais.

3.1.1. Potência em aparelhos

Equipamentos que tenham componentes com caráter indutivo ou capacitivo tais como motores e fontes de alimentação dissipam dois tipos de potência: ativa (medida em W) e reativa (medida em VAR - Volt Ampère Reativo). A potência ativa é aquela que produz trabalho, como por exemplo, a rotação de um motor. Já a potência reativa não produz trabalho, ela é usada para estabelecer fluxos magnéticos e elétricos. A soma vetorial dessas duas potências resulta na potência aparente (medida em VA). Na figura 4 pode-se ver o triangulo de potência em que P representa a potência ativa (parte real), Q a potência reativa (parte imaginária) e S a potência aparente (soma vetorial de P e Q) [6].

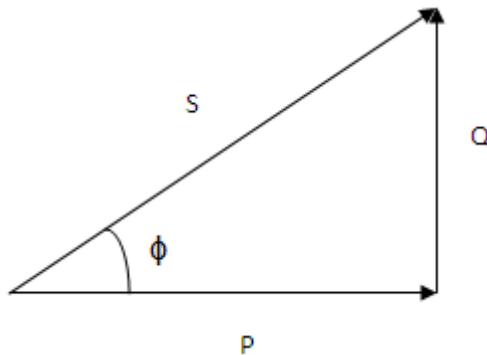


Figura 4: Triangulo de potência

O fator de potência (FP) é definido, como pode ser visto na equação abaixo, como a razão entre a potência ativa e a potência aparente. Para o caso de sinais senoidais simples (harmônica fundamental apenas) o fator de potência assume a forma de $\cos(\phi)$ onde ϕ é o ângulo entre S e P (figura 4). Ou seja, o fator de potência é um indicador da proporção de energia ativa e reativa consumidas pelo sistema, se o fator de potência é próximo de 0 isso significa que o sistema consome muita energia reativa relativamente à ativa e se o fator de potência é próximo de 1 o sistema consome muita energia ativa relativamente à reativa.

$$FP = \frac{P}{|S|} , \text{ sendo } S = P + jQ \text{ e } j = \sqrt{-1}$$

A energia reativa é aquela que é trocada de forma cíclica entre o componente e a rede e cria problemas para rede tais como a subutilização ou sobrecarga da instalação devido ao espaço ocupado pela potência reativa, perdas por aquecimento ou efeito joule e quedas de tensão devido à sobrecarga nos cabos [7].

O Brasil, bem como vários outros países, limita o fator de potência a um mínimo. Isso serve para otimizar o consumo impedindo que o usuário consuma uma parcela grande de energia reativa. Como pode ser visto na resolução de número 456 da ANEEL (Agência Nacional De Energia Elétrica) clientes industriais devem ter um fator de potência de no mínimo 0,92. Caso o cliente industrial tenha um fator de potência inferior a esse ele paga uma multa proporcional ao desvio do fator de potência mínimo [8]. Vale ressaltar que o usuário comum residencial não precisa seguir essa regra, ela só é válida para usuários de grande porte.

Vários países estão adotando legislações que obrigam os fabricantes de equipamentos de eletroeletrônicos voltados ao usuário final a também seguirem a regra do fator de potência. No início de 2001 a União Europeia passou a exigir que todos os eletroeletrônicos vendidos com potência superior a 70W passem a ter circuitos de correção de potência de modo a consumirem a menor quantidade possível de energia reativa [6]. Assim é esperado que outros países sigam essa linha.

Um ponto importante a ressaltar é que um fator de potência baixo não significa eficiência baixa. Um aparelho que conta com circuito corretivo não consome menos eletricidade (do tipo que se paga). Isso porque o valor cobrado nas contas de energia elétrica considera apenas o consumo de energia ativa. Assim, o ônus de pagar pelo circuito corretivo fica com o usuário final, mas o real interessado no baixo consumo de energia reativa são as companhias distribuidoras.

3.1.2. Complicadores da medida de potência

Pelo que foi exposto anteriormente a potência de interesse, no caso de um aparelho doméstico qualquer, é a potência ativa, ou potência útil. Mas o cálculo dessa potência tem algumas barreiras que serão descritas nesse tópico.

Antes do aparecimento dos semicondutores e dispositivos de estado sólido, os circuitos elétricos eram compostos basicamente por elementos com características resistivas, indutivas e capacitivas (circuito RLC). Esses componentes também são denominados como componentes passivos e têm a propriedade de não deformar o sinal de entrada. As cargas com essas características são chamadas de cargas lineares [9].

Atualmente, a maioria dos sistemas é composta por cargas não lineares, ou seja, também possuem componentes ativos como diodos, transistores, tiristores, triacs, etc. Tais componentes deformam a onda de entrada. Exemplos desses sistemas são computadores, fontes de alimentação, variadores de velocidade, etc. [9]. Na figura 5 possível ver a imagem com o sinal de tensão senoidal, mas com corrente deformada.

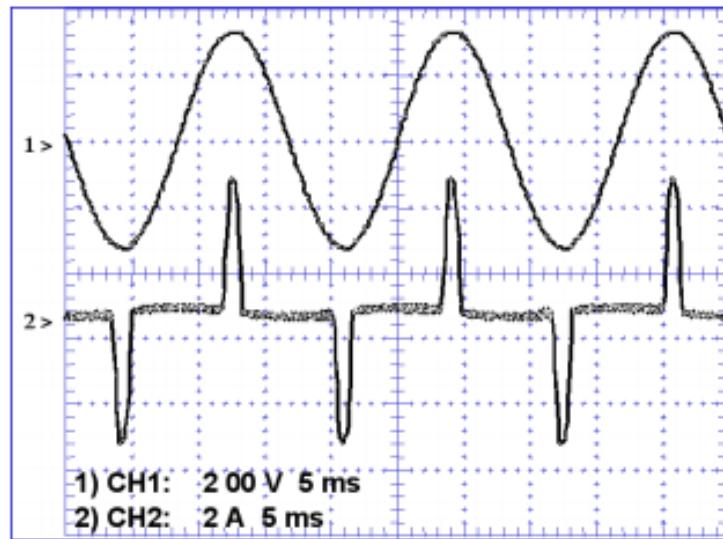


Figura 5: Sinal da tensão de entrada (acima) e corrente de entrada deformada (abaixo) [F4].

A medida da potência ativa é definida como o valor médio da potência instantânea (definição independente da forma da onda). Ou seja:

$$P = \overline{p(\sigma)} = \frac{1}{T} \int_0^T p(\sigma) d\sigma$$

Quando se trata de sinais não senoidais como o da figura 5 o sinal deve ser decomposto em suas frequências harmônicas e a potência ativa é gerada por tensões e correntes de mesma frequência. Desse modo, para sinais periódicos a potência ativa total é dada pela seguinte equação:

$$P = I_0 U_0 + \sum_{k=1}^{\infty} U_k I_k \cos \varphi_k$$

Como exemplo, para a forma de onda da corrente na figura 5 têm-se as seguintes frequências harmônicas:

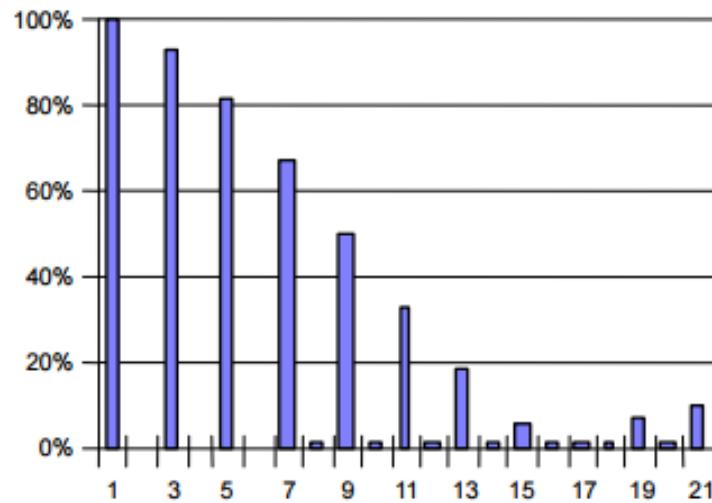


Figura 6: Harmônicos da corrente na figura 5 [F4].

Como só há um harmônico de tensão, a única componente da corrente que contribui para potência ativa nesse caso é a primeira.

Assim, diante do que foi comentado acima e de demais pesquisas o grupo levantou os seguintes fatores:

- Tendência de os eletroeletrônicos começarem a vir com fator de potência próximo a 1 (legislação Europeia);
- Complexidade da medida da potência ativa (incompatível com os sensores disponíveis);
- A medida do fator de potência com os sensores disponíveis conteria erros que acabariam comprometendo as medidas;
- O esforço necessário para a criação de sensores novos seria demasiado e comprometeria a execução das demais etapas do projeto.

3.1.3. Medida adotada pelo grupo

Diante do exposto anteriormente o grupo optou por fazer um dispositivo que meça a potência aparente.

$$S = U_{\text{RMS}} I_{\text{RMS}}$$

O valor eficaz da tensão pode ser medido tanto por um sensor RMS simples como por um True-RMS já que a tensão da rede sempre terá a forma senoidal. Já o valor eficaz de corrente deve ser medido com um sensor True-RMS para que as frequências harmônicas presentes no sinal também sejam levadas em conta. Vale lembrar as frequências harmônicas mudam de eletroeletrônico para eletroeletrônico gerando deformações diferentes na corrente em aparelhos diferentes.

3.2. Comunicação sem fio

Para o projeto, as tecnologias Wi-Fi, Bluetooth e ZigBee foram vistas como as principais possibilidades para comunicação da tomada com a central. Dessa forma, foi feito um comparativo básico entre essas tecnologias para que pudéssemos escolher a mais conveniente ao projeto da Smart Plug.

A figura 7 oferece um panorama, com relação a alcance e taxa de transmissão das tecnologias de comunicação sem fio disponíveis.

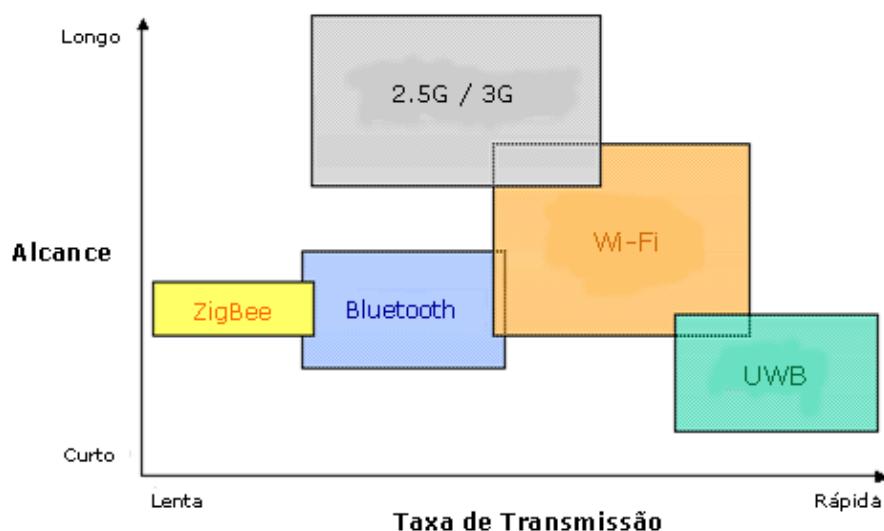


Figura 7: Posicionamento das tecnologias sem fio [F5].

A tecnologia Bluetooth permite apenas 8 dispositivos numa mesma rede, isso limitaria o número de tomadas a 7 (já que a central também é um nó da rede). Além disso, é previsto que, no futuro, diferentes tipos de dispositivos se comuniquem entre si e isso aumenta a demanda por uma tecnologia que suporte mais dispositivos em rede. Assim, esse ponto foi crucial e, portanto, descartamos a tecnologia Bluetooth.

A tecnologia Wi-Fi, por outro lado possibilita um número razoável de dispositivos ligados à rede (255), mas possui custo elevado além de consumir muita energia quando comparada com o Bluetooth e o ZigBee. Como um dos objetivos do projeto é auxiliar o usuário a economizar energia não faria sentido escolhermos a tecnologia que mais consome energia para fazer a comunicação sem fio.

Foi construída uma tabela simples, mostrada abaixo, que auxiliou o grupo na tomada de decisão.

	ZigBee	Bluetooth	Wi-Fi
Custo	Baixo	Baixo	Alto
Consumo	Muito baixo	Baixo	Alto
Taxa de transmissão	Baixa	Media	Alta
Número de dispositivos	65535	8	255

Tabela comparativa de tecnologias sem fio.

Diante do exposto acima, o grupo optou pelo ZigBee já que é uma tecnologia voltada para o baixo consumo de energia e permite um elevado número de dispositivos na mesma rede. A transmissão de dados no projeto não consome muita banda, isso faz do ZigBee mais uma vez a escolha ideal.

3.2.1. Introdução ao ZigBee

O ZigBee é uma tecnologia relativamente nova, foi apresentada ao público em 2005. Ela foi definida por um conjunto de mais de 200 empresas de diferentes segmentos do mercado que formaram a ZigBee Alliance. Seu surgimento se deve basicamente a carência de um padrão que supra as necessidades de comunicação sem fio entre sensores e dispositivos de controle.

Por necessidades pode-se entender:

- Baixa latência;
- Eficiência energética;
- Possibilidade de implementação de redes com elevado número de dispositivos;
- Baixa complexidade dos nós da rede.

As redes ZigBee possuem dois tipos de dispositivos: os de função reduzida (RFD - Reduced Function Device), e os de função completa (FFD - Full Function Device).

Os dispositivos FFD são aqueles aptos a funcionarem em qualquer um dos modos de operação do padrão: coordenador, roteador ou dispositivo final. Podem se comunicar tanto com outros FFD quanto com dispositivos RFD [10].

Os dispositivos RFD, por sua vez, são dispositivos que só podem se comunicar com dispositivos FFD. Dessa forma fica claro que esses dispositivos poderão atuar apenas como end-pointings da rede. São dispositivos mais simples e de menor custo, visando um consumo de energia ainda mais reduzido [10].

O dispositivo pode ser classificado em três funções logicas: coordenador, roteador ou dispositivo final. Conforme a sua posição na rede e sua disponibilidade de funções (FFD ou RFD).

- Coordenador: Função mais completa, o dispositivo coordenador inicia a rede, pode ser visto como raiz da rede além de poder funcionar como ponte para outras redes. Esse dispositivo é capaz de armazenar informações sobre a rede incluindo, por exemplo, o repositório das chaves de segurança.
- Roteador: Função que permite o uso do dispositivo como roteador intermediário, transmitindo dados de outros dispositivos. Um dispositivo roteador também pode executar uma aplicação final.

- Dispositivo final: Função mais simples, permite apenas a comunicação com o dispositivo pai (coordenadores ou roteadores), ele não transmite dados que não sejam provenientes dele mesmo. Essa configuração permite que o dispositivo fique adormecido por um tempo significativo, o que proporciona economia de energia. Esse tipo de dispositivo também requer menos memória e, portanto, possui fabricação mais barata.

Na Figura 8 é possível visualizar as diferentes topologias de rede possíveis com os dispositivos assumindo as três funções citadas acima.

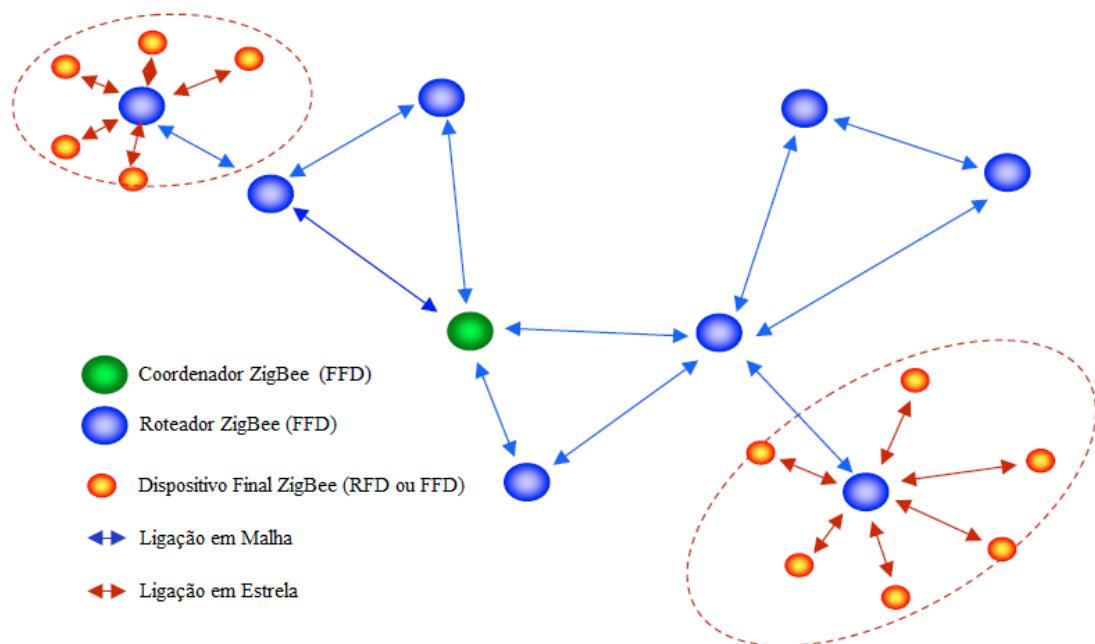


Figura 8: Modelo de rede ZigBee [F5].

3.2.2. Relação com projeto

Foi escolhido o protocolo ZigBee pelos seguintes fatores:

- Taxa de transmissão exigida pelo sistema é pequena;
- Eficiência energética;
- Suporte a grande número de dispositivos numa mesma rede (máximo de 65535 dispositivos por cada dispositivo coordenador);

- Grande alcance já que um dispositivo pode trocar informação com todos os demais dispositivos na rede até atingir seu alvo;
- Interoperabilidade, ou seja, a capacidade de se comunicar de forma transparente com outros sistemas.

Por questões de economia de energia e sabendo de antemão que o alcance do sensor escolhido não é muito baixo (carca de 30m em ambientes internos) o módulo ZigBee localizado na tomada será configurado como um dispositivo final. Caso o alcance do sensor fosse menor o grupo deveria configurar o módulo ZigBee da tomada como dispositivo de função completa (no caso, roteador) para que ele pudesse rotear as informações provenientes de outra tomada que por ventura estivesse fora do alcance da central.

Como o alcance do sensor escolhido é maior que imaginado inicialmente, optou-se pela topologia em estrela da rede como pode ser visto na Figura 9.

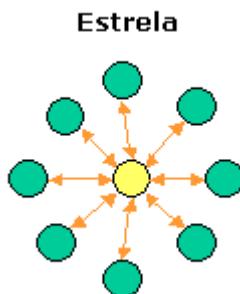


Figura 9: Topologia adotada [F5].

3.3. Linguagem e Framework

Durante o processo de escolha da linguagem e do ambiente de programação mais apropriado para o desenvolvimento desse projeto nos deparamos com inúmeras possibilidades, entretanto tomamos a decisão de utilizar a linguagem de programação Python com o Framework Django. Os fatores listados a seguir foram determinantes para essa decisão:

- Ambos possuem licença compatível com Software Livre;
- Fácil manipulação de banco de dados através de Mapeamento objeto-relacional – sem a necessidade de usar SQL;

- As operações CRUD (Create, Read, Update e Delete) são implementadas de forma trivial;
- O fato de ser “batteries included”- ter uma biblioteca padrão rica e versátil, que está disponível imediatamente, sem a necessidade de o usuário fazer download de pacotes separados [11] - aumenta significativamente a produtividade;
- Django tem a maior comunidade e aceitação entre os frameworks web Python.

A seguir faremos uma breve descrição da linguagem e do framework adotados.

3.3.1. Linguagem Python

Python é uma linguagem de propósito geral de alto nível, multi paradigma, suporta o paradigma orientado a objetos, imperativo, funcional e procedural. Possui tipagem dinâmica e uma de suas principais características é permitir a fácil leitura do código e exigir poucas linhas de código se comparado ao mesmo programa em outras linguagens.

Apesar de muitas pessoas associarem o nome Python à cobra homônima, o nome foi inspirado no grupo humorístico britânico Monty Python. Essa linguagem foi concebida no final de 1989 por Guido van Rossum no Instituto de Pesquisa Nacional para Matemática e Ciência da Computação (CWI), nos Países Baixos, como um sucessor da linguagem ABC, também da CWI, capaz de tratar exceções e prover interface com o sistema operacional Amoeba através de scripts [12].

Atualmente possui um modelo de desenvolvimento comunitário, aberto e gerenciado pela organização sem fins lucrativos Python Software Foundation. Apesar de várias partes da linguagem possuírem padrões e especificações formais, a linguagem como um todo não é formalmente especificada.

Python Combina uma sintaxe concisa e clara com os recursos poderosos de sua biblioteca padrão e por módulos e frameworks desenvolvidos por terceiros. Como já comentado anteriormente, no desenvolvimento desse projeto usaremos o framework Django.

3.3.2. Framework Django

Django é um framework para desenvolvimento rápido para web escrito em Python. Utiliza o padrão de arquitetura de software MTV (Model/Template/View), esse padrão divide uma aplicação de software em três partes interligadas separando as representações internas de informações a partir das formas que a informação é apresentada ao usuário.

Django foi criado originalmente como sistema para gerenciar um site jornalístico na cidade de Lawrence, no Kansas. Foi projetado para lidar com dois desafios: os prazos apertados de uma redação e os requisitos rígidos dos experientes desenvolvedores web que o criaram [13]. Tornou-se um projeto de código aberto e foi publicado sob a licença BSD em 2005. O nome Django foi inspirado no músico de jazz Django Reinhardt.

Django se concentra no máximo de automatização possível lançando mão do princípio DRY (Don't Repeat Yourself) fazendo com que o desenvolvedor aproveite ao máximo o código já feito, evitando a repetição.

O Django tem uma linguagem de templates poderosa, extensível e amigável. Com ela pode-se separar design, conteúdo e código em Python. Algumas de suas características são:

- Mapeamento objeto-relacional: os modelos são definidos em Python e possuem uma API que facilita a manipulação dos dados. Apesar disso é possível escrever comandos SQL caso seja necessário;
- Interface de administração: gera facilmente interfaces para inserção e modificação de dados (CRUD);
- URLs fáceis e elegantes: usando-se o recurso de reescrita de URLs facilita o acesso às aplicações;
- Sistema de templates: possui uma linguagem de template para separar a lógica de programação do design;
- Sistema de cache: aumento de performance pois as páginas são geradas uma vez e acessadas do cache.

- Internacionalização: possui suporte a desenvolver aplicações de forma a facilitar o processo de internacionalização.

3.4. Arquitetura REST e RESTful Services APIs

REST consiste em um conjunto coordenado de restrições arquiteturais aplicadas a componentes, conectores e elementos de dados dentro de um sistema de hipermídia distribuído [14]. O termo foi definido no ano 2000 por Roy Fielding e é a sigla para Representational State Transfer. A aplicação mais comum de REST é a própria World Wide Web, que utilizou REST como base para o desenvolvimento do HTTP 1.1.

A arquitetura REST aplica várias restrições para seus componentes e elementos de dados. Algumas destas restrições são:

- Client-Server: separar a arquitetura em cliente e servidor com o objetivo de distinguir as tarefas de cada componente para que o desenvolvimento seja independente um do outro. Assim, o cliente (consumidor do serviço) não se preocupa com tarefas como comunicação com banco de dados, gerenciamento de cache, log, etc. E o contrário também é válido, o servidor (provedor do serviço) não se preocupa com tarefas como interface, experiência do usuário, etc. Permitindo a evolução independente das duas arquiteturas;
- Stateless: qualquer requisição que seja feita pelo cliente deve conter individualmente todas as informações necessárias para que o servidor possa processar o serviço requisitado adequadamente. Portanto, o cliente deve guardar as informações necessárias para guardar o estado atual, não o servidor;
- Cacheable: como as requisições são independentes uma das outras, é possível que haja requisições do mesmo tipo, aumentando o processamento desnecessariamente. Para não afetar o desempenho, é preciso que as informações possam ser guardadas em cachê;
- Layered System: arquitetura em camadas para que seja possível a adição ou a exclusão de uma camada sem afetar as outras.

Aplicando as restrições de arquitetura para os padrões de *Web Services*, é necessário ter os seguintes requisitos:

- Base URI (Universal Resource Indicator) que seja única;
- Formato padrão para os tipos de dados que serão utilizados - nosso projeto será utilizado JSON;
- Padronizações dos métodos HTTP em GET, PUT, POST ou DELETE, por exemplo, sendo que cada um destes métodos tem uma função distinta.

Os sistemas que seguem os princípios REST são freqüentemente chamados de RESTful, como é o caso da API que foi implementada para a comunicação entre a central doméstica e o servidor Web.

4. ARQUITETURA DO SISTEMA SMART PLUG

Com base nos requisitos levantados e tendências futuras, construiu-se uma arquitetura composta de quatro camadas. Esse capítulo detalha a arquitetura lógica e física do sistema.

4.1. Requisitos do Sistema

Primeiramente vamos listar os requisitos funcionais:

- Sistema deve possibilitar o ligamento e desligamento de dispositivos remotamente;
- Sistema deve possibilitar a visualização do perfil (histórico) de consumo de energia elétrica do aparelho ligado à tomada do usuário;
- Sistema deve possibilitar a visualização do consumo atual do dispositivo acoplado à tomada;
- Agendamento de horário / programação do horário de funcionamento;
- Sistema deve detectar uma queda de energia e sua subsequente volta e só ligar o aparelho conectado à tomada após a estabilização da tensão de entrada;
- Sistema deve funcionar tanto em 110V como 220V.

Agora vamos listar os Requisitos não Funcionais:

- Tempo de resposta deve ser no máximo de 5s;
- Comunicação sem Fio do dispositivo com a central deve ser segura;
- Comunicação do usuário com o servidor WEB deve ser segura;
- Interface deve ser intuitiva ao usuário;
- Aparelho não deve permitir que o usuário sofra choques elétricos.

4.2. Arquitetura em 4 camadas

A tendência é de que no futuro, muitos aparelhos estejam conectados à internet - já em 2020 espera-se um total de 20 bilhões de dispositivos conectados à rede mundial [15]. No caso específico do projeto, é esperado que um usuário tenha várias tomadas inteligentes em sua casa, além de outros aparelhos. Desse modo, é natural dividir a arquitetura do sistema em quatro camadas (figura 10): a camada das Tomadas, a camada da central doméstica, a camada do servidor web e a camada do usuário.



Figura 10: Arquitetura de camadas do Sistema Smart Plug.

Essas camadas interagem da seguinte maneira:

- i. As tomadas enviam e recebem informações da central doméstica.
- ii. A central doméstica envia e recebe informações do servidor WEB.
- iii. O servidor WEB envia e recebe informações dos Usuários.

Desse modo todas as camadas estão conectadas e podem interagir. Entretanto, para o usuário enviar um comando de liga/desliga para a tomada é necessário que o comando seja enviado para o servidor web, que por sua vez enviará o comando para a central doméstica e essa finalmente enviará o comando para a tomada.

Abaixo vamos especificar como ocorrem essas interações.

- I. A comunicação das Tomadas com a central doméstica é realizada através da utilização de Shields Zigbee (portanto a comunicação é sem fio). A atualização de dados é feita de 3s em 3s.

- II. A comunicação da central doméstica com o servidor web é realizada através de API RESTful e HTTP. A cada 2s a central verifica junto ao servidor se há alguma pendência a ser resolvida.
- III. A comunicação do servidor web com o usuário é realizada através da internet utilizando HTTP. O usuário poderá acessar o sistema através de computadores ou dispositivos *mobile*.

4.2.1. Detalhes das camadas

A primeira camada – Tomadas – é formada por um conjunto de tomadas inteligentes, ou smart plugs, que são construídas a partir de diversos componentes de hardware (sensores, microcontrolador e shield zigbee). O capítulo 6 abordará com detalhes o desenvolvimento desses dispositivos.



Figura 11: Camada de Tomadas e seus componentes.

A segunda camada – Central Doméstica – consiste em um dispositivo que controla diretamente as tomadas, recebe as informações de consumo daquelas e as envia para o servidor web. A figura 12 mostra os componentes da central doméstica (Xbee explorer, módulo Xbee, Banco de dados e programs P1, P2, P3 e P4). Mais detalhes sobre a desenvolvimento da central doméstica será encontrado no capítulo 7.

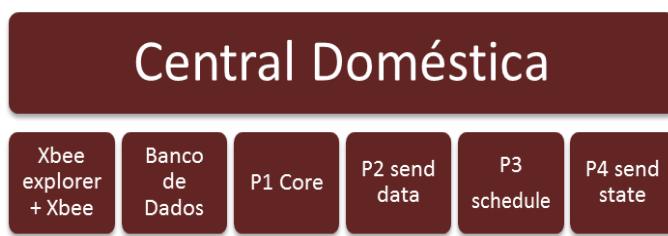


Figura 12: Arquitetura lógica da central Doméstica.

A Terceira camada – Servidor Web, ilustrada na Figura 13 – gera a interface com o usuário, possibilita o agendamento e a criação de gráficos de consumo, possibilita acionamento de tomadas, além de detectar se houve queda de comunicação entre o servidor e a central doméstica. Mais detalhes sobre o servidor serão encontrados no capítulo 8.



Figura 13: Arquitetura lógica do Servidor Web.

Quarta camada – Usuário, ilustrada na Figura 14 – Cada casa (central doméstica associada) pode ter vários usuários e cada usuário pode acessar o sistema Smart Plug através de computadores ou dispositivos móveis.



Figura 14: Formas pelas quais os usuários se conectam ao sistema Smart Plug.

4.2. Arquitetura Física

Como já descrito, a Smart Plug conta com sensores, um microcontrolador simples um módulo de comunicação sem fio, central doméstica e servidor web. A figura 15 mostra como esses componentes estão organizados fisicamente e como é realizada a comunicação entre os mesmos componentes.

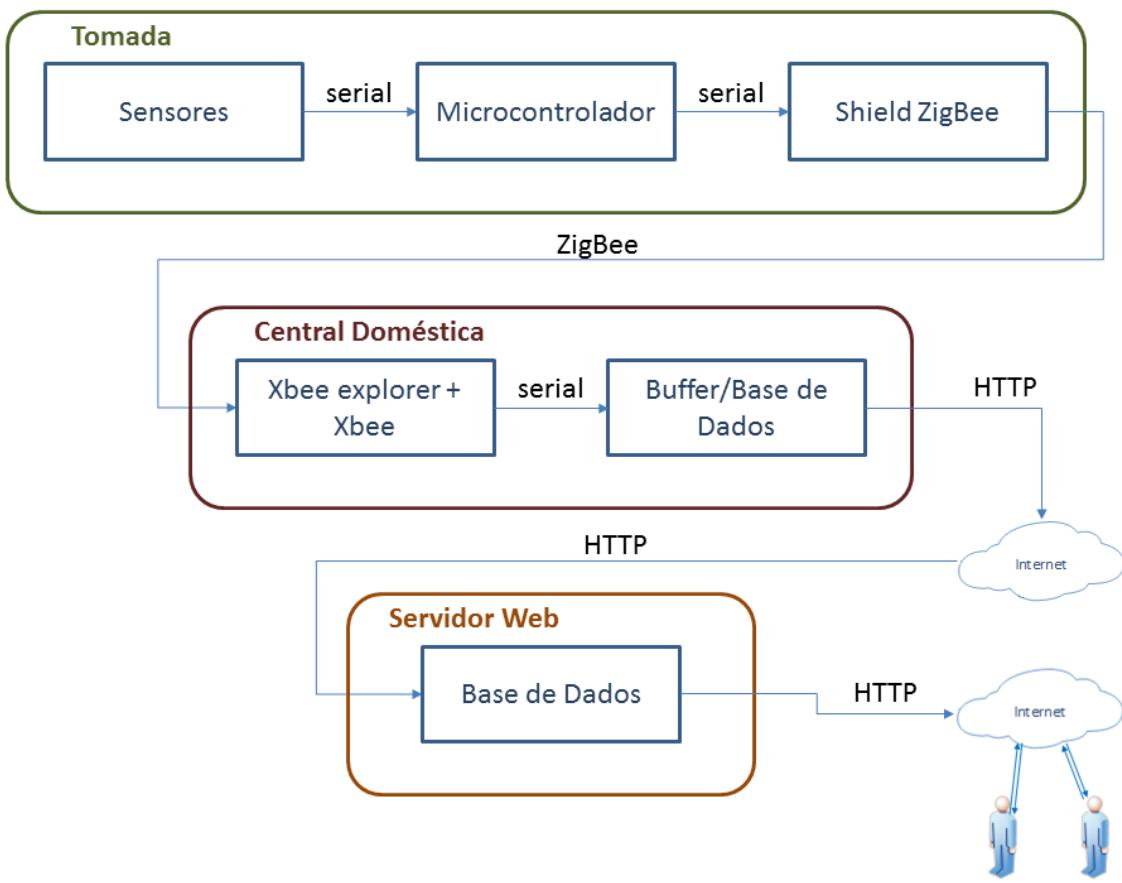


Figura 15: Arquitetura física do sistema.

Essa arquitetura possibilita uma otimização do sistema com a parte do hardware mais cara e com maior consumo de energia centralizada e a parte mais barata e com menor consumo distribuída em vários dispositivos.

A figura 16 mostra a arquitetura completa do sistema Smart Plug, nela pode-se observar que vários usuários têm acesso ao servidor web através de diversos dispositivos. Cada usuário está relacionado a uma central doméstica e essa central doméstica está relacionada a inúmeras tomadas inteligentes.

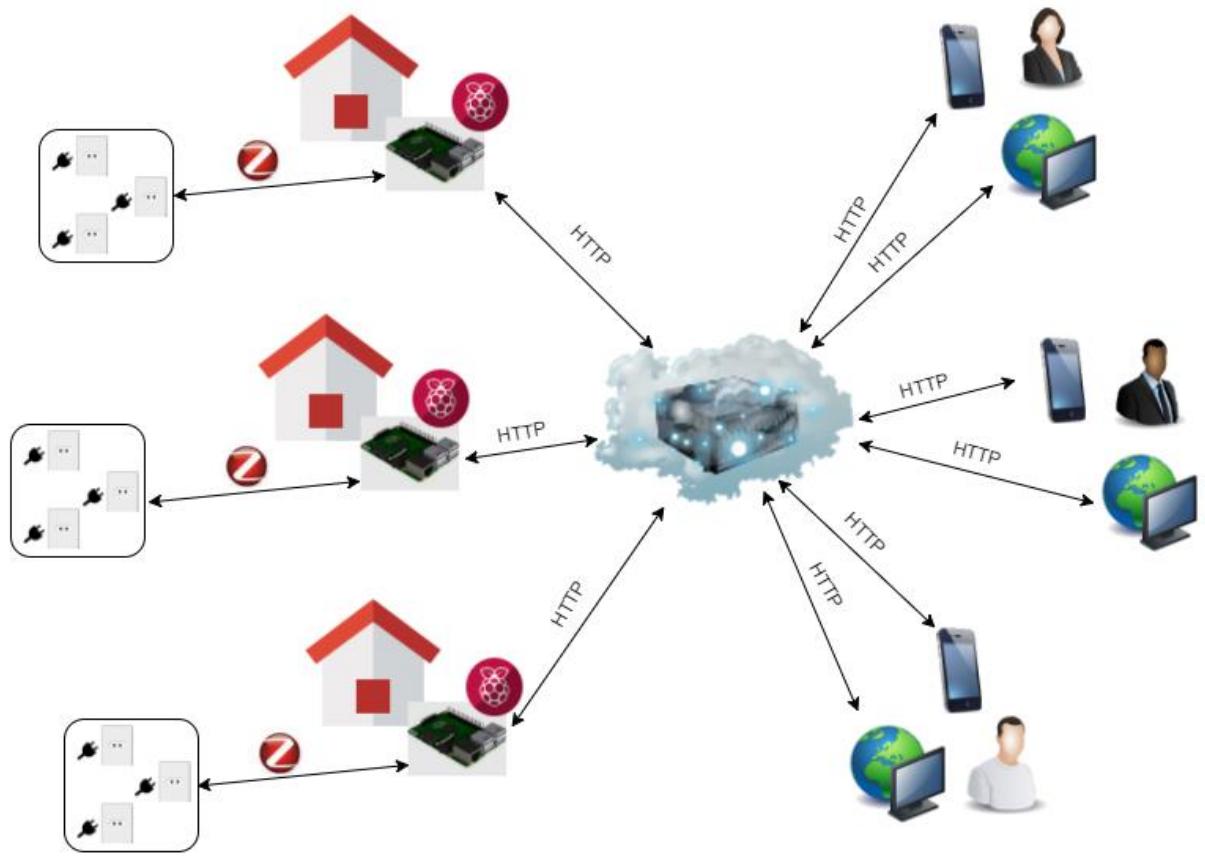


Figura 16: Arquitetura geral do Sistema Smart Plug.

5. PROJETO DE HARDWARE

Nesse capítulo será detalhado todos os componentes de hardware que foram utilizados no projeto da Smart Plug. Também será abordado como os circuitos foram criados e como o circuito impresso foi desenhado. No fim do capítulo será encontrada uma breve explicação do funcionamento do módulo de detecção e do software instalado no arduino.

5.1. Relé

O funcionamento do circuito é bastante simples, quando a saída do Arduino (IN) está em 5V o diodo no optoacoplador aciona o foto-transistor que passa a conduzir, assim o transistor mais à esquerda na figura 17 também conduz e aciona o relé.

Na construção do circuito do relé foi feito o isolamento entre as partes de baixa e alta tensão no circuito. Para isso usou-se um optoacoplador como pode ser visto na figura 17. O resistor de 470Ω na saída do Arduino limita a corrente no diodo interno do optoacoplador. O diodo do lado esquerdo do relé absorve sinais elétricos gerados pela bobina do relé quando essa liga ou desliga.

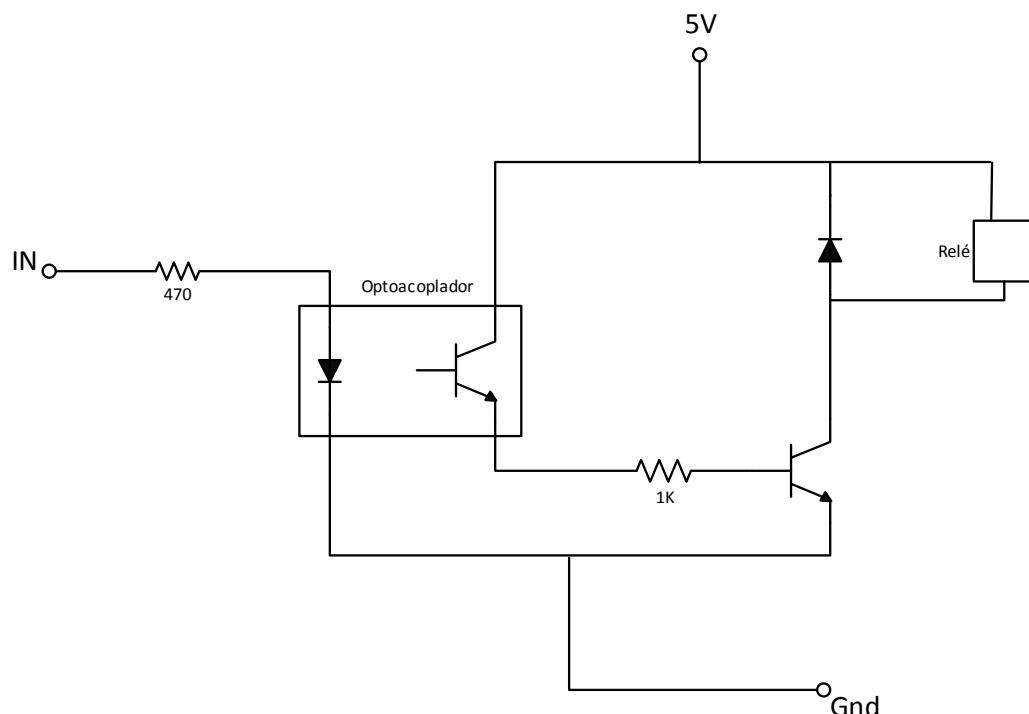


Figura 17: Circuito do relé.

5.2. Sensor de corrente

Para a medição de corrente usou-se um sensor baseado no efeito Hall.

O dispositivo ACS712 possui um caminho de cobre onde a corrente passa e gera um campo magnético, depois esse sinal é convertido num valor proporcional de tensão. O sinal de saída é analógico e a corrente zero ocorre para tensão de saída é $V_{cc}/2$.

O sensor pode ser utilizado tanto para corrente alternada como continua. Ele também conta com isolamento entre as duas partes do circuito.

A montagem feita foi bastante semelhante à sugerida pelo datasheet, a diferença é que foi colocado um LED para indicar que o sensor está ligado. Na figura 18 é possível ver o diagrama com o circuito.

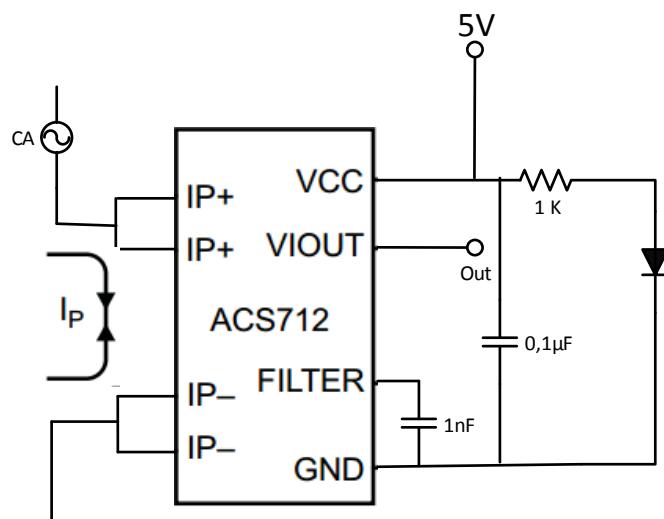


Figura 18: Sensor de corrente e circuito [F6].

O sensor ACS712 é linear, como pode ser visto na figura 19, por isso não foi necessário o levantamento da curva de funcionamento. Para uma corrente igual a zero e tensão de alimentação 5V o sensor produz uma saída de 2,5V. E nos extremos produz 0,5V para uma corrente de -20A e 4,5V para uma corrente de 20A.

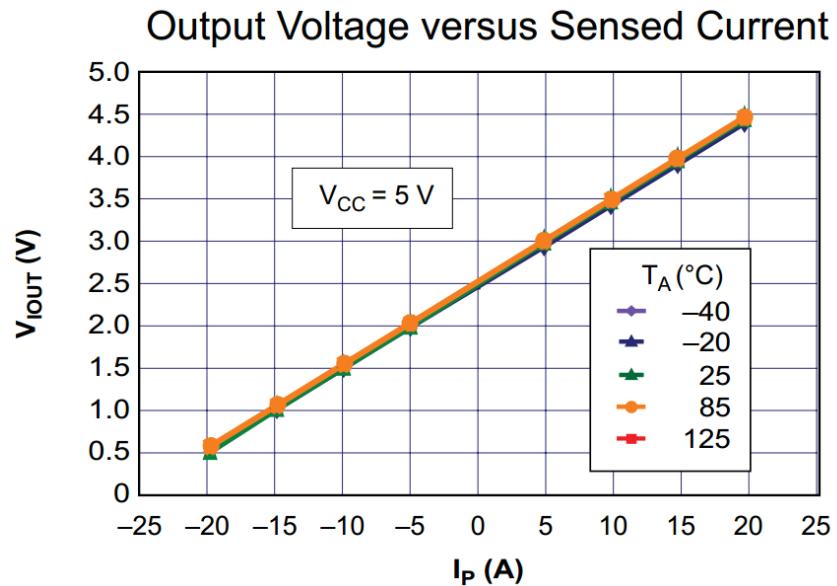


Figura 19: Comportamento da tensão de saída dada a corrente de entrada.

Com esse sensor é possível calcular o valor eficaz real (true rms) da corrente. Para isso foi feito um programa seguindo a lógica descrita na Figura 20.

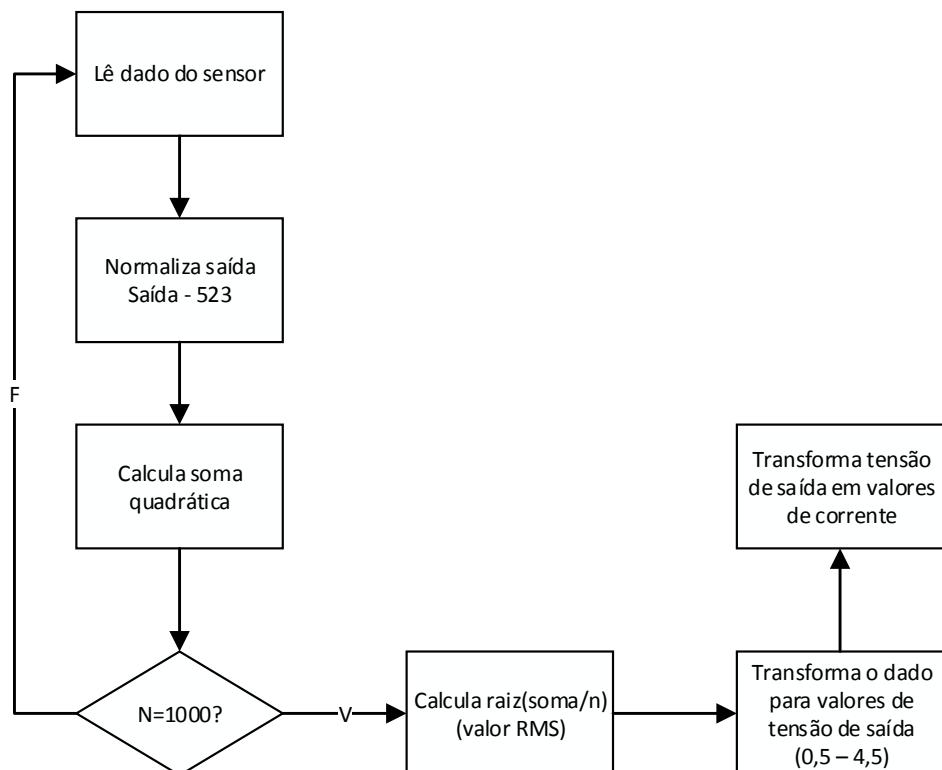


Figura 20: Lógica de funcionamento do programa que lê a corrente.

Como pode ser observado na figura 19, primeiro o programa faz a leitura do sensor. O sensor gera como resultado uma tensão que varia de 0,5V a 4,5V, essa tensão é colocada na porta analógica de um Arduíno que converte ela para um valor digital em 10 bits, gerando assim 1024 níveis (de 0 até 1023). Como o sensor mede correntes positivas e negativas, o zero dele se dá para uma tensão de saída de Vcc/2 (2,5V). Assim é necessário normalizar a saída subtraindo-se 523 (número gerado pelo zero na prática). Em seguida é feita a soma quadrática dos valores que vão sendo lidos do sensor, quando são feitas 1000 medidas é calculada a raiz da soma sobre o número de amostras (valor RMS):

$$I_{rms} = \sqrt{\frac{1}{T} \int_0^T i^2(t) dt}$$

Esse valor, porém, se encontra na forma de nível. Para transforma-lo em valor de tensão de saída correspondente ao sensor deve-se multiplicá-lo por 0.0048828125, que é o resultado de 5/1024 e representa o número de volts por divisão na porta do Arduino. Por fim, o valor em volts deve ser convertido na corrente medida pelo sensor, como pode ser visto no gráfico anterior, uma diferença de tensão de 2V (2,5V até 4,5V por exemplo) corresponde a uma diferença de corrente de 20A. Dessa forma deve-se dividir o resultado anterior por 2/20 para se obter o valor de corrente.

5.3. Sensor de Tensão

O sensor de tensão escolhido é bastante simples e barato. A tensão de entrada passa por um optoacoplador em serie com um resistor de 220KΩ. O led infravermelho do optoacoplador emite luz com intensidade proporcional a tensão de entrada e aciona um foto-transistor. Quanto mais polarizado o transistor maior a corrente que passa por ele e mais próximo de 5V fica a saída S. De forma análoga ao sensor de corrente, a porta do Arduino converte o sinal analógico de saída em um número de 10 bits. O circuito Sensor de Tensão está ilustrado na Figura 21.

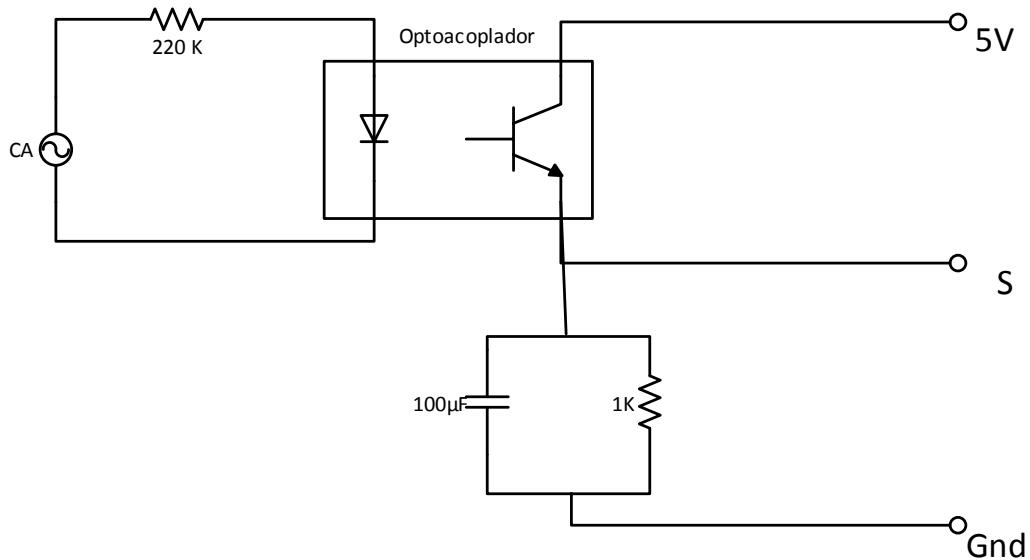


Figura 21: Circuito do sensor de tensão.

5.4. Interligação dos sensores

Os sensores de tensão e corrente e o relé estão interligados como mostra a figura 22. É importante ressaltar que o relé deve ser colocado após (mais próximo à saída) o sensor de tensão para que possa detectar se a tomada está energizada ou não independentemente da posição do relé. O relé deve estar ligado no normalmente aberto (NA). Dessa forma, caso a tomada inteligente pare de funcionar (se a bateria acabar por exemplo) o aparelho conectado à ela é desligado por questões de segurança.

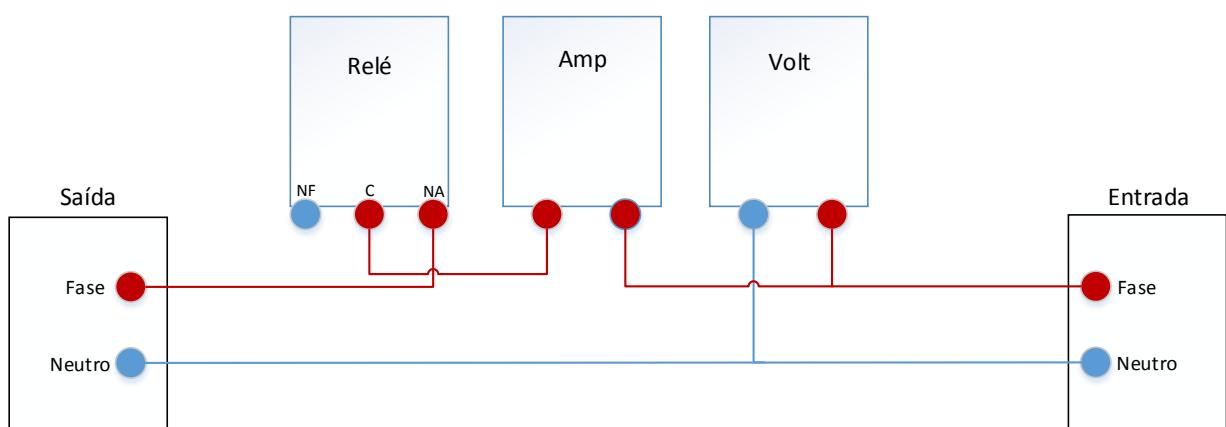


Figura 22: Componentes em blocos.

5.5. Circuito Impresso

Para o Projeto do circuito impresso foi usado o programa EAGLE 7.3.0 Light. Por questões de praticidade escolheu-se construir o circuito na forma de um Shild para Arduino. Isso é indicado pelo módulo ARDUINO_R3_SHIELD no diagrama abaixo.

Os demais circuitos foram criados como já foi detalhado anteriormente. A interligação dos sensores deve ser feita conforme indica a figura 23. Foram adicionados ainda dois Leds, o primeiro (esquerdo) indica se o circuito com o Arduino está alimentado ou não já o segundo informa se o rele está ativado (círculo fechado) ou desativado (círculo aberto).

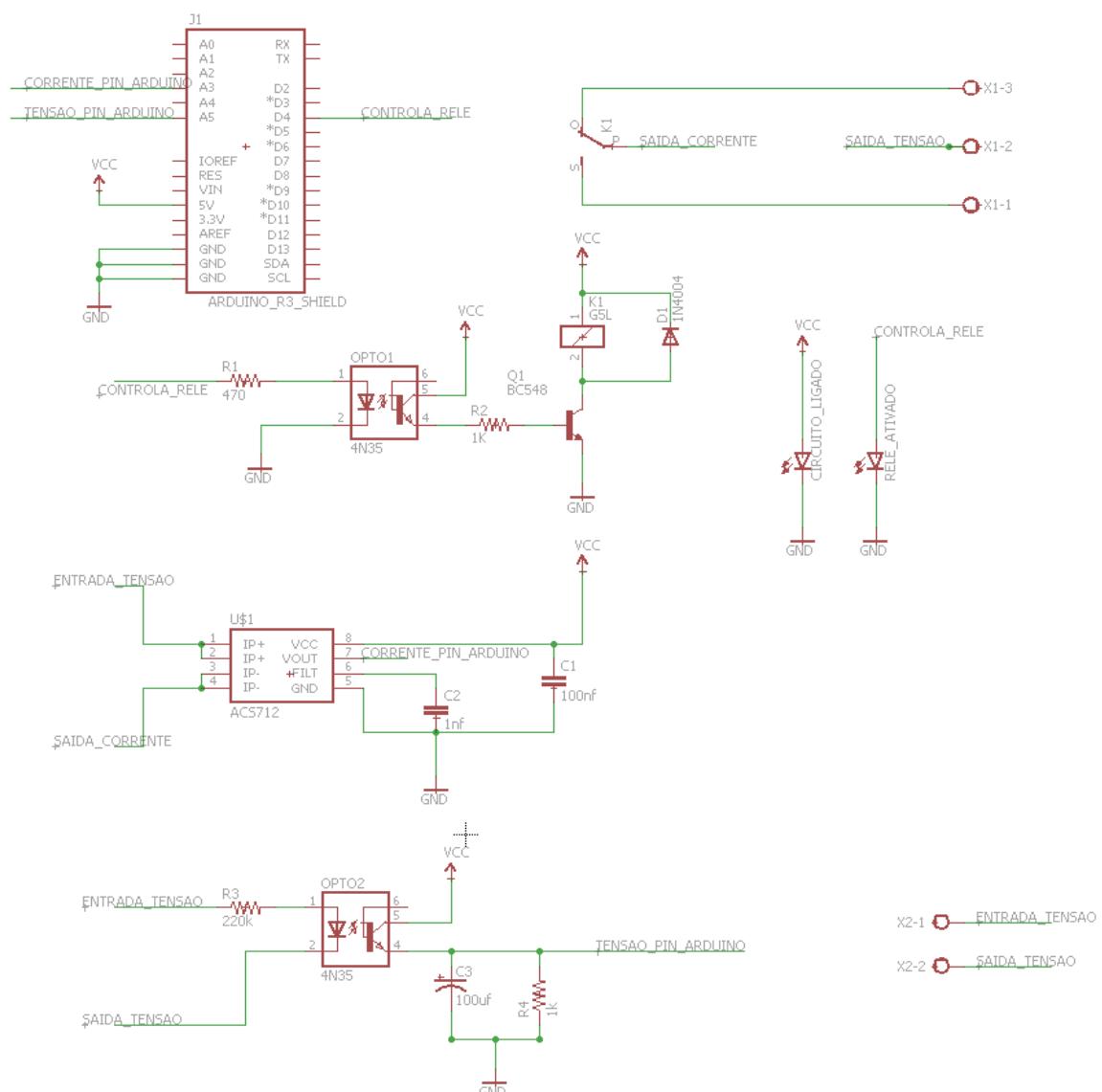


Figura 23: Diagrama do circuito elétrico global.

Após a construção do circuito na forma lógica acima, foi gerado o digrama com todos os componentes físicos e suas respectivas ligações. Fez-se necessário agrupar esses componentes em uma forma conveniente para então converter as ligações em trilhas.

As trilhas que irão operar na tensão e corrente da tomada devem ser mais grossas que as trilhas internas dos sensores. Assim foi feita a construção manual das trilhas mais grossas e de algumas trilhas finas. Após essa construção foi adicionada a geração automática de trilhas do programa para o preenchimento das demais trilhas, mas o programa não conseguiu realizar parte das trilhas sendo que a parcela que ele fez foi considerada insatisfatória. Dessa forma, todas as trilhas tiveram que ser feitas manualmente. O resultado pode ser visto na figura 24.

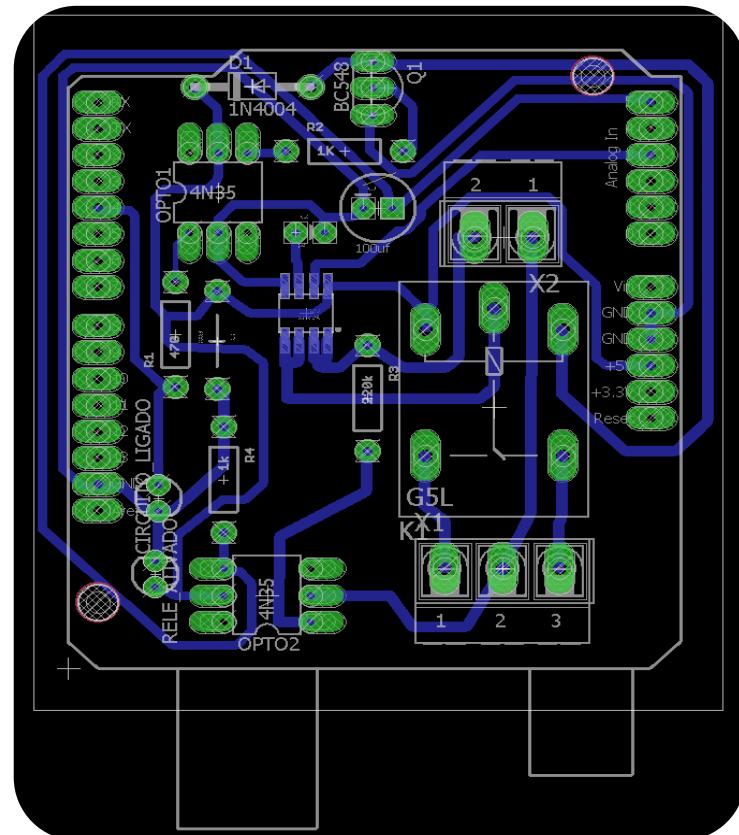


Figura 24: Circuito já com as trilhas.

Por fim, editou-se o circuito acima de modo que apenas as trilhas e os pads sobrassem. Então, foi gerada a imagem pelo próprio programa, que está mostrada na Figura 25. Essa imagem que foi impressa em uma folha de transparência que foi utilizada para o decalque do circuito.

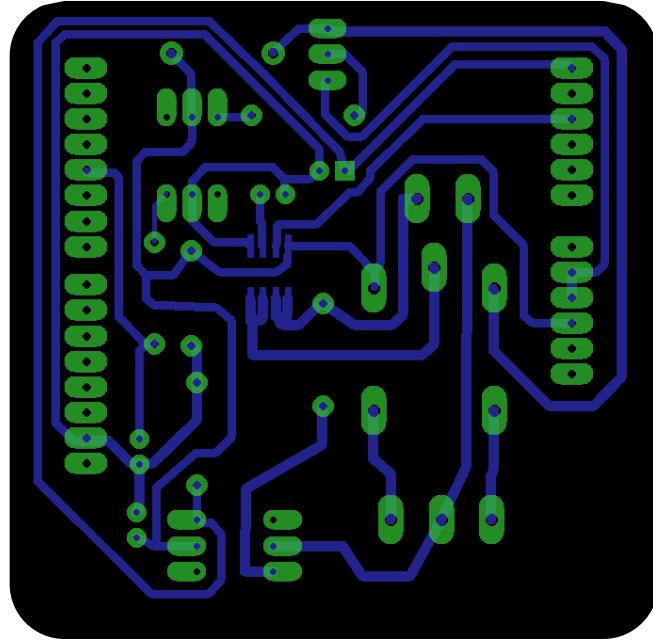


Figura 25: Resultado a ser impresso para montagem do circuito.

De posse do diagrama com os pads e as trilhas o circuito foi impresso em uma folha de transparência e depois transferido para uma placa de fenolite já previamente cortada no tamanho desejado. A placa foi corroída em percloroeto para que só restasse cobre nas partes onde havia trilhas e pads. Em seguida a placa foi perfurada com um perfurador manual, mas os pads de encaixe do shield ficaram desalinhados. Assim foi usada uma pequena furadeira para alinhar os furos. O resultado pode ser visto na figura 26. Nota-se que o circuito é espelhado.

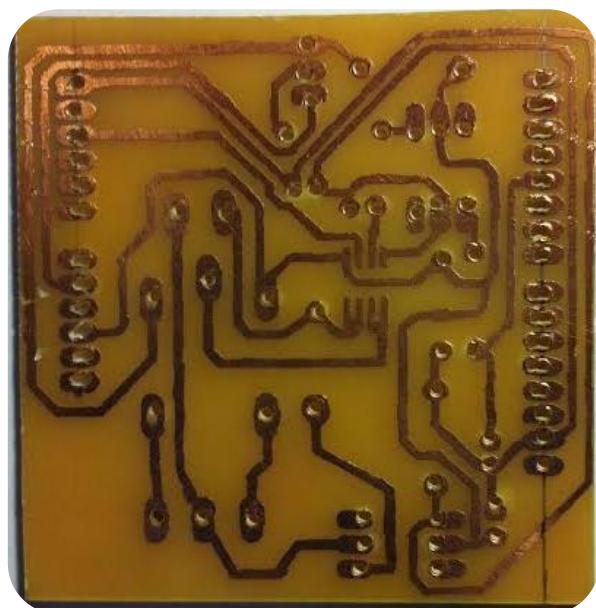


Figura 26: Resultado do circuito impresso.

De posse da placa já perfurada, deu-se prosseguimento para a soldagem dos componentes. A soldagem foi feita pelo próprio grupo com exceção do sensor de corrente ACS712 que é um componente smd (superficial monting device) e não é de fácil soldagem. O resultado da soldagem pode ser visto na figura 27. No canto superior esquerdo é possível ver a face de baixo do circuito, no canto superior direito tem-se a vista de cima, já na parte inferior ve-se a lateral.

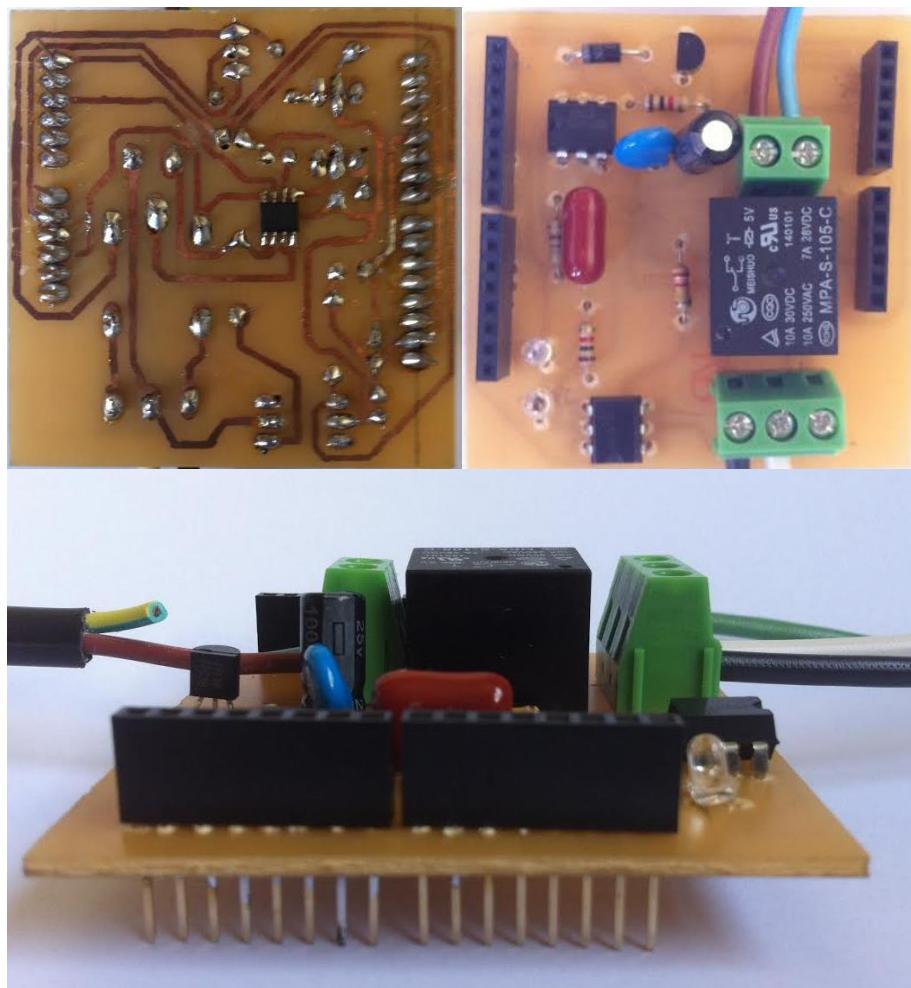


Figura 27: Diferentes vistas do shield.

Depois de soldados os componentes o shield foi acoplado ao Arduino para a realização dos testes. O resultado pode ser visto na figura 28.

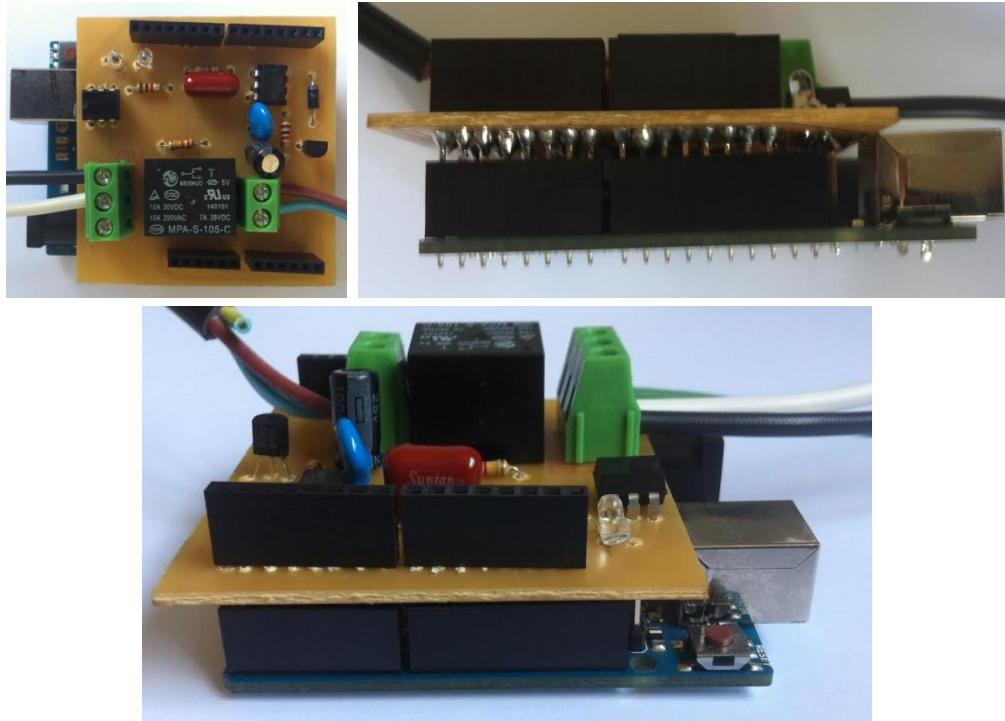


Figura 28: Diferentes vistas do shield integrado com Arduino.

5.6. Demais Componentes de Hardware

Além da placa do circuito impresso que contém os sensores de tensão e corrente e o relé, também foi preciso usar alguns componentes de hardware para a realização do projeto. Esses componentes estão listados a seguir.

5.6.1. Arduino

É uma plataforma open-source muito popular de prototipagem. Ele usa um microcontrolador Atmel AVR facilmente programável em uma linguagem padrão, semelhante ao C/C++. Ele conta com 14 pinos de entrada/saída digitais e 6 pinos de entrada analógica. No projeto são usados 1 pino digital para o sinal que é enviado ao relé, 2 pinos analógicos para o recebimento dos dados de tensão e corrente dos sensores.



Figura 29: Arduino Uno [F7].

5.6.2. Xbee Explorer

Esse módulo foi um grande facilitador do projeto, isso porque ele simplifica a comunicação sem fio entre a estação central e a tomada. Há um componente nessa placa que faz com que o computador a “veja” como uma porta serial (como ocorre quando a conexão é feita diretamente por um cabo). Assim, a união do Xbee explorer com os módulos Xbee cria um cabo virtual e todo procedimento pode ser feito de maneira análoga ao caso em que há um cabo físico ligando os componentes.



Figura 30: Módulo Xbee explorer usado (Sparkfun) [F8].

5.6.3. Xbee

Módulo responsável pela comunicação sem fio. Foi escolhido para o projeto o módulo que funciona usando o protocolo Zigbee. Esse protocolo foi desenvolvido tendo-se como foco a economia de energia e é amplamente utilizado em redes de sensores. No caso do projeto desenvolvido, a comunicação ocorre nos dois sentidos Arduino/PC e PC/Arduino. Os módulos Xbee contam ainda com criptografia AES de 128bits. Eles são facilmente programáveis por meio do Xbee explorer, basta fazer o download do programa fornecido pela Digital (empresa que faz o xbee) e acoplar o

Xbee no Xbee explorer. Com uma potência de 1mW o alcance em ambientes internos é de 30 metros, o que é suficiente para o padrão de moradia da população, em ambientes externos o alcance aumenta para 100 metros.



Figura 31: Módulo Xbee [F9].

5.6.4. Integração dos componentes na tomada

A placa com o conjunto de sensores envia o dado para o arduino que processa essa informação e envia para a central por meio do Xbee. Para que o Xbee se encaixe no arduino foi necessário usar um shield, que aparece na parte superior da figura 32. O dado de ativação e desativação do relé proveniente da central chega na tomada pelo Xbee, o arduino processa informação e envia para o relé.

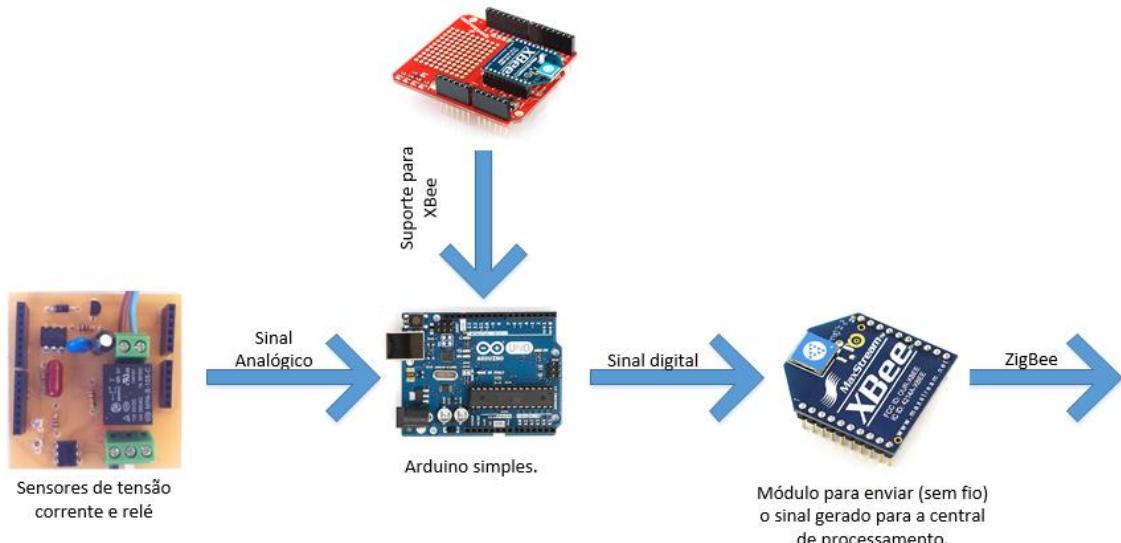


Figura 32: Integração dos componentes na tomada – Todas as figuras utilizadas nessa composição, já foram referenciadas anteriormente.

5.6.5. Integração dos componentes na central

Compondo o hardware da central doméstica, existem três elementos. Primeiro tem-se um computador, foi usado um notebook, mas poderia ser um Raspberry Pi por exemplo, que é responsável pelo processamento dos dados. Tem-se também o módulo Xbee responsável pela comunicação sem fio com a tomada. E por fim tem-se o módulo Xbee explorer que faz a ponte de comunicação entre o Xbee e a central, como podemos ver na Figura 33.



Figura 33: Integração dos componentes na central doméstica – Todas as figuras utilizadas nessa composição, já foram referenciadas anteriormente.

5.7. Detector de queda

Um problema bastante comum, principalmente no Brasil, é a queima de aparelhos elétricos após um apagão ou queda de energia. A queima geralmente ocorre não com a queda brusca da tensão, mas sim com o reestabelecimento da mesma. Geralmente, quando a tensão retorna, ela atinge um valor mais alto do que o valor correto e essa sobretensão danifica o aparelho. Na figura 34 é possível ver essa situação, primeiramente tem-se uma tensão normal na rede de amplitude “a”, em seguida ocorre a queda de energia. No retorno da tensão percebe-se um valor de amplitude “A” bem maior que a tensão normal da rede “a”.

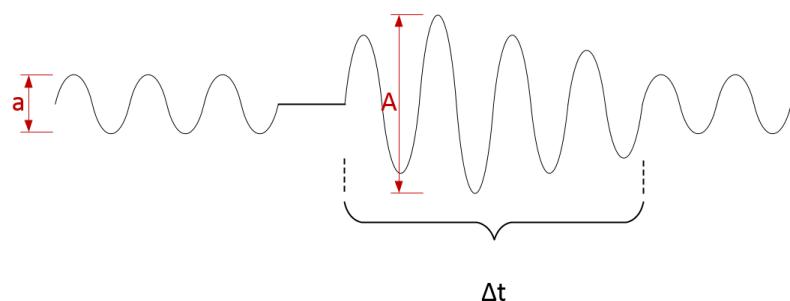


Figura 34: Pico de Tensão pós religamento.

Nota-se também na Figura 34 que após um intervalo de tempo Δt a tensão retorna para o seu valor normal de operação. O que a tomada inteligente se propõe a fazer nessa situação para impedir a queima do aparelho segue os seguintes passos:

- Detecta a queda de energia.
- Abre o circuito do aparelho que esteja ligado a ela.
- Detecta o retorno da energia elétrica.
- Espera um intervalo Δt para que a tensão estabilize.
- Fecha o circuito, possibilitando o funcionamento do aparelho.

Dessa forma, o que o aparelho ligado à tomada “enxerga” é a situação descrita na Figura 35. Uma queda de tensão e sua posterior volta já em um nível que não danifica o aparelho.

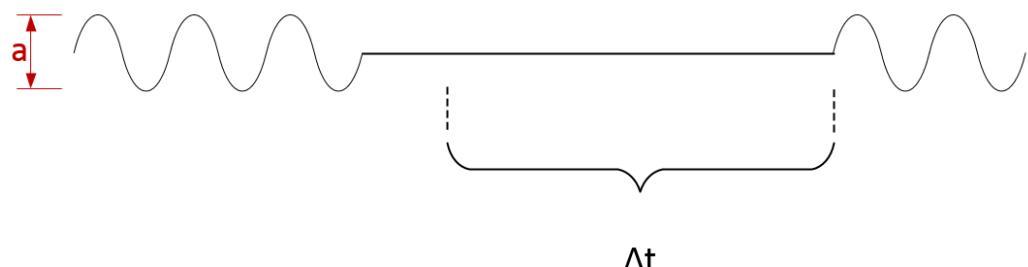


Figura 35: Onda sem pico de tensão.

5.8. Software da Tomada (Arduino)

Como já mencionado anteriormente, tentou-se centralizar ao máximo o processamento do sistema na central doméstica, porém o cálculo da corrente eficaz foi feito na própria tomada porque ele exige a medida de várias correntes instantâneas, e transmitir todas essas correntes instantâneas para a central geraria um overhead na rede. O mesmo serve para a medida de tensão.

Para a medida dos valores de corrente e tensão usa-se o conversor analógico-digital do arduino. A Frequência padrão do conversor é de 10 kHz, como um loop for com 10.000 iterações resulta em um tempo de amostragem de $\frac{10000}{10}$ kHz = 1 s e um período de rede elétrica dura $\frac{1}{60}$ Hz = 0,01667 s tem-se então que uma amostragem contabiliza 60 ciclos da fase elétrica ($\frac{1}{0,01667}$).

Na Figura 36 é possível ver o funcionamento do software que roda no arduino. Ele consiste basicamente de um programa principal e uma interrupção configurada para ocorrer a uma frequência de 10 Hz.

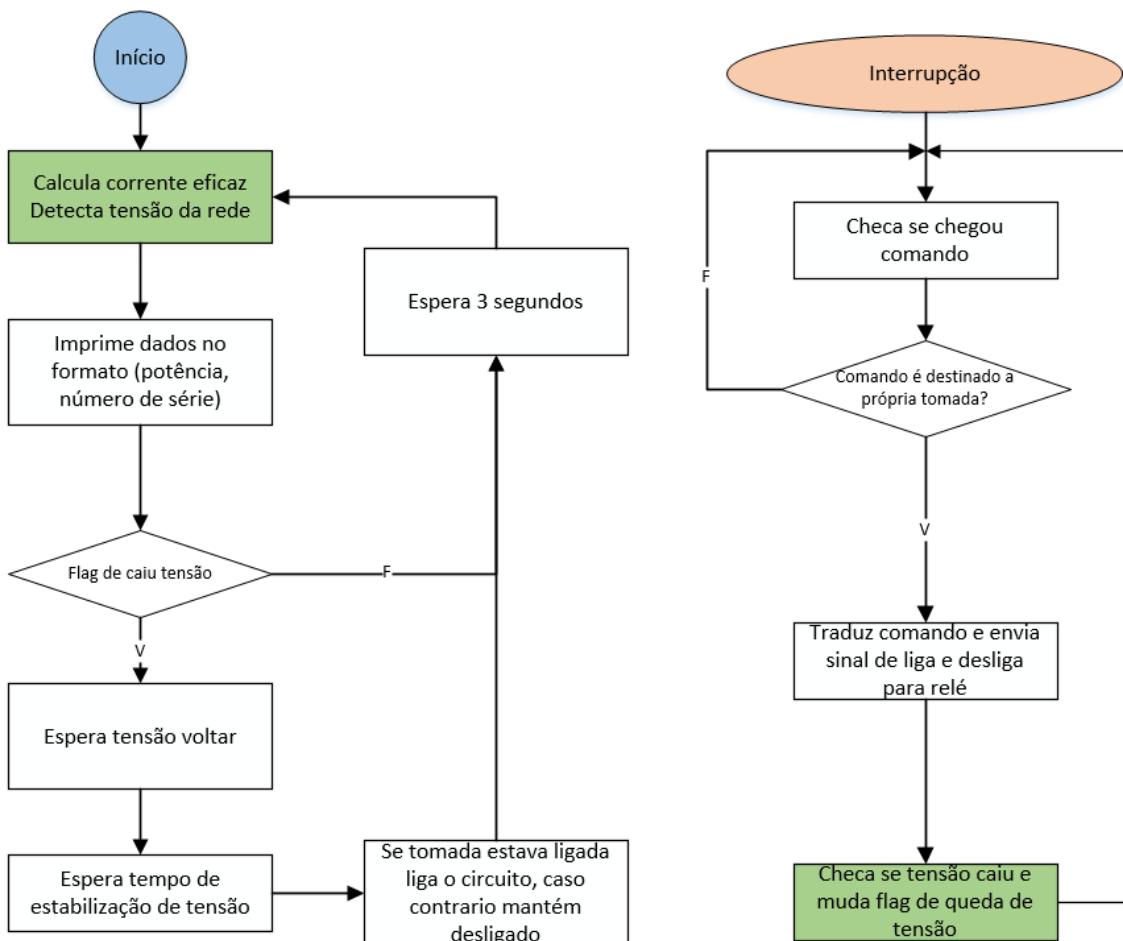


Figura 36: Funcionamento do software instalado no arduino.

O Programa principal calcula a tensão e a corrente, imprime os dados (envia para central doméstica) e checa se a flag indicativa de queda de tensão está ativada. Se a flag está ativada é porque ocorreu uma queda de tensão e, nesse caso, o programa espera a tensão voltar para, em seguida, esperar um tempo de estabilização. Se não houve queda, o programa retorna para o seu ciclo normal de operação e espera 3 segundos para a amostragem do próximo valor de tensão/corrente.

A interrupção, por sua vez, é responsável por checar se há algum sinal proveniente da central doméstica informando o ligamento e desligamento da tomada. Obviamente este sinal só é processado se for destinado a própria tomada (isso é verificado através do número de série). A interrupção também checa se houve queda de tensão e em caso positivo muda a flag de tensão. Nota-se que a

checagem de queda de tensão (última etapa da interrupção) não pode ser feita caso o programa principal esteja calculando a tensão e corrente eficazes, isso porque uma leitura analógica leva um tempo suficiente para quebrar a cadência de amostras e alterar o sinal de saída.

6. CENTRAL DOMÉSTICA

Nesse capítulo será detalhado como foi projetado o software da central doméstica. Esse software é responsável pela comunicação entre a tomada inteligente e o servidor web.

6.1. Requisitos funcionais do Software da central doméstica

Foram levantados os seguintes requisitos funcionais para a central doméstica:

- 1) A central doméstica deve fazer requisições ao servidor web de 2s em 2s para verificar se existe alguma atividade a ser executada;
- 2) A central doméstica deve mandar sinais de On/Off para o Arduino, esses sinais são responsáveis pelo acionamento e desligamento das tomadas;
- 3) A central doméstica deve receber dados dos sensores e armazená-los temporariamente;
- 4) A central doméstica deve receber o aviso de alterações no estado das tomadas;
- 5) A central doméstica deve ser capaz de se comunicar com o Software sevidor.
 - Receber comandos e agendamentos;
 - Enviar informações de estado da tomada;
 - Enviar informações de consumo.

6.2. Arquitetura lógica da central doméstica

Após a definição dos requisitos da central doméstica foi desenhado um esboço da arquitetura lógica da mesma. Esse esboço pode ser visto na Figura 37. O servidor web receberá os dados de consumo e estado da tomada e enviará comandos de liga e desliga, além de agendamentos.

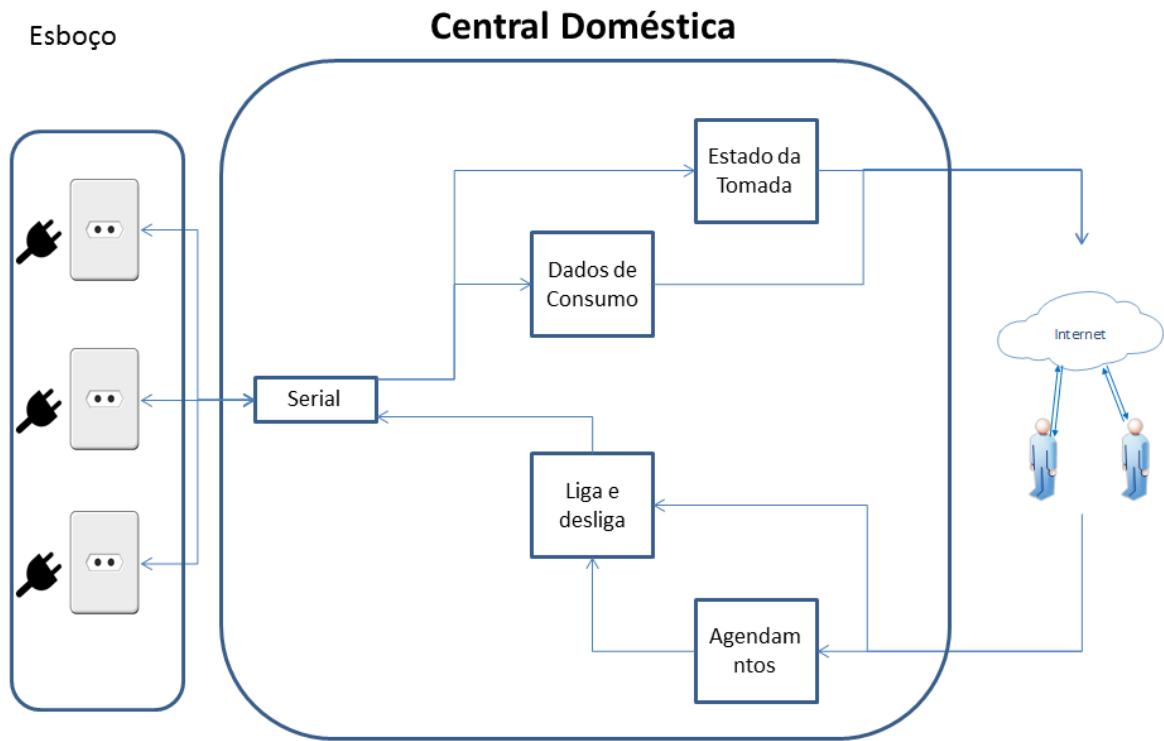


Figura 37: Esboço da arquitetura lógica da central doméstica.

6.3. Composição da central doméstica

A central doméstica é composta por um computador com sistema operacional Linux. Para esse trabalho o grupo utilizou um notebook, mas um Raspberry Pi é a solução que o grupo acredita ser mais adequada para o desenvolvimento de um protótipo comercial.

O Raspberry Pi é um computador do tamanho de um cartão de crédito, que se conecta a um monitor de computador ou TV, e usa um teclado e um mouse padrão, desenvolvido no Reino Unido pela Fundação Raspberry Pi. Todo o hardware é integrado numa única placa [16].

Foram desenvolvidos 4 programas para compor a central doméstica. A seguir uma breve descrição de cada programa:

- **p1_core**: Recebe os dados de consumo das tomadas e as alterações de estado das mesmas. Faz as requisições junto ao servidor web para verificar se há alguma tarefa para se executada;

- **p2_send_data:** Envia para o servidor web os dados de consumo das tomadas;
- **p3_schedule:** Executa a ação previamente cadastrada de ligar e desligar uma tomada;
- **p4_send_state:** Envia para o servidor web a alteração de estado de uma tomada.

6.4. Conceitos importantes

Para o melhor entendimento do tópico 7.5 é necessário entender dois conceitos simples:

JSON - JavaScript Object Notation, é uma formatação leve de troca de dados. Para seres humanos, é fácil de ler e escrever. Para máquinas, é fácil de interpretar e gerar. JSON é em formato texto e completamente independente de linguagem, pois usa convenções que são familiares às linguagens C e familiares, incluindo Python [17]. Estas propriedades fazem com que JSON seja um formato ideal de troca de dados. Abaixo um exemplo:

```
{
  /*propriedade: valor, */
  projeto: "Smart Plug",
  site: "www.mysmartplug.com.br"
}
```

CRON - pode ser interpretado como um serviço do Linux que é carregado durante o processo de boot do sistema. Trata-se de uma ferramenta que permite programar a execução de comandos e processos de maneira repetitiva ou apenas uma única vez [18].

O horário Unix *Epoch* é definido como o número de segundos passados desde as 00:00:00 do dia a 1 de janeiro de 1970 do Tempo Universal Coordenado (UTC) no calendário gregoriano. Apesar de não considerar segundos bissextos, é largamente utilizado em sistemas operacionais do tipo Unix bem como em outros sistemas.

6.5. Descrição dos Programas

Nesse tópico será detalhado cada um dos quatro programas que compõem a central doméstica.

6.5.1. Programa p1_core

O primeiro programa, chamado de **p1_core** é dividido em dois threads. Esses threads, chamados de “coleta_dados” e de “verifica_pendencias” funcionam paralelamente conforme pode ser observado na figura 38. O funcionamento de cada thread é explicado nos itens a seguir.

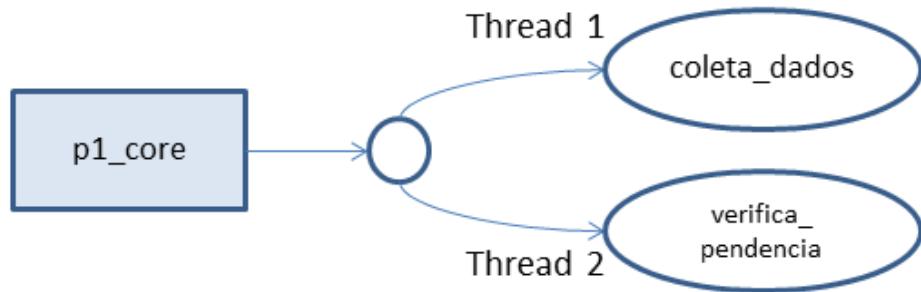


Figura 38: Divisão do programa 1 em duas threads.

I. Thread 1 – coleta_dados

O thread “coleta_dados” recebe os dados provenientes das tomadas que chegam à central doméstica pelo receptor conectado ao USB da mesma. Esses dados seguem o seguinte formato:

Potência Instantânea Identificador da tomada

ou

on/off Identificador da tomada

A rotina criada para o “coleta_dados” irá identificar se o dado recebido é um valor de potência ou um valor de estado. Se, por exemplo, chegar o dado: **125 001**, isso significa que uma potência instantânea de 125W está sendo consumida na tomada

cujo número de série é 001. Entretando se chegar o dado: **off 001** isso significa que ouve uma queda na tensão e o programa do arduino abriu o circuito (tomada desligada).

O processo completo do thread “coleta_dados” é mostrado na figura 39:

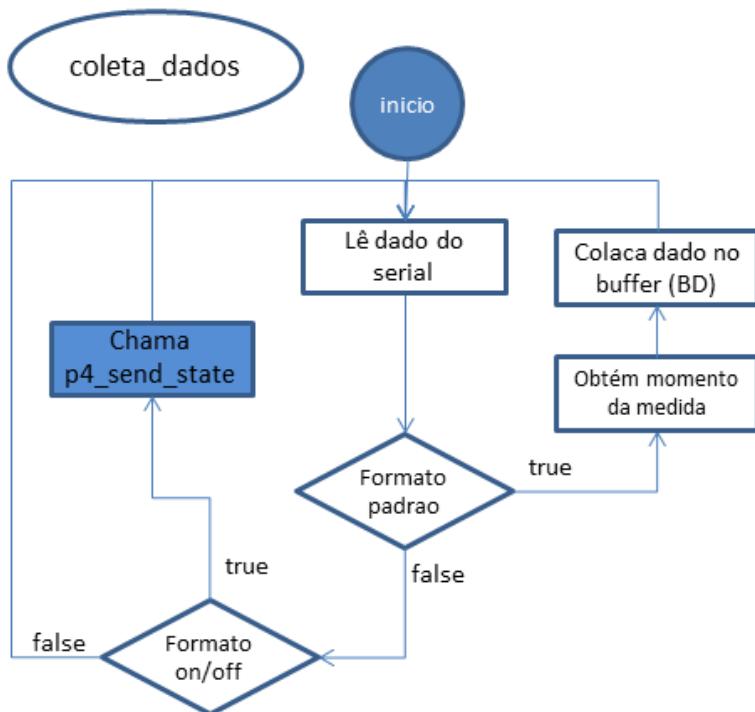


Figura 39: Lógica de funcionamento da thread coleta_dados.

Como pode ser visto na figura 39, caso seja identificado pela rotina que o dado que acabou de chegar é uma informação de potência, é obtido o momento que a informação chegou ao programa. O dado de potência, o número identificador da tomada e o *Epoch* são armazenados em um *buffer*. Para implementar esse *buffer* foi utilizado uma base de dados simples, o SQLite. Esse *buffer* será esvaziado pelo programa **p2_send_data**, mais detalhes no item 7.5.2.

Caso seja identificado pela rotina que o dado que acabou de chegar é uma informação de estado da tomada, o programa **p4_send_state** é chamado. Esse programa é responsável pelo envio do estado da tomada ao servidor web, mais detalhes no item 7.5.3.

II. Thread 2 – verifica_pendencias

O thread “verifica_pendencias” faz requisições ao servidor web a cada 2s. Através dessas requisições, ou da ausência das mesmas, o servidor poderá verificar uma possível queda de energia na residência (ou apenas uma queda no wi-fi da casa). Mais detalhes sobre essa verificação é encontrada no item 8.5..

A cada requisição será verificado se existe alguma pendência a ser resolvida. Existem quatro possíveis pendências:

- Enviar dados (send_data);
- Adicionar tomada (add_plug);
- Ligar ou desligar tomada (turn);
- Alterar agendamento (update_schedule).

A lógica de funcionamento do thread “verifica_pendencias” pode ser encontrada na figura 40.

Quando é verificada a existência da pendência “send_data” o programa **p2_send_data** é executado. Mais detalhes sobre esse programa podem ser encontrados no item 7.5.2. Sempre que um usuário estiver *logado* no site (servidor web) essa pendência aparecerá na lista, dessa forma o usuário recebe a informação do consumo atual da tomada.

Caso seja verificada a pendencia “add_plug” a central doméstica chama um método simples que faz o cadastro de uma nova tomada no SQLite. A seguir um exemplo do padrão JSON dos dados recebidos. A primeira propriedade, “action” é exatamente a pendência “add_pug” e a segunda propriedade é o número serial da nova tomada.

```
{
  "action": "add_plug",
  "number": 33
}
```

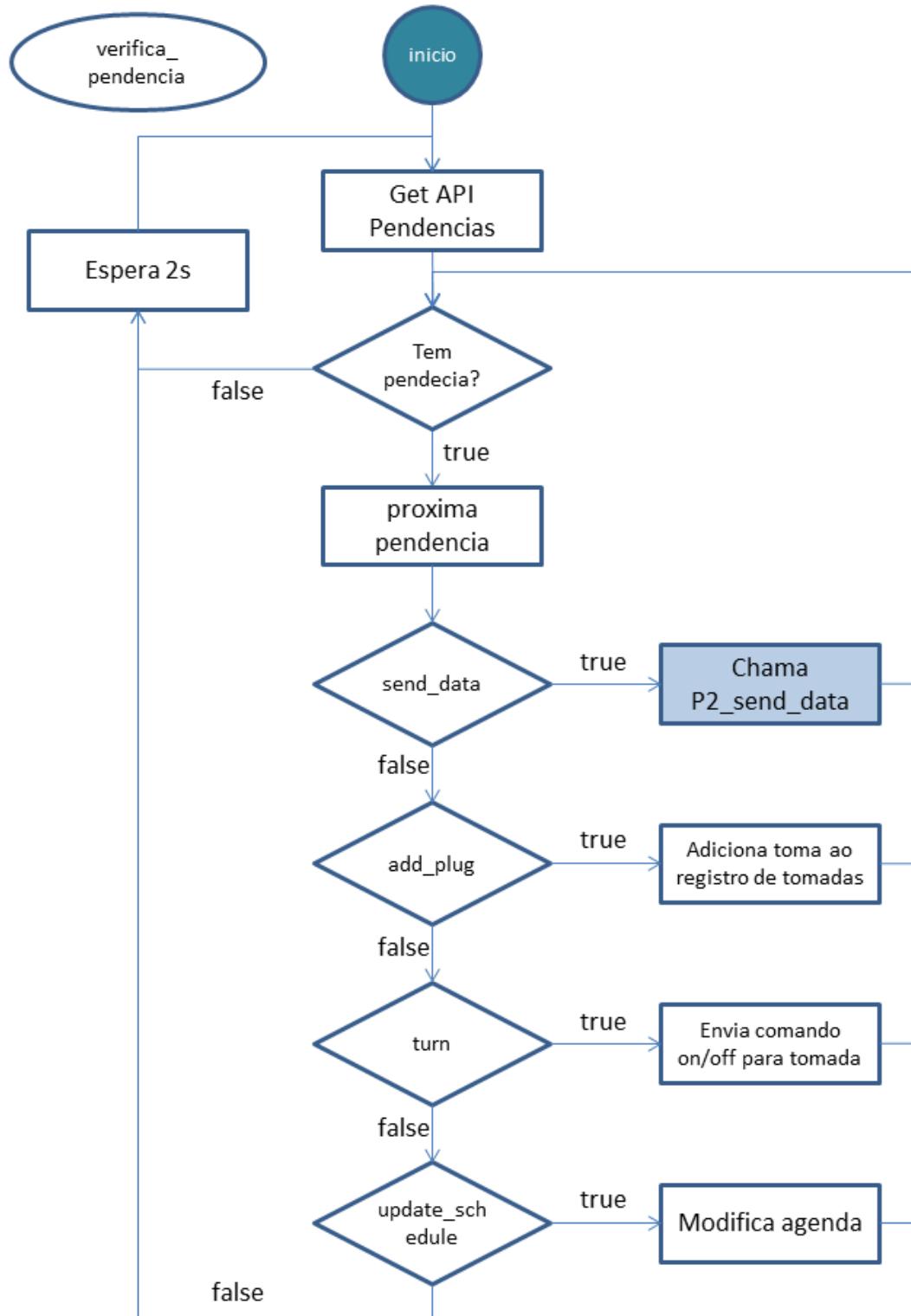


Figura 40: Lógica de funcionamento da thread `verifica_pendencias`.

Caso a pendência “turn” seja encontrada na lista de pendências, a central doméstica irá enviar o comando de liga ou desliga para a tomada específica. O padrão JSON recebido pela central doméstica é apresentado abaixo. A primeira propriedade, como das vezes anteriores, é a pendência. A segunda propriedade é a ação a ser executada (liga/desliga) e a última propriedade é o número de série da tomada

```
{
  "action":"turn",
  "turn":"on",
  "number":"55"
}
```

O comando de liga e desliga a ser enviado para tomada deve seguir um formato rígido da seguinte forma:

- Comando liga: “<número de série da tomada> :H*”
- Comando desliga: “<número de série da tomada> :L*”

Conhecendo o seu próprio número de série a tomada identificará quando um comando é direcionado a ela ou não. Logicamente, ao receber um comando com número de série que não seja o seu ela irá descarta-lo. O sinal de dois pontos “:” serve como separador, as letras “H” e “L” indicam o liga e desliga respectivamente e, por fim, o símbolo “*” marca o fim da cadeia.

Quando “update_schedule” é a pendência encontrada pelo thread é executado um método simples que faz a alteração dos agendamentos existentes. Esses agendamentos estão em outra tabela do SQLite. A seguir o padrão JSON recebido pela central doméstica. Da mesma forma que nos exemplos anteriores, a primeira propriedade é a pendência. A segunda propriedade é a ação a ser executada onde o primeiro número é o dia da semana, o segundo número a hora de execução da ação e, entre os colchetes finais, o comando agendado e o número de série da tomada

```
{
  "action":"update_schedule",
  "schedule":{
    "1 12:00":[
      [ "off", 2 ]
    ],  } }
```

6.5.2. Programa p2_send_data

O programa **p2_send_data** é responsável por retirar do *buffer* os dados de consumo das tomadas e enviar para o servidor web.

Esse programa pode ser executado de duas maneiras:

1. Diariamente, num horário pré-determinado;
2. Chamado pelo thread “verifica_pendencias” caso o usuário esteja logado no sistema.

No primeiro caso é usado o CRON para programar a execução do programa **p2_send_data** diariamente as 0h. Essa forma de implementar foi escolhida para diminuir o tráfego de dados, pois quando o usuário da residência não está logado no sistema não é necessário que os dados de consumo sejam enviados para o servidor.

A segunda possibilidade de execução do programa é exatamente quando o usuário está logado, pois ele deve receber a informação de consumo mais atual. O servidor irá solicitar os dados de consumo junto à central doméstica a cada 3s (igual ao tempo de coleta de dados).

Para o envio de dados da central doméstica ao servidor web foi usado o padrão JSON abaixo, onde “plug” se refere ao número de série da tomada (previamente cadastrada no sistema), “pot” se refere à potência instantânea medida e “collected_at” se refere ao *Epoch* em que o dado foi registrado.

```
[
  {
    "plug":8,
    "pot":1.2,
    "collected_at":"2015-11-08T18:15:45.356592"
  },
  {
    "plug":9,
    "pot":4.5,
    "collected_at":"2015-11-08T18:15:45.397283"
  }
]
```

Após o envio dos dados o *buffer* é limpo, pois não há necessidade de manter essa informação na central doméstica. No entanto, se o *buffer* contiver dados de tomadas que ainda não estão cadastradas no servidor web esses dados são mantidos na central doméstica até que o usuário cadastre as tomadas no servidor. O grupo tomou essa decisão para não perder o histórico de consumo da tomada.

6.5.3. Programa p3_schedule

O programa **p3_schedule** é acionado pelo CRON a cada minuto. Esse programa verifica se existe alguma atividade programada (ligar e desligar tomada) e a executa.

6.5.4. Programa p4_send_state

O programa **p4_send_data** é responsável por enviar o estado da tomada para o servidor web. Esse programa pode ser chamado em duas situações: quando há queda de energia na tomada ou quando algum agendamento foi executado. A seguir o padrão JSON enviado ao servidor.

```
{  
  "turn": "on",  
  "plug_number": "1"  
}
```

7. SOFTWARE WEB

O software web é aquele responsável pela interface usuário/sistema, comunicação com central doméstica e por todas as outras funcionalidades do sistema. Através dele que o usuário pode alterar o estado dos equipamentos (ligado/desligado) e acessar as informações armazenadas. Para criar este software foi utilizada a linguagem Python com o framework Django.

Nesse capítulo serão encontrados detalhes do projeto do software web.

7.1. Requisitos do software web

A seguir temos os requisitos levantados para o desenvolvimento do software web. No apêndice A encontra-se a definição completa dos requisitos.

- 1) Permitir o login de vários usuários.
- 2) Cadastro de usuários (ligados a uma central doméstica).
- 3) Permitir o cadastro de várias centrais domésticas (casas).
- 4) Permitir o usuário visualizar o estado das tomadas.
- 5) Permitir o usuário ligar e desligar tomadas.
- 6) Permitir o usuário agendar ligar/desligar.
- 7) Armazenar dados de consumo.
- 8) Permitir ao usuário visualizar os dados de consumo.
- 9) Elaborar graficos kWh X Tempo.
- 10) Permitir ao usuário cadastrar/ alterar informações do aparelho ligado a Tomada.
- 11) Permitir ao usuário fazer comparações entre a informação cadastrada e o valor de consumo da tomada.
- 12) Software deve funcionar como um servidor Web.
- 13) Software deve funcionar em Browser (Chrome engine).
- 14) Software pode ter interface App (opcional).

7.2. Protótipo de navegação

Conforme os requisitos 12 e 13 a interface é baseada em formato web, com janelas e formulários. Os principais objetos manipulados pela interface serão mostrados através de abas: Tomadas, Comodos, Agendamento, Histórico e Adicionar Usuário. O wire frame das telas se encontra no Apêndice B. Na Figura 41 há um diagrama de navegação simplificado.

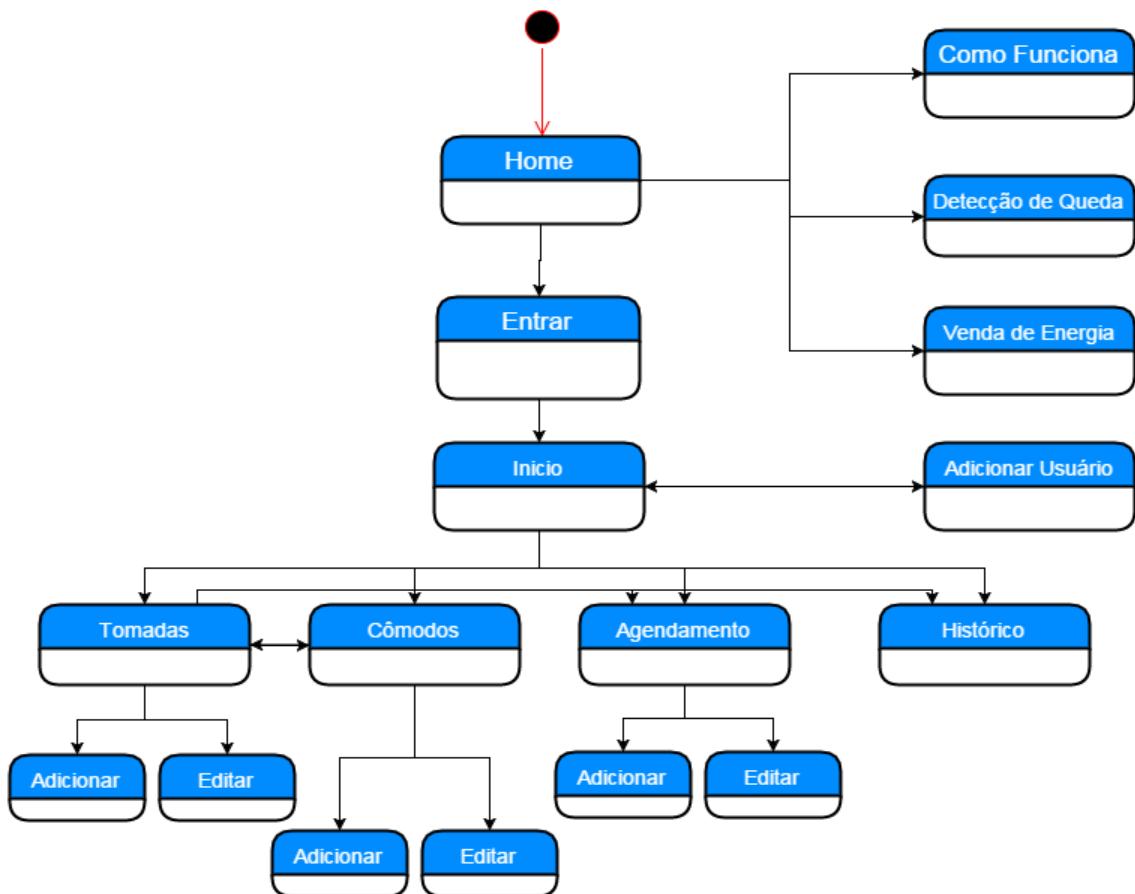


Figura 41: Diagrama de navegação simplificado

7.3. Diagrama de Classes

A Figura 42 apresenta o Diagrama de Classes do software web. Nessa figura podem ser observadas várias classes criadas automaticamente pelo framework de desenvolvimento. As principais classes são user, house, plughistory, room, scheduleplug, smartplug e pendingaction, essa ultima responsável por gerar a lista de pendencias utilizada pela central doméstica.

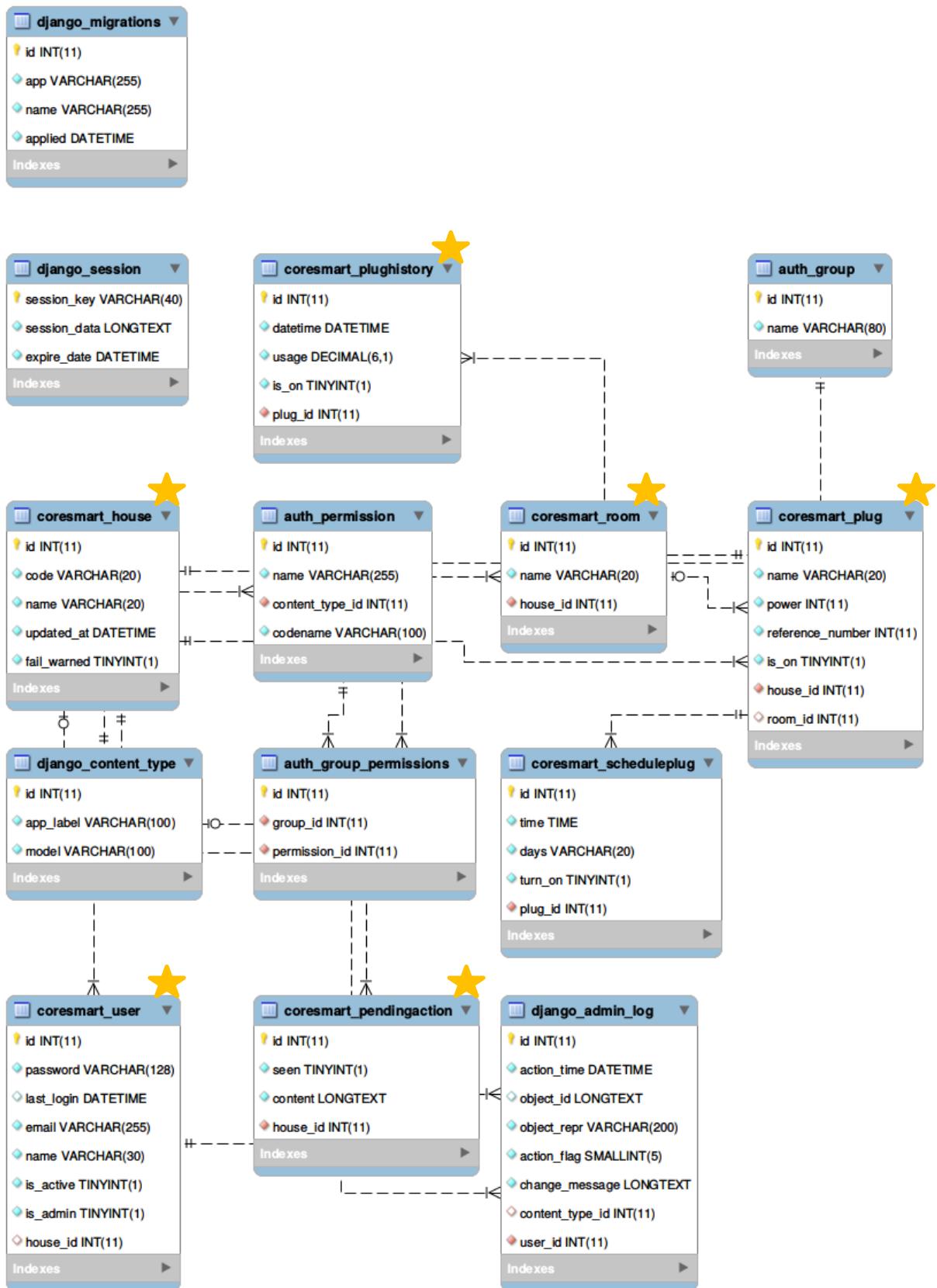


Figura 42: Diagrama de Classes do software web

7.4. Construção dos Gráficos de Consumo

Um dos requisitos do sistema deve ser a criação de gráficos de consumo de energia elétrica para visualização do usuário.

Como o hardware da Smart Plug mede a potencia aparente, devemos considerar os dados recebidos como uma aproximação da potencia ativa. Além disso, ainda temos que converter os dados de potencia em dados de energia elétrica, ou seja, transformar os valores instantâneos da potencia que estão em Watts em valores totais de energia consumida, em um dado intervalo de tempo, que devem estar em kWh.

Para fazer essa transformação devemos considerar os seguintes passos:

- i. Considerar a aproximação da potencia aparente medida como potencia ativa;
- ii. Calcular, a partir dos dados medidos (uma série de pontos, formando uma distribuição de dados discreta), a energia elétrica consumida em intervalos de tempos desejados;
- iii. Fazer todas as converções de unidades necessárias para obter os valores de energia em kWh.

Abaixo vamos explicar o método escolhido para executar a transformação dos valores de potência em instantes específicos, em valores de energia ao longo do tempo. Também citaremos mais 2 métodos considerados, mas que não forem adotados, mostrando os motivos da escolha. A ordem de apresentação adotada visa mostrar os passos pelos quais passamos para tomar a decisão final. Em todos os métodos a periodicidade de aquisição de dados pelo hardware da tomada é de uma medida a cada 3 segundos.

A base teórica para os métodos consiste no fato de que a energia é a integral temporal da potência. Assim ao plotar os dados de potência medidos em um gráfico, tentamos aproximar uma função contínua e calcular sua respectiva integral.

Os métodos alternativos não foram implementados mas são importantes para demonstrar a evolução de nosso raciocínio.

7.4.1. Método dos Retângulos

Tomando como base as distribuições de dados explicitadas nas Figuras 43 e 44, podemos calcular o consumo de energia em um certo período de tempo.

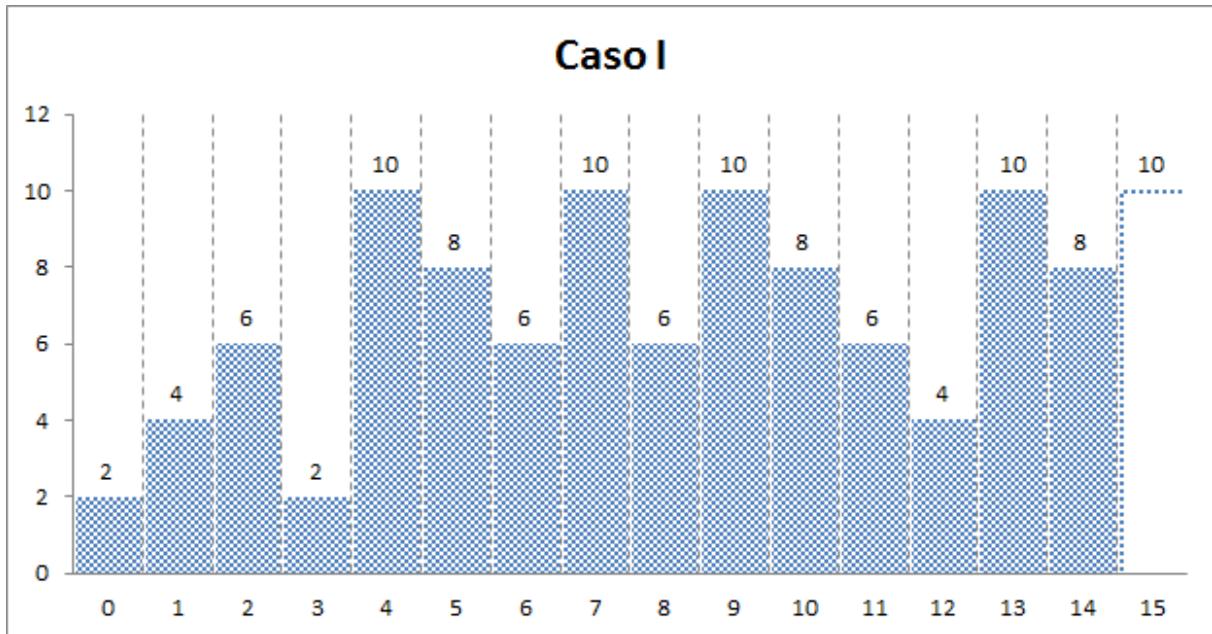


Figura 43: Gráfico da distribuição – Caso 1

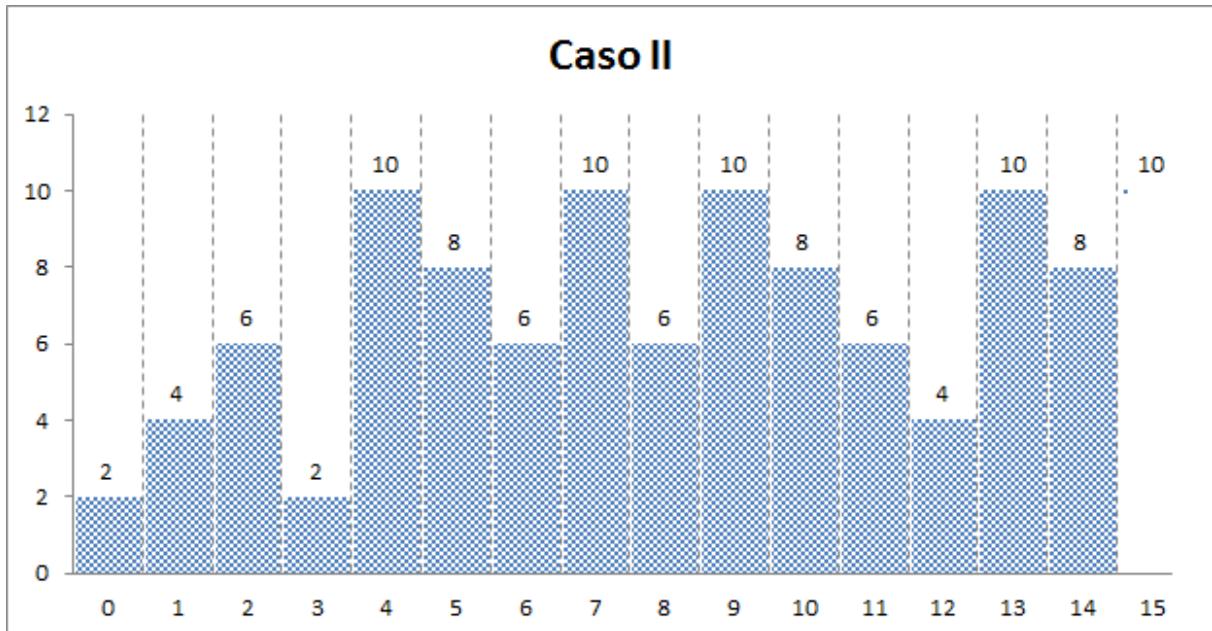


Figura 44: Gráfico da distribuição – Caso 2

Como sabemos que a Energia é a integral temporal da Potência, podemos fazer:

$$\int P(t)dt = \text{Área do Gráfico} \approx \sum_{t=1}^n A_{retângulo\ t} = \sum_{t=1}^n P(t)\Delta t$$

Como os intervalos de tempo são muito pequenos, da ordem de segundos, quando comparados aos intervalos em que desejamos calcular a integral, da ordem de horas, dias e meses, fazemos a suposição de que a função é constante nesse pequeno intervalo.

Desse modo considerando um período com n medidas, o cálculo pode ser feito das seguintes formas:

i. Considerando o último ponto medido:

$$\int P(t)dt \approx E_{Retângulo} = A_{retângulo\ 1} + A_{retângulo\ 2} + \dots + A_{retângulo\ (n-1)}$$

$$\int P(t)dt \approx E_{Retângulo} = P(t_1)\Delta t + P(t_2)\Delta t + \dots + P(t_{n-1})\Delta t$$

$$E_{Retângulo} = \Delta t[P(t_1) + P(t_2) + \dots + P(t_{n-1})] \text{ (Expressão 1)}$$

ii. Sem considerar o último ponto medido:

$$\int P(t)dt \approx E_{Retângulo} = A_{retângulo\ 1} + A_{retângulo\ 2} + \dots + A_{retângulo\ (n)}$$

$$\int P(t)dt \approx E_{Retângulo} = P(t_1)\Delta t + P(t_2)\Delta t + \dots + P(t_n)\Delta t$$

$$E_{Retângulo} = \Delta t[P(t_1) + P(t_2) + \dots + P(t_n)] \text{ (Expressão 2)}$$

Esse método tem a característica de que temos que escolher se o último ponto do período é considerado, criando uma pequena imprecisão, pois se o considerarmos, obteremos um retângulo a mais de área e se o ignorarmos, estamos negligenciando um ponto da medida. Entretanto essa imprecisão é muito pequena, ainda mais

quando consideramos períodos grandes, como 1 hora, que possui 1200 intervalos de 3 segundos.

A desvantagem principal deste método é que tornaria o software mais difícil de ser implementado em relação à metodologia adotada, além disso, seria mais custoso do ponto de vista computacional.

7.4.2. Método dos Trapézios

O Método dos Trapézios visa diminuir os erros cometidos no Método dos Retângulos, seus diferenciais são que ele leva em consideração o último ponto medido no intervalo considerado e que não considera a função constante nos intervalos, mas sim variando continuamente. Ainda que o fato de levar em consideração o último ponto propicia um ganho de precisão muito pequeno. Na figura 45, se encontra uma ilustração deste método.

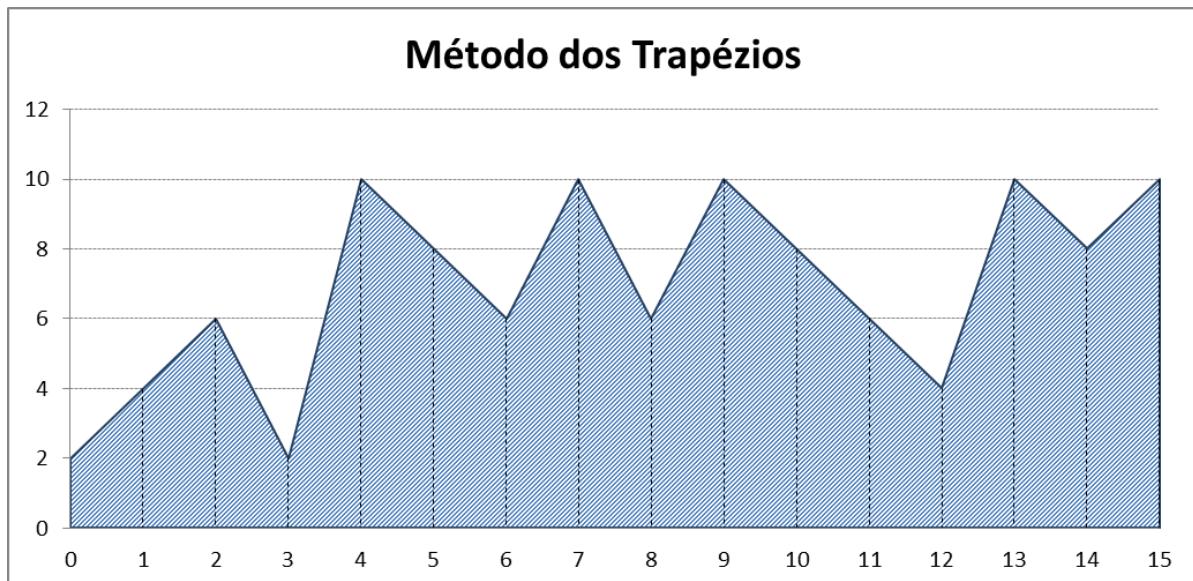


Figura 45: Método dos trapézios

Calculando a energia como a área de uma série de trapézio, temos:

$$\int P(t)dt = \text{Área do Gráfico} \approx \sum_{t=1}^n A_{trapézio\ t} = \sum_{t=1}^n \frac{(B + b)h}{2}$$

Como indica o gráfico, o cálculo da integral em um período com n medidas é feito da seguinte forma:

$$\int P(t)dt \approx E_{Trapézio} = A_{trapézio\ 1} + A_{trapézio\ 2} + \dots + A_{trapézio\ n-1}$$

$$\int P(t)dt \approx E_{Trapézio} = \frac{[P(t_1) + P(t_2)]\Delta t}{2} + \frac{[P(t_2) + P(t_3)]\Delta t}{2} \dots + \frac{[P(t_{n-1}) + P(t_n)]\Delta t}{2}$$

$$E_{Trapézio} = \Delta t \left[\frac{P(t_1)}{2} + P(t_2) \dots + P(t_{n-1}) + \frac{P(t_n)}{2} \right] \text{(expressão 3)}$$

Para comparar os métodos, será feita uma análise da expressão da energia pelos trapézios, e pelo caso II dos retângulos. É fácil perceber que a diferença consiste no fato de que os pontos iniciais e finais, isto é, P_1 e P_n , estão divididos por 2 no caso dos trapézios. Fazendo a diferença entre eles:

$$E_{Retângulo} - E_{Trapézio} = \Delta t [P(t_1) + P(t_2) + \dots + P(t_n)] - \Delta t \left[\frac{P(t_1)}{2} + P(t_2) \dots + P(t_{n-1}) + \frac{P(t_n)}{2} \right] \text{(Expressão 4)}$$

$$\text{Denominaremos } \frac{[P(t_1) + P(t_n)]}{2} \text{ de } D \text{ (diferença)}$$

Assim, para tentar achar a variação percentual entre os métodos, fazemos a razão:

$$\frac{E_{Retângulo}}{E_{Trapézio}} = \frac{(E_{Trapézio} + D)}{E_{Trapézio}} = 1 + \frac{D}{E_{Trapézio}} \text{(Expressão 5)}$$

Tentamos quantificar analiticamente a diferença entre os métodos do retângulo e do trapézio, mas é impossível, pois chegamos a uma expressão do tipo:

$$\frac{\frac{A}{2} + \frac{B}{2}}{\frac{A}{2} + \frac{B}{2} + K} \quad (\text{expressão 6})$$

Como K pode assumir infinitos valores, não é possível encontrar uma proporção exata para o caso geral.

Entretanto analisando o caso de diferença máxima, temos uma variação de 100%, esta é uma situação extremamente rara e que provavelmente não ocorreria na prática, e mesmo com a variação de 100% o valor resultante será tão baixo, que pouco importa. Ela ocorre quando temos um equipamento ligado apenas nos instantes iniciais e finais do período, fazendo com que o valor de K seja igual a 0, e consequentemente igualando a expressão 6 a 1. E finalmente, substituindo o valor de D obtido na expressão 5, temos uma diferença de 100%.

Há ainda muitos casos em que a diferença entre os dois procedimentos é mínima, por exemplo para o caso em que apenas um equipamento funcionou durante um período, e que seu funcionamento foi de acordo com o esperado, ou seja, sem variações importantes nas medidas de potência.

Assim, considera-se a hipótese de que todos os valores intermediários são iguais

$$P(t_1) = P(t_2) = \dots = P(t_{n-1}) = P(t_n) = X$$

A expressão 4, com 1200 intervalos ficaria:

$$\frac{\frac{[X+X]}{2} * \Delta t}{1200 \frac{X}{2} \Delta t} = \frac{1}{600}$$

Assim a variação porcentual, em um caso muito mais realista, seria de apenas 0,167%.

Este método teria desvantagens de a implementação ser um pouco mais difícil, e processamento ligeiramente mais custoso, do que o procedimento adotado.

7.4.3. Método da Média Total

Este último modelo foi o adotado no projeto, ele consiste em considerar o intervalo total em que se quer medir a energia, calcular a média dos pontos medidos nesse intervalo, e multiplicar pelo tempo total. A Figura 46 ilustra o método.

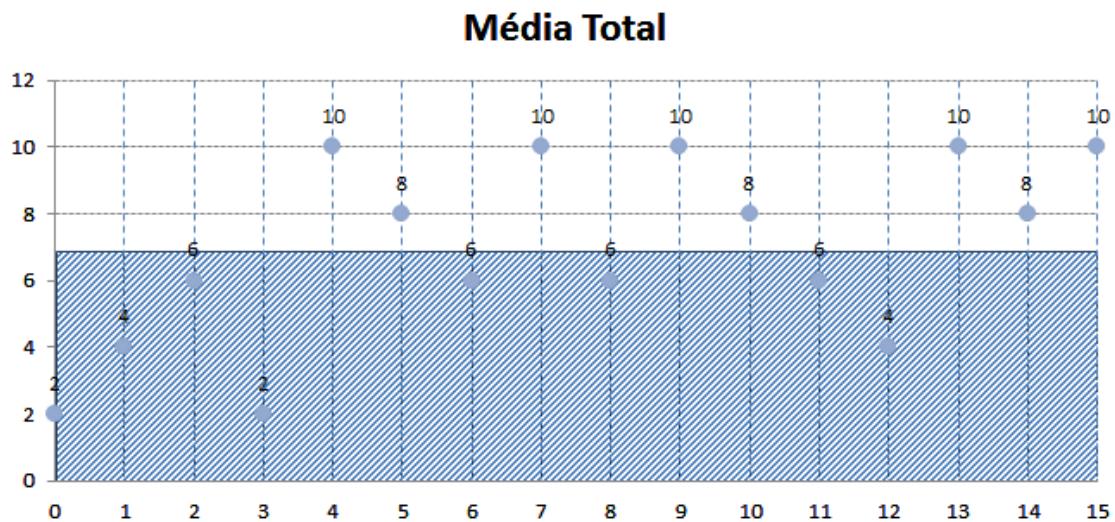


Figura 46: Média total

Mais uma vez, devemos calcular uma área para aproximar a Energia.

$$\int P(t)dt = \text{Área do Gráfico} \approx A_{\text{retângulo médio}} = \text{Base} * \text{Altura}$$

$$E_{\text{Média}} = \frac{P(t_1) + P(t_2) + \dots + P(t_n) * \text{Tempo}_{\text{total}}}{n}$$

$$E_{\text{Média}} = \frac{P(t_1) + P(t_2) + \dots + P(t_n)}{n} * (n - 1) * \Delta t \text{ (Expressão 7)}$$

Ao que chegamos que após a simplificação por meio do corte do t , a expressão 7 é igual a expressão 2, o que significa, que os métodos da Média Total e dos Retângulos são equivalentes. Com a vantagem de que o SQLite fornece diretamente

a média das medidas, e é necessário apenas um cálculo para obter a energia, tornando a implementação muito mais fácil e o processamento mais leve.

Para finalizar a discussão, queríamos apontar novamente, que estes métodos e comparações foram adicionados nesta monografia para demonstrar a nossa evolução de pensamento diante do problema da medida de energia.

7.4.4. Últimas considerações sobre os Gráficos de Consumo

Um aspecto importante da confecção dos gráficos, e a comparação entre eles, conferindo inteligência ao sistema e não se colocando apenas como um mero reproduutor de dados.

Como já mencionado a Smart Plug deve gerar relatórios com certas análises feitas em cima dos gráficos produzidos. Uma análise muito importante é a comparação de gastos mensais, já que muitas vezes as companhias de eletricidade podem cobrar tarifas diferentes em caso de aumento de consumo, e claras a maioria das pessoas faz o seu planejamento financeiro de mês a mês, e com a noção da tendência de seu gasto com energia elétrica pode planejar melhor suas finanças.

Para fazer isso, o sistema calcula mês a mês o consumo total de eletricidade, e compara a variação percentual e absoluta do gasto de energia, apresentando ao usuário as variações calculadas.

7.5. Detecção de queda de energia ou detecção de falha de comunicação via Software.

O sistema da Smart Plug é capaz de detectar quedas de energia elétrica ou falhas de comunicação entre o servidor web e a central doméstica. Essa capacidade de detecção se dá através de dois modelos, o primeiro é uma detecção através de hardware, e o segundo é uma detecção via Software.

A utilidade dessa funcionalidade é garantir dois objetivos, a proteção dos equipamentos elétricos que ficam sujeitos a picos de tensão em casos de queda e

posterior restabelecimento da energia elétrica, e também, alertar o usuário sobre esta queda ou uma possível falha de comunicação do Sistema, provocado, por exemplo, por uma falha na conexão com a Internet.

Para realizar o primeiro objetivo, ou seja, proteger os equipamentos ligados à Smart Plug, é utilizada a detecção por Hardware que está explicada em detalhes no item 6.7 em que através do circuito construído, é possível perceber a falta de energia elétrica e impedir o religamento imediato dos aparelhos no caso da volta do abastecimento, assim impedindo danos de natureza de queima equipamentos.

No que se refere ao segundo objetivo, a comunicação ao usuário da queda e/ou falha na comunicação com a tomada, é adotada uma solução por software independente daquela adotada pelo hardware. Apesar de estes dois métodos estarem desconexos, eles são capazes de atingir os objetivos estabelecidos perfeitamente. Um mesmo problema pode ter diferentes soluções, e cada uma delas pode estar focada em um objetivo específico, como é o caso presente.

Para a detecção de falha via Software é seguido o seguinte procedimento:

O servidor web recebe requisições da central doméstica a cada 3s, caso não receba nenhuma requisição por 5 minutos o servidor considera que houve uma queda de energia elétrica ou falha de comunicação com a tomada. É importante ressaltar que o sistema não consegue distinguir qual desses dois problemas ocorreu, e sua ação é emitir um “Relatório de Detecção de Queda” a ser enviado por email ao usuário.

Este método é baseado no conceito de Watchdog, uma ideia amplamente aplicada no campo da computação.

7.6. Funcionalidades do site mySmartPlug

A URL escolhida para o site é www.mysmartplug.com.br. O IP para acesso é <http://52.27.44.60/>. Na Figura 47 está a tela inicial do site. A seguir serão apresentadas as funcionalidades da Smart Plug.



Figura 47: Tela Inicial da Smart Plug

7.6.1. Cadastro de Usuário

Ao adquirir a Smart Plug, o usuário recebe fisicamente, junto ao produto, uma etiqueta contendo um código de verificação. No momento em que entra no site, ele deve clicar no botão de cadastro e inserir seus dados pessoais e o numero da etiqueta de verificação, a partir deste momento ele tem acesso total a sua tomada.

FIRST TIME?		SIGN IN	
Name	<input type="text" value="Your name"/>	E-mail	<input type="text" value="tomaz@sp.com.br"/>
E-mail	<input type="text" value="tomaz@sp.com.br"/>	Password	<input type="text" value="....."/>
Password	<input type="text" value="....."/>	SIGN IN	
Código de ativação	<input type="text" value="Código de ativação"/>		
CADASTRAR			

Figura 48: Tela de login ou Cadastro Inicial

7.6.2. Envio de Convite

Como há somente um código de verificação por produto, para permitir que terceiros utilizem a Smart Plug, o usuário deve convidá-los através de uma tela no site, esse convite é enviado através de um email, para os endereços cadastrados pelo usuário principal.

7.6.3. Cadastro de Tomada

O usuário poderá cadastrar cada tomada conectada a Smart Plug no site, este passo é importante para facilitar seu próprio monitoramento, e permitir que o sistema faça as análises comparando os dados nominais aos dedos medidos.

Ele pode fazer isto acessando a tela de cadastro de tomadas, e inserindo o nome do equipamento ligado a ela, sua potência nominal e o cômodo em que estará utilizando a tomada. Na figura 49 podemos ver a ilustração desta funcionalidade.

The screenshot shows a user interface for adding a new outlet. At the top, there is a navigation bar with icons for menu, SMARTPLUG (highlighted in red), COMO FUNCIONA, DETECÇÃO DE QUEDA, VENDA DE ENERGIA, and a search bar. Below the navigation is a section titled 'ADICIONAR TOMADA' (Add Outlet). The form contains four input fields: 'Nome' (Name) with placeholder 'Nome do aparelho ligado à essa tomada', 'Potência nominal (valor em Watts - apenas números)' (Nominal Power (value in Watts - only numbers)) with placeholder 'Potência nominal', 'Número de referência' (Reference Number) with placeholder 'Número de referência', and 'Cômodo' (Room) with placeholder 'Cômodo (opcional)'. At the bottom right of the form is a blue 'ADICIONAR' (Add) button.

Figura 49: Tela de Cadastro de Tomada

7.6.1. Lista de Cômodos

Ao cadastrar uma tomada o usuário poderá cadastrar cômodos da casa para posteriormente gerar gráficos de consumo dos mesmos. Existe uma aba com todos os cômodos cadastrados pelo usuário. Tal aba está representada na Figura 50.

Figura 50: Tela de cômodos cadastrados

7.6.2. Monitoramento das tomadas

Outra característica muito importante do sistema. Nessa funcionalidade temos uma tela de controle em que podemos acompanhar inúmeros dados sobre as tomadas ligadas a Smart Plug.

Em um formato de tabela temos listadas cada tomada, acompanhada de seu respectivo consumo nominal, consumo medido, indicador se ela está ligada ou desligada, link para os gráficos, e link para o histórico de consumo.

A Smart Plug permite que seu usuário ligue e desligue seus equipamentos elétricos remotamente. Isso é feito através do site, que tem uma tela específica para permitir essas interações. A tela de monitoramento pode ser vista na Figura 51.

Figura 51: Tela de Monitoramento de Tomadas

7.6.3. Agendamentos

Uma funcionalidade fundamental da Smart Plug é o agendamento para que os equipamentos liguem e desliguem remotamente nos horários determinado pelo usuário. Ele deve acessar a tela de Agendamento e escolher os dias e horários na semana em que deseja que a tomada ligue ou desligue. É possível fazer uma programação diária, semanal, ou avulsa (um horário isolado em que a ação de desligar ou ligar é executada apenas uma vez). As telas de agendamento estão ilustradas nas Figura 52 e 53.



The screenshot shows the 'ADICIONAR AGENDAMENTO' (Add Schedule) page. At the top, there are navigation links: 'SMARTPLUG' (highlighted in red), 'COMO FUNCIONA', 'DETECÇÃO DE QUEDA', 'VENDA DE ENERGIA', and 'Tor' (partially visible on the right). The main form has the following fields:

- Tomada:** A dropdown menu currently showing '-----'.
- Ligar:** A checkbox labeled 'Ligar'.
- Hora:** A dropdown menu labeled 'Hora' with 'Hora' selected.
- Minutos:** A dropdown menu labeled 'Minutos'.
- Repetir:** A list of days of the week with checkboxes: Domingo, Segunda-feira, Terça-feira, Quarta-feira, Quinta-feira, Sexta-feira, and Sábado. 'Domingo' is checked.
- ADICIONAR:** A blue button at the bottom right.

Figura 52: Tela para criar agendamentos

AGENDAMENTO

[+ ADICIONAR AGENDAMENTO](#)

Tomada	Dias	
Desligar Tomada do Notebook às 10:15:00	Terça-feira - Quarta-feira	Editar - Apagar
Desligar Tomada do Notebook às 19:52:00	Quarta-feira	Editar - Apagar
Desligar Televisao às 19:32:00	Quarta-feira	Editar - Apagar
Desligar Televisao às 19:49:00	Quarta-feira	Editar - Apagar
Desligar Televisao às 19:53:00	Quarta-feira	Editar - Apagar
Desligar Ventilador1 às 19:47:00	Domingo	Editar - Apagar

Figura 53: Tela com todos os agendamentos realizados

7.6.4. Gráficos

Na seção dos gráficos o usuário pode acompanhar o seu consumo diário e mensal de energia, os gráficos dos meses anteriores são armazenados permitindo uma visão completa do perfil de consumo. Os dados dos gráficos também são valiosos para o sistema fazer relatórios. Alguns exemplos de gráficos estão representados nas Figuras 54 e 55.



Figura 54: Tela com o gráfico de consumo



Figura 55: Tela com o gráfico de potência

7.6.5. Produção e Envio de Relatórios

Comparando os dados de consumo mês a mês, a Smart Plug detecta onde há aumento e de gasto de energia, também é capaz de perceber discrepâncias entre potencia nominal e potencial medida. Além de relatar as quedas de energia e falta de comunicação que ocorreram durante o funcionamento do sistema.

Todos esses relatórios produzidos ficarão disponíveis no site para a consulta do usuário, e serão arquivados caso ele deseje. Também são enviados por email para facilitar o monitoramento. O usuário pode decidir com que frequência recebe estes relatórios.

7.6.6. Detecção de Queda

Como as quedas de energia são muito comuns, elas acabam por produzir prejuízos para as pessoas devido à queima de equipamentos em suas residências. Isto ocorre porque o pico de tensão ocasionado pelo reestabelecimento da energia danifica os aparelhos.

A Smart Plug possui um dispositivo que bloqueia a tensão de chegar aos eletrodomésticos enquanto ela está muito alta. Após a tensão baixar a seus valores normais, a energia é fornecida novamente aos aparelhos.

Alem da proteção o sistema avisa seu usuário da falta de energia, por email. Um exemplo de mensgem de queda pode ser visto na figura 56.



Figura 56: Mensagem de detecção de queda

8. PLANO DE TESTES E ATINGIMENTO DE METAS.

Este capítulo apresenta o plano de testes criado pelo grupo para a validação do projeto. Cada teste proposto contém sua descrição e o resultado associado, além de observações caso necessário.

8.1. Plano de testes proposto

Todas as funcionalidades do hardware, software web e da central doméstica foram testadas individualmente durante seus respectivos processos de desenvolvimento. Para o plano de testes de validação o grupo optou por fazer os testes do sistema integrado, já que estes englobam aquelas funcionalidades e o exato funcionamento com todos os componentes implica que cada parte isolada esta executando sua tarefa corretamente.

Descrição dos testes:

1. Acesso ao site pelo IP.

Verificar se é possível acessar o site através do seu IP.

2. Acesso ao site pela URL.

Verificar se é possível acessar o site através de sua URL.

3. Cadastro de Usuário.

Verificar se o cadastro de usuário está funcionando corretamente. O usuário recebe previamente um número de série da central doméstica.

4. Login.

Verificar se o usuário é identificado corretamente.

5. E-mail de Convite.

Verificar se a funcionalidade de convidar outros usuários para utilizar a plataforma, com o envio de e-mail convite, está funcionando.

6. Acesso de Usuário Convidado.

Verificar se o usuário convidado está possibilitado de se cadastrar no site.

7. Navegação Básica no Site.

Verificar se é possível navegar no site, se todos os links estão em ordem, e não há nenhum caso de uso que leve a erro na página.

8. Ligar/Desligar equipamento.

Verificar se o sistema está de fato conseguindo Ligar/Desligar os equipamentos através dos comandos do usuário.

9. Agendar Ligar/Desligar equipamento.

Verificar se o agendamento de Ligar/Desligar realizado através do site está sendo executado corretamente.

10. Indicador de equipamento ligado.

Verificar se o site está com o desenho indicativo de equipamento ligado/desligado funcionando corretamente, ou seja, se ele apresenta um quadrado verde quando o aparelho está ligado e um quadrado vermelho quando está desligado.

11. Cadastro de Tomada.

Verificar se o usuário pode cadastrar algum equipamento, ele deve ser capaz de cadastrar um nome e uma potência nominal a uma tomada.

12. Alteração de equipamento.

Verificar se os dados de uma tomada estão realmente sendo alterados depois de uma solicitação desse tipo.

13. Apresentação da potência medida.

Verificar se o site está mostrando a potência que está sendo medida através do hardware.

14. Variação da potência medida.

Verificar se a potência medida está variando quando alteramos o equipamento que está ligado a tomada, ou quando alteramos a

intensidade de funcionamento dele. Por exemplo, aumentando nível de um ventilador, secador de cabelo ou ar condicionado.

15. Envio de e-mail no caso de potência medida discrepante.

Verificar se o sistema está calculando a diferença entre a potência nominal e a potência medida. Se a diferença calculada é maior do que 20% do valor nominal deve enviar um e-mail de aviso para o usuário.

16. Geração dos gráficos.

Verificar se os gráficos com os dados de consumo estão sendo gerados.

17. Envio de relatórios diários por e-mail.

Verificar se o sistema está enviando por e-mail para o usuário com o relatório de consumo diário.

18. Envio de relatórios mensais por e-mail.

Verificar se o sistema está enviando por e-mail para o usuário com relatório de consumo mensal.

19. Detecção de queda de energia.

Verificar se o sistema é capaz de perceber uma falha de fornecimento de energia elétrica, nesse caso a smart plug deve abrir o circuito e enviar uma alteração de estado para o site.

20. Detecção de queda de comunicação.

Verificar se o sistema é capaz de perceber a falta de comunicação entre o servidor e a central doméstica, essa falta de comunicação ocorrerá caso haja queda da conexão com a internet devido à falta de energia em toda residência. O sistema deve enviar uma mensagem de aviso de falha de comunicação.

21. Relatório de detecção de queda de energia/comunicação.

Verificar se o sistema está produzindo e enviando por e-mail um relatório em que aponta detecção na queda ou falha na comunicação com a tomada.

22. Relatórios diários.

Verificar se o relatório de consumo diário está apresentando as informações determinadas. Estas informações são a descrição do tipo de relatório (diário), o número do relatório, a data, o gráfico do consumo total de energia elétrica no dia em questão, o valor absoluto do consumo, o gráfico de consumo dos cômodos e o gráfico de consumo individual de cada tomada. Todos os valores de consumo de energia devem estar em kWh.

23. Relatórios mensais.

Verificar se o relatório de consumo mensal está apresentando as informações determinadas. Estas informações são a descrição do tipo de relatório (mensal), o número do relatório, a data, o valor absoluto do consumo de energia elétrica do mês atual e do mês anterior, o cálculo de variação do consumo entre os meses consecutivos, o gráfico do consumo total no mês, o gráfico do consumo dos cômodos, o gráfico do consumo individual de cada tomada e uma observação final mostrando a variação do consumo de energia com relação ao mês anterior. Todos os valores de energia devem estar em kWh.

8.2. Atingimento das metas estabelecidas

Nesta seção são apresentados os resultados dos testes. Para apresentar os testes é utilizado um método de tabela, em que é apontado o teste e o seu resultado.

8.2.1. Tabela de Testes do Sistema Integrado

A tabela foi construída levando em consideração todos os testes propostos e a verificação se eles funcionavam totalmente de acordo com o proposto.

Teste	Funcionamento Total
1. Acesso ao site pelo IP.	SIM
2. Acesso ao site pela URL.	SIM
3. Cadastro de Usuário.	SIM
4. Login.	SIM
5. E-mail de Convite.	SIM
6. Acesso de Usuário Convidado.	SIM
7. Navegação Básica no Site.	SIM
8. Ligar/Desligar Equipamento.	SIM
9. Agendar Ligar/Desligar equipamento.	SIM
10. Indicador de equipamento ligado.	SIM
11. Cadastro de Tomada.	SIM
12. Alteração de Equipamento.	SIM
13. Apresentação da potência medida.	SIM
14. Variação da potência medida.	SIM
15. Envio de e-mail no caso de potência medida discrepante.	SIM
16. Geração dos Gráficos.	SIM
17. Envio de relatório diários s por e-mail.	SIM
18. Envio de relatórios mensais por e-mail.	SIM
19. Detecção de queda de energia.	SIM
20. Detecção de queda de comunicação.	SIM
21. Relatório de detecção de queda de energia/comunicação.	SIM
22. Relatórios diários.	SIM
23. Relatórios mensais	SIM

8.2.2. Atingimento de metas

Como todos os testes propostos obtiveram sucesso, podemos considerar que o sistema atingiu completamente suas metas propostas. Já que ele é capaz de realizar todos os seus requisitos funcionais e permite ao usuário total controle sobre as funcionalidades propostas.

9. BUSINESS PLAN

Acreditamos que a Smart Plug não é apenas um projeto acadêmico, e que tem potencial de se tornar um produto. Neste capítulo é feito um estudo sobre a viabilidade econômica do sistema, considerando o mercado para este setor, que engloba tamanho do mercado, expectativas de crescimento, produtos semelhantes já existentes, empresas em atividade no segmento, e possibilidades de aplicação futura.

Também são propostos modelos de negócios, é estimado o preço do produto e finalmente são analisadas as possibilidades de crescimento e expectativa de faturamento para uma empresa detentora da Smart Plug em um período de 10 anos. É importante assinalar que a Smart Plug é um projeto de Engenharia Elétrica e que não faz parte do escopo do projeto produzir um Plano de Negócios extremamente rigoroso, seguindo procedimentos já estabelecidos de contabilidade e economia, mas sim uma análise intuitiva e qualitativa, para tentar estabelecer se este produto é viável, ainda assim, foi tentado adicionar o maior número de cálculos possíveis visando obter um resultado palpável.

Durante este estudo são tomadas algumas hipóteses que não estão rigorosamente justificadas e que são baseadas em um estudo qualitativo do mercado.

9.1. Ideia

A automação residencial é uma tendência evidente para o século XXI. Nesse sentido é possível imaginar diversos tipos de produtos e serviços que serão aplicados em larga escala nas moradias. Existem aplicações extremamente complexas que monitoram a saúde dos moradores, conferem inteligência ao lar e etc., até aplicações mais simples, como a da Smart Plug que permite controlar equipamentos remotamente.

Para o caso em questão, ainda há mais uma vertente de uso, que ainda não pode ser implementada, mas que muitas previsões consideram válida para as próximas décadas. É o desenvolvimento da Smart Grid, explicada no capítulo 1, quando será possível vender energia para a rede elétrica através de uma unidade geradora

residencial, a Smart Plug poderá garantir a inteligência necessária para otimizar estas transações comerciais com a rede.

9.2. Mercado

O mercado de automação residencial engloba diversos segmentos, entre eles:

- a. Entretenimento
- b. Iluminação
- c. Segurança (Sistema de Alarmes)
- d. Controle Remoto
- e. Ambiente (Ar condicionado, Calefação).

Não há um segmento específico para a Smart Plug, assim como não há para muitos outros produtos, pois eles trabalham com mais de uma aplicação, assim para fazer a análise mercadológica consideraremos a soma de todos os segmentos.

No Brasil há aproximadamente 57 milhões de residências, e segundo a Aureside (Associação Brasileira de Automação Residencial e Predial) em 2013 a expectativa de crescimento do setor era de 30% ao ano. Há ainda pesquisa realizada pela Motorola Mobility apontando que 78% dos brasileiros estariam interessados em automação residencial. Dado os grandes números da população brasileira e seu interesse em automação, e as grandes taxas de crescimento anual do setor, fica caracterizada uma boa oportunidade para um novo empreendimento.

Há diversas empresas ativas neste setor, como The Link, Belkin, Orvibo, Fifhtplay e a brasileira Contollar.

A Fifhtplay, sediada na Bélgica, é uma empresa muito interessante para este projeto, pois tem um produto muito parecido com a Smart Plug, realizando diversas das funcionalidades propostas por este projeto, como controle remoto de equipamentos, agendamentos, e venda de energia para a rede elétrica.

9.3. Modelo de Negócios

Para explorar o potencial econômico da Smart Plug, consideram-se algumas vertentes principais. A primeira é auferir renda através da venda do produto físico, o ganho que se obtém ao vender a tomada na loja. E a segunda é o aluguel do serviço, cobra-se uma assinatura mensal ou anual para que o usuário possa continuar a ter acesso ao sistema. Outras possibilidades, em que há menos possibilidade de ganho, é a cobrança pela instalação da tomada ou a cobrança de tarifa extra pelo número de usuários por tomada.

Tomamos o modelo de assinaturas como mais vantajoso, já que ele torna possível o ganho ao longo do tempo, muito depois de a tomada ser vendida o consumidor ainda continua pagando sua assinatura, e é possível mitigar os efeitos de um eventual custo de produção elevado da tomada, já que se pode cobrar um preço inferior ao custo de produção e depois recuperar o dinheiro ao longo do tempo com a cobrança das assinaturas.

O modelo ainda prevê que as transações devem ser do tipo B2C, Business-to-Consumer, e também sem considerar qualquer intermediário ou outra loja varejista.

9.4. Público Alvo

O público alvo da Smart Plug são os moradores de residência e as empresas que desejam otimizar seu consumo de energia elétrica e de aumentar o grau de automação de suas atividades.

9.5. Precificação

Temos que estabelecer dois preços, o de venda do circuito físico e o de assinatura, neste caso definindo valores para o período de tempo oferecido, no caso, mensal.

Primeiro vamos tratar da precificação do circuito físico. Para isso é necessário analisar o preço dos componentes individuais, e o custo de montagem.

Componente	Quantidade	Preço Unitário	Total
Arduino Uno REV 3	1	100	100
Xbee	2		
Shield Xbee para arduino	1	-----	380
Xbee explorer	1		
Sensor de corrente ACS712 20A	1	25	25
Placa de fenolite	1	7	7
Relé MPA-S-105-C	1	4	4
Optoacoplador 4n25	2	0,9	1,8
Resistor 1K	3	0,1	0,3
Resistor 220K	1	0,1	0,1
Resistor 470	1	0,1	0,1
Leds Vermelhos	1	0,5	0,5
Leds Verdes	1	0,4	0,4
Capacitor 100 μ F (ELETROLITICO)	1	0,2	0,2
Capacitor 0,1 μ F (poliéster)	1	0,6	0,6
Capacitor 1nF (cerâmico)	1	0,6	0,6
Transistor NPN PC 548	1	0,2	0,2
Diodo 1N4004	1	0,06	0,06
Borne de 2 pinos	1	1	1
Borne de 3 pinos	1	1	1
Total			522,86

Portanto os custos dos componentes somam R\$ 522,86. Para o custo de montagem é preciso considerar diversos aspectos, de difícil análise, como custos com mão de obra, compra de ferramentas, aluguel de instalações, logística, conta de luz, água e etc. Como estes gastos são muito complicados para estimar individualmente, e fugiriam do escopo deste Plano de Negócios, deve-se pensar em qual porcentagem do custo total da tomada serão devidos a estes encargos. Para isso vamos tomar os custos fixos como sendo 20% dos custos totais de produção. Assim assumindo o preço dos componentes como invariantes ao longo do tempo (o que o torna, na prática, um custo fixo) e somando com as despesas fixas de produção, chegamos que o custo de produção da Smart Plug é de:

$522,86 + CF = CP$, $CF = 0,2*CP$ – onde CF = Custo Fixo de Produção e CP = Custo de Produção.

CP = 653,58

Aproximamos CP para R\$ 654,00.

Agora devemos definir qual é o preço cobrado pela assinatura do sistema. Neste valor devem estar embutidos alguns valores. O preço de manutenção do sistema, funcionários e equipamentos necessários para o funcionamento dele, a recuperação de prejuízos para o caso do modelo híbrido, lucro esperado e o custo de desenvolvimento, embora este último tenha pouca importância com o passar do tempo.

Desse modo, vamos considerar o seguinte cenário:

Modelo de Assinaturas

Primeiramente consideramos que o gasto com o desenvolvimento do software deve vir com o capital inicial da empresa, e não o contemplaremos para os cálculos, então o preço tem que embutir principalmente o lucro esperado e a recuperação de prejuízos, além da mão de obra e equipamentos necessários para sua manutenção. Outro fator importante a ser levado em consideração é que o dinheiro “perdido” por uma venda abaixo do custo de produção, poderia estar rendendo em alguma aplicação financeira e também a depreciação causada pela inflação. Todos estes fatores levam a definição do que chamaremos de CO, Custo de Operação em um ano. Mais uma vez é preciso ressaltar que estes valores são extremamente complicados de se levantar, e como o plano de negócios tem uma pretensão qualitativa, iremos assumir algumas hipóteses.

Vamos estabelecer como meta, ter um lucro de 200%, sobre o custo total de cada unidade da Smart Plug, ao final de 1 ano, sendo que este custo, CT (Custo Total), é definido como $CT = CP + CO$.

Uma hipótese a ser assumida é que CO será 15% de CT, assim:

$$CT = 654,00 + 0,15*CT$$

$$CT = 769,42$$

Aproximaremos CT para R\$ 770, e CO para R\$ 115,50.

Portanto cada unidade vendida, ao final de 1 ano (não necessariamente de janeiro a janeiro, mas qualquer período de 1 ano), deve ter produzido uma receita de $3 \times CT = RE$ (Receita Esperada) = R\$ 2.310,00.

Considerando o Modelo Híbrido, e tomando um desconto no preço da venda de uma tomada como 50% do custo de produção, teremos:

I) Valor de Venda do produto físico (VPF): R\$ 327,00.

Deste modo deve ser recuperado ao longo do ano $RE - VPF = 2.310,00 - 327,00 = R\$ 1.983,00$, chegando ao valor da assinatura mensal:

II) Valor da Assinatura Mensal (VAM): = R\$ 165,25 ($1.983,00 / 12$).

A partir do segundo ano, poderíamos escolher diminuir o valor da assinatura, já que não há o prejuízo com a venda da tomada física. Mas como o cliente já pagou esse valor por 1 ano, acreditamos que podemos manter este valor de assinatura indefinidamente.

Deste modo, determinando estes dois valores, terminamos a especificação do cenário proposto.

9.6. Expectativa de Faturamento

Tendo o preço do produto e analisando o mercado, e sua taxa de crescimento, podemos estimar qual a expectativa de faturamento nos próximos 10 anos.

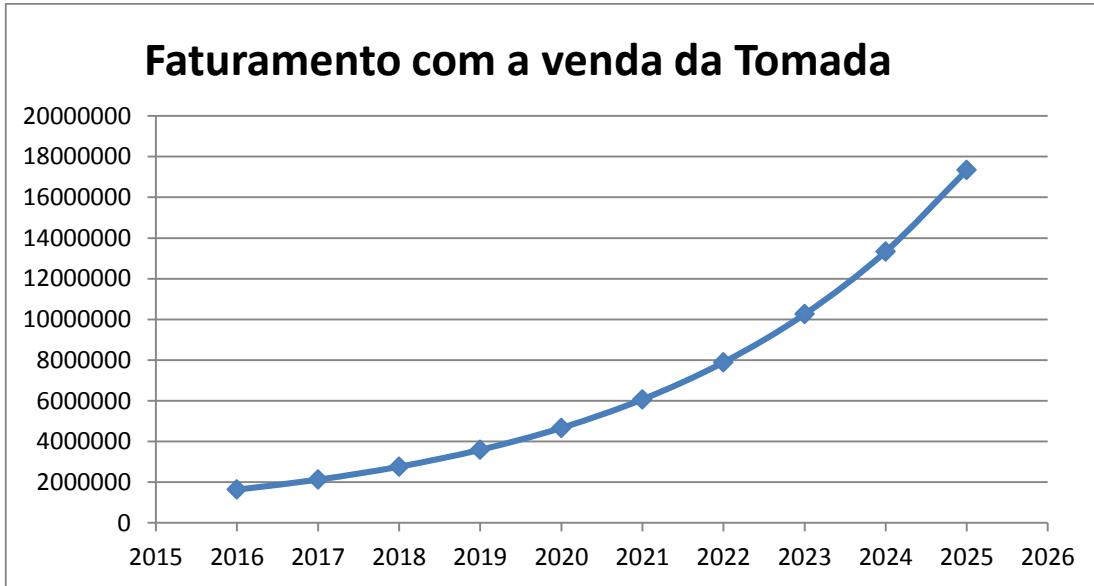
O faturamento será composto de 2 fatores, o primeiro são os produtos vendidos no mesmo ano, e o outro são as assinaturas que já não tem mais relação com a recuperação do dinheiro perdido com o desconto de venda.

Considerando o crescimento anual do mercado de 30%, como está referenciado na seção “Mercado”, os ciclos de assinatura de 1 ano começando sempre em janeiro e imaginando que no primeiro ano sejamos capazes de vender 5.000 unidades, teremos:

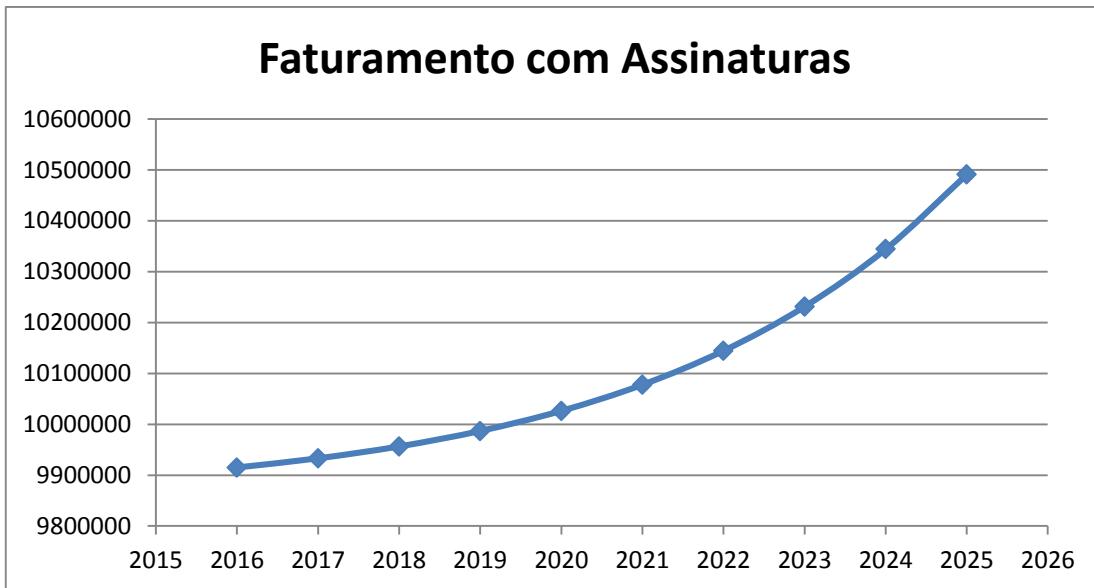
Evolução do número de unidades vendidas:



Ao final de 10 anos, teremos atingido 53022 residências um número modesto quando comparamos os 57 milhões de lares brasileiros, mas que ainda assim, podem gerar um grande faturamento.

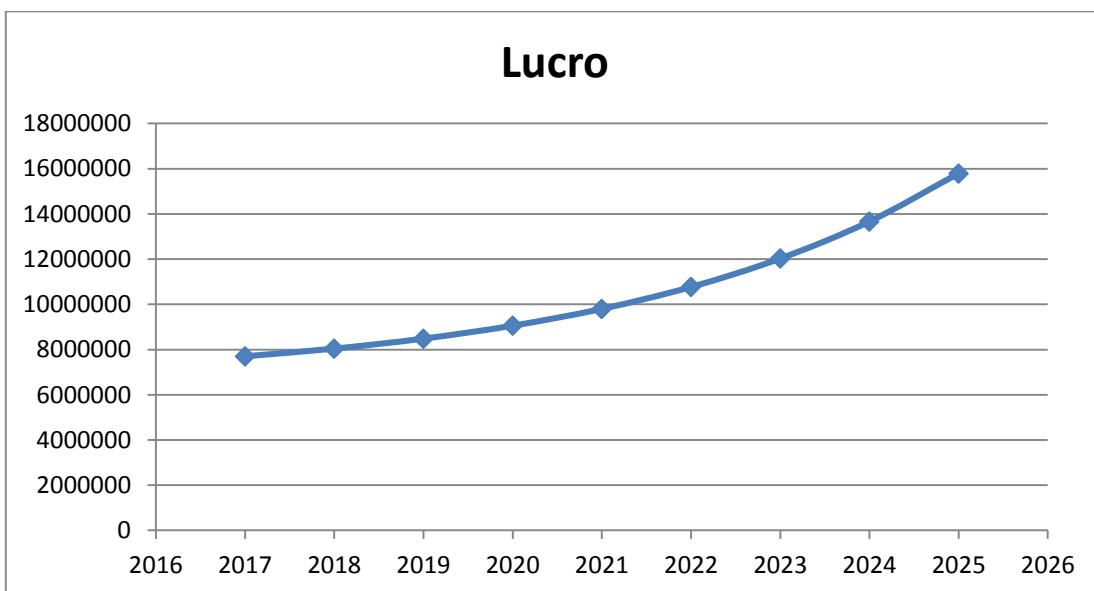


Apenas o faturamento com a venda da tomada, passará de R\$ 17 milhões em 2025



O faturamento com assinaturas chegará a quase 11 milhões em 10 anos.

Finalmente vamos analisar qual terá sido o lucro da empresa no período, ele será a diferenças do faturamento total, dado pela soma do faturamento com a venda da tomada com o faturamento com assinaturas, subtraindo os custos totais de produção e operação. Não foram considerados descontos com pagamentos de impostos.



O lucro obtido em 10 anos será de R\$ 18.553.084,00.

Portanto o produto tem uma capacidade aceitável de gerar receita, ele ainda tem a capacidade de crescer a uma tava maior, e representar uma fatia maior do mercado, possibilitando um grande aumento nos resultados.

9.7. Análise de SWOT

Existe um modo muito interessante de expressar qualitativamente as possibilidades de um negocio. É o SWOT, na Figura 57 apresentamos este diagrama:

SWOT – Neste diagrama representamos as Forças e Fraquezas de origem interna do plano, (SW – Strength and Weakness), e as Oportunidade e Ameaças de origem externa (OT – Opportunities and Threats).

Diagrama SWOT	
Forças (Fator Interno) 1- Produto bem organizado e funcional. 2- Expertise dos sócios sobre a tecnologia utilizada.	Fraquezas (Fator Interno) 1- Pouca experiência em empreendedorismo . 2- Baixo capital inicial dos sócios
Oportunidades (Fator Externo) 1- Mercado em rápido crescimento (30% ao ano). 2- Mercado carece de muitas empresas no setor.	Ameaças (Fator Externo) 1- Pouca quantidade de mão de obra qualificada disponível. 2- Preços elevados dos componentes do sistema.

Figura 57: Análise de SWOT da Smart Plug

9.8. Possibilidades Futuras

Com o surgimento da possibilidade de venda de energia produzida por unidades residenciais para a Rede Elétrica, surge uma nova aplicação para a Smart Plug e também um novo modelo de negócio.

A demanda por um sistema que fosse capaz de acompanhar a flutuação na cotação dos preços da eletricidade, e que pudesse gerenciar os melhores momentos para a venda dela, certamente aumentaria. Além disso, seria possível pensar em um modelo que cobrasse uma porcentagem sobre todo o lucro auferido com a venda de energia, já que a gestão dessas transações ficaria sob a responsabilidade do sistema.

10. CONSIDERAÇÕES FINAIS

Neste Capítulo serão encontrados alguns pensamentos finais referentes ao projeto. Discutiremos as contribuições deste trabalho, as dificuldades que foram encontradas durante seu desenvolvimento, a abrangência de temas da Engenharia que ele comprehende e os planos para serem implementados no futuro. Finalmente há a conclusão do trabalho.

10.1. Contribuições

O conceito por trás das Smart Plugs é recente sendo que os primeiros modelos chegaram ao mercado em 2014. Geralmente elas possibilitam apenas o acionamento remoto, algumas ainda medem a potência consumida. Mas todos os modelos vistos usam o Wifi como tecnologia de comunicação sem fio, essa tecnologia não é escalável e possui um preço relativamente alto. Um dos diferenciais da tomada proposta pelo grupo é o uso da tecnologia Zigbee. Essa tecnologia se adequa muito melhor ao contexto de internet das coisas que vivenciamos hoje. Além de permitir um grande número de dispositivos na mesma rede, o Zigbee também faz um uso mais eficiente da energia e permite que aparelhos se comuniquem entre si, e não apenas com um nó central.

Outro grande diferencial que a tomada proposta fornece é a possibilidade de aumentar a vida útil dos aparelhos. Ao evitar que o aparelho esteja sujeito a uma sobretensão pós-queda de energia, a tomada impede que o mesmo se danifique ou queime. Essa funcionalidade está bastante adequada com a realidade do Brasil, onde apagões e quedas de energia são frequentes em todo território nacional.

Mesmo funcionalidades presentes em alguns modelos comerciais, como a medição de consumo, são potencializadas na tomada proposta pelo grupo. O sistema gera relatórios bastante detalhados sobre o consumo da residência possibilitando ao usuário consultar as informações de tomadas agrupadas em cômodos. O sistema ainda possui um aviso de consumo excessivo, informando o usuário quando a tomada está consumindo uma potência maior que a potência nominal do aparelho ligado à tomada.

10.2. Dificuldades

Durante a realização do projeto foram encontradas algumas dificuldades. A realização da divisão do sistema em camadas e sua posterior integração provocou erros e problemas inesperados que atrasaram a finalização do projeto. Nesse sentido conseguir criar uma Central Doméstica que fosse capaz de atender seus requisitos se mostrou desafiador. A detecção de queda de energia e seu posterior reconhecimento pelo software também foi complicado e para contornar isso foi criada a detecção por software avisando o usuário de qualquer tipo de falha de comunicação prolongada entre a central doméstica e o servidor.

O cálculo da potência se mostrou muitas vezes confuso e difícil de realizar, tanto em sua medição no Hardware como no tratamento dos dados pelo Software. Precisamos considerar hipóteses bem justificadas como a aproximação da potência aparente com a potência ativa, e outras aproximações para facilitar cálculos de conversão de potência para energia.

O Business Plan trouxe dificuldades pois não é fácil encontrar informações coerentes sobre o tamanho de um mercado, presença de empresas em certos setores, e inclusive há certas informações que são quase impossíveis de supor de antemão como a parcela de alguns tipos de custo no preço final e etc., e muitas hipóteses devem ser assumidas prejudicando a exatidão do modelo.

10.3. Abrangencia do projeto

Neste projeto foram necessárias inúmeras competências adquiridas durante o curso de Engenharia Elétrica com Ênfase em Computação, o objetivo desta seção é ressaltar todas as disciplinas que serviram como base para o desenvolvimento do sistema.

A montagem do Hardware implica conhecimentos de Circuitos Elétricos I e II, e de Práticas de Eletricidade e Eletrônica, em que foi aprendido desde como alimentar a tomada, até fazer um circuito impresso, documentar os diagramas do circuito, trabalhar com soldas e embasar o método utilizado para captura da potência. Para transferir as informações medidas, e executar as ordens de ligar/desligar, utilizamos

um Arduino, que aprendemos a manipular nas aulas de Laboratório de Processadores sobre microcontroladores.

A confecção do Software, tanto da Central Doméstica como do Servidor, necessita de conhecimentos de programação, onde obtivemos boa base com o Laboratório de Programação, noções de projeto e gerenciamento de software, obtidos em Engenharia de Software I e II, em que também aprendemos um pouco sobre bancos de dados e a aplicação destes conceitos em projetos nos Laboratórios de Software I e II.

O conhecimento obtido nas disciplinas de Redes de Computadores também se fez necessário, principalmente na comunicação entre a tomada, a central doméstica e o servidor, e na escolha da tecnologia Zig Bee.

Ainda pode ser citada a matéria de Criação e Administração de Empresas de Computação, onde tivemos a ideia de tornar o sistema da Smart Plug em um produto, e tivemos noções de como analisar sua viabilidade econômica e produzir um Plano de Negócios.

Tendo isto em vista, o projeto certamente teve grande abrangência dentro da Engenharia de Computação, e proporcionou uma série de desafios em que foi necessário combinar diferentes conhecimentos.

10.4. Trabalhos Futuros

Existem diversos aspectos que podem ser incorporados futuramente em nosso projeto. Nesta seção vamos mostrar alguns deles. Eles não foram realizados pelo grupo por diversas razões, como tempo necessário para a implementação, conhecimento requerido e tecnologia atual.

10.4.1. Algoritmo estocástico para aquisição de dados

A tomada projetada captura dados a cada 3 segundos, em um período de 1 hora são 1.200 valores captados, em um dia este número chega a 28.800, todas essas medidas também geram custo de energia e de processamento. Nesse sentido seria de grande importância criar uma maneira de captar os dados de forma mais eficiente,

diminuindo o número de medições, mas mantendo a acuidade dos valores de consumo calculados. Para isso poderá ser criado futuramente, um algoritmo estocástico que contemple esses dois requisitos.

10.4.2. Detecção de Perfil

Outra característica a ser adotada, que também está relacionada a conferir inteligência ao sistema, é um algoritmo que perceba o perfil de consumo do usuário, assim quando houver uma situação de uso dos equipamentos incomum, o sistema emite um alerta ao usuário. Isso pode ser útil para ele desligar remotamente aparelhos que esqueceu ligado, ou mesmo evitar um acidente, como incêndio, quando um ferro de passar roupa, ou uma torradeira são deixados ligados por muito tempo.

10.4.3. Controle do Stand-by

O Stand-by tem a função de permitir uma resposta rápida dos aparelhos a comandos do controle remoto, ou algum tipo de intervenção do usuário, e de evitar picos de tensão no acionamento dos eletrodomésticos. Entretanto é notório que o Stand-by é uma grande fonte de gasto de energia. Para evitar isso, a Smart Plug poderia ser dotada de um algoritmo que percebesse quando os equipamentos entram em stand-by, e não têm perspectiva de uso, como é o caso de um rádio ou uma televisão de madrugada, e desligar o stand-by até o início da manhã, ou seja, ser dotado de um mecanismo que permitia ao sistema interferir no stand-by e saber quando ele deverá desativar o stand-by.

10.4.4. Criar um aplicativo para Android e para iOS

A Smart Plug não possui interface para aplicativos, como estes são umas das principais funcionalidades utilizadas pelos usuários de celular, que compreendem grande parte da população, é muito importante para o sucesso da Smart Plug como produto, que possua uma versão de aplicativo para os sistemas operacionais mais difundidos, como o Android e o iOS.

10.4.5. Adicionar uma porta USB

Uma tendência vista em muitos países desenvolvidos é adicionar às tomadas uma porta USB, a Smart Plug poderia aderir a esta tendência e permitir que seus usuários também possam conectar aparelhos com USB, como é o caso de MacBooks. Todas as funcionalidades do sistema como acionamento remoto, agendamento, medida de energia, e etc., também deveriam ser expandidos para os equipamentos ligados na porta USB.

10.4.6. Customização de Relatório

Pode-se implementar uma funcionalidade que permita ao usuário customizar seus relatórios de consumo de energia, fazendo com que ele possa escolher quais informações estarão presentes no relatório, se deseja que eles sejam enviados por e-mail e com qual periodicidade ele os deseja receber.

10.4.7. Interação com a Smart Grid

O ultimo ponto a ser considerado é a venda de energia para a rede elétrica, neste caso ainda há uma defasagem na tecnologia da própria rede, e não só na da Smart Plug.

A ideia é que as unidades que são abastecidas pela rede, não serão apenas consumidoras de eletricidade, mas se tornarão também unidades geradoras que venderão seu excedente de produção para a rede.

Nesse contexto haverá demanda para um sistema inteligente que seja capaz de analisar os melhores instantes para vender a eletricidade, e que também seja capaz de armazená-la para realizar as vendas nos momentos de interesse. Os instantes certos para vender a energia mudam porque poderá haver além da mudança usual na demanda, como ocorre em momentos de pico, haverá também mudança na oferta, já que em dias ensolarados a produção de eletricidade pelas unidades residenciais que serão basicamente solares, ou em menor grau eólica e fóssil (geradores de óleo e carvão), será maior, portanto aumentando a oferta. E de acordo com a lei de oferta e procura, diminuindo o preço. Não é só a variação na

ensolação que produzirão esse efeito, variações na velocidade dos ventos e nos preços do petróleo e carvão, também.

Assim um sistema que fosse dotado de uma inteligência capaz de acompanhar essas flutuações no preço da energia, e que pudesse armazená-la para vender em momentos adequados, seria de grande utilidade para uma empresa, ou mesmo uma residência.

10.5. Conclusão

Este projeto demonstrou a concepção e a implementação de um sistema de automação residencial, tendo como objetivo reduzir o consumo de eletricidade e aumentar a comodidade de seus usuários. Sua ideia principal foi adicionar funcionalidades a uma tomada comum, e integrá-la com sistemas computacionais de modo a conferir às pessoas mais informações e maior controle sobre suas casas. Foi necessário integrar uma série de conhecimentos sobre circuitos elétricos, micro-controladores e programação, para que o sistema pudesse funcionar como um todo. Já que ele envolvia tanto a construção de um Hardware como a de Softwares.

Tivemos que desenvolver um plano para construir essas partes separadamente, e depois criar maneiras de integrá-lo. Um plano de testes subsequente foi muito importante para garantir que os requisitos funcionais do sistema estavam, de fato, operando satisfatoriamente.

Há ainda a certeza de que projetos deste tipo serão muito importantes no século XXI, e de que eles irão se difundir em ritmo acelerado nos próximos anos. A presença de empresas que já comercializam produtos similares com o sistema desenvolvido neste trabalho é uma prova irrefutável de sua relevância. O fato de que ele pode ser usado para racionalizar o consumo de energia, também é de grande valor já que a sustentabilidade é uma característica fundamental da economia moderna.

Os avanços tecnológicos esperados na Rede Elétrica, discutidos ao longo desta monografia, ainda trazem mais possibilidades de aplicações e expansões para a Smart Plug.

Desse modo, por envolver um grande número de competências ligadas à Engenharia Elétrica e de Computação, por se tratar de um tema atual e de possibilidade de aplicação prática, por estar inserido no contexto da sustentabilidade e do uso racional dos recursos naturais, por ter espaço para melhorias e para incorporação de novas características conforme o avanço das tecnologias existentes e finalmente por já haver sistemas de natureza similar implementados por empresas, podemos dizer que o projeto apresenta a complexidade e abrangência desejadas em um projeto de formatura e se mostra bem alinhado com as tendências tecnológicas futuras.

11. REFEÊNCIAS

- [1] SCHONARTH, J. P. Aparelho em stand by eleva conta em até 20%. *Gazeta do Povo*. 2013. Disponível em: <<http://www.gazetadopovo.com.br/economia/aparelho-em-stand-by-eleva-conta-em-ate-20-0rneb09bnlyfyop0gc3sq1k78>>. Acesso em: 30 de Outubro de 2015.
- [2] RUSSO, M. Modo stand by: o pequeno vilão doméstico do consumo de energia. *O Globo*. 2014. Disponível em: <<http://oglobo.globo.com/economia/defesa-do-consumidor/modo-stand-by-pequeno-vilao-domestico-do-consumo-de-energia-14542263>>. Acesso em: 30 de Outubro de 2015.
- [3] FILHO, A. M. O Consumo de Energia no Modo Standby. *Espaço Acadêmico*. 2001. Disponível em: <<http://www.espacoacademico.com.br/005/05mendes.htm>>. Acesso em: 30 de Outubro de 2015.
- [4] FILHO, A. M. O Consumo de Energia no Modo Standby (2). *Espaço Acadêmico*. 2008. Disponível em: <<http://www.espacoacademico.com.br/089/89amsf.htm>>. Acesso em: 30 de Outubro de 2015.
- [5] IND., Smart Grid, O que é. *CPFL*. 2015. Disponível em: <<http://www.cpfl.com.br/energias-sustentaveis/sites-tematicos/smart-grid.aspx>>. Acesso em: 20 de Novembro de 2015.
- [6] TORRES, G. Tudo o que você precisa saber sobre fontes de alimentação. *Clube do Hardware*. 2008. Disponível em: <<http://www.clubedohardware.com.br/artigos/tudo-o-que-voce-precisa-saber-sobre-fontes-de-alimentacao/976/10>>. Acesso em: 8 de Maio de 2015.
- [7] IND., Grandezas Elétricas. *Eficien*. 2010. Disponível em: <<http://www.eficien.com.br/fontes-energia-impactos/>>. Acesso em: 8 de Maio de 2015.

- [8] IND., RESOLUÇÃO ANEEL Nº 456. *ANEEL*. 2000. Disponível em: <<http://www.aneel.gov.br/cedoc/bres2000456.pdf>>. Acesso em: 8 de Maio de 2015.
- [9] IND., Estudos de Energia. *Q Energia*. 2012. Disponível em: <<http://www.qenergia.pt/content/index.php?action=detailfo&rec=337>>. Acesso em: 9 de Maio de 2015.
- [10] IND., Padrão 802.15.4. *UFRJ*. 2015. Disponível em: <http://www.gta.ufrj.br/grad/10_1/zigbee/padrao.html>. Acesso em: 15 de Maio de 2015.
- [11] IND., Python Advanced Library. *Python Software Foundation*. 2015. Disponível em: <<https://www.python.org/dev/peps/pep-0206/>>. Acesso em: 05 de Novembro de 2015.
- [12] IND., Python. *Wikipedia*. 2015. Disponível em: <<https://pt.wikipedia.org/wiki/Python>>. Acesso em: 05 de Novembro de 2015..
- [13] IND., Conheça o Django. *Django Brasil*. 2015. Disponível em: <<http://www.djangobrasil.org/>>. Acesso em: 17 de Maio de 2015.
- [14] IND., REST. *Wikipedia*. 2015. Disponível em: <<https://pt.wikipedia.org/wiki/REST>>. Acesso em: 05 de Novembro de 2015.
- [15] LABBATE, A. Como será a prestação de serviços ao cliente em 20 anos? *CIO*. 2015. Disponível em: <<http://cio.com.br/opiniao/2015/10/05/como-sera-a-prestacao-de-servicos-ao-cliente-em-20-anos/>> Acesso em: 05 de Outubro de 2015.
- [16] IND., WHAT IS A RASPBERRY PI? *Raspberry Pi Foundation*. 2015. Disponível em: <<https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>> Acesso em: 10 de Novembro de 2015.

[17] IND., Introdução ao JSON. JSON ORG. 2015. Disponível em: <<http://www.json.org/json-pt.html>> Acesso em: 11 de Novembro de 2015.

[18] ALECRIM, E. Usando cron e crontab para agendar tarefas. *Viva o Linux*. 2005. Disponível em: <<http://www.vivaolinux.com.br/artigo/Usando-cron-e-crontab-para-agendar-tarefas>> Acesso em: 11 de Novembro de 2015.

12. REFERÊNCIAS DE FIGURAS

- [F1] IND., Connected Home. *D-Link*. 2015. Disponível em: <<http://us.dlink.com/products/connected-home/wi-fi-smart-plug-3/>>. Acesso em: 20 de Maio de 2015.
- [F2] IND., Connected Home. *D-Link*. 2015. Disponível em: <<http://us.dlink.com/products/connected-home/wi-fi-smart-plug/>>. Acesso em: 20 de Maio de 2015.
- [F3] IND., Resource Center. *Belkin*. 2015. Disponível em: <<http://www.belkin.com/us/p/P-F7C027/>>. Acesso em: 20 de Maio de 2015.
- [F4] IND., Power Factor Correction. *ON Semiconductor*. 2015. Disponível em: <http://www.onsemi.com/pub_link/Collateral/HBD853-D.PDF> Acesso em: 09 de Maio de 2015.
- [F5] Frias, R. N. ZigBee: O que é. *Teleco*. 2015. Disponível em: <http://www.teleco.com.br/tutoriais/tutorialzigbee/pagina_1.asp>. Acesso em: 15 de Maio de 2015.
- [F6] Allegro MicroSystems. Datasheet: Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor. 2006..
- [F7] IND., Arduino uno. *Arduino*. 2015. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 24 de Outubro de 2015.
- [F8] IND., SparkFun XBee Explorer USB. *SparkFun*. 2015. Disponível em: <<https://www.sparkfun.com/products/11812>>. Acesso em: 24 de Outubro de 2015.

[F9] IND., XBee 802.15.4 (Series 1). *Gravitech*. 2015. Disponível em: <<http://www.gravitech.us/xb80s163porf.html>>. Acesso em: 24 de Outubro de 2015.

13. APÊNDICE A – DEFINIÇÃO DE REQUISITOS

Abaixo a metodologia adotada para a definição de requisitos do software web:

No: 1	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não funcional
Requisito: Permitir o login de vários usuários.		
Descrição: Permite que o usuário tenha acesso ao sistema através de uma identificação.		
Prioridade: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa		
Status: <input type="checkbox"/> Proposto <input type="checkbox"/> Aprovado <input type="checkbox"/> Incorporado <input checked="" type="checkbox"/> Validado		
Estabilidade: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa		
Origem: <input checked="" type="checkbox"/> Usuário <input type="checkbox"/> Cliente <input type="checkbox"/> Competidor <input type="checkbox"/> Interna		
Requisitos associados: 2		

No: 2	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não funcional
Requisito: Cadastro de usuários (ligados a uma central doméstica).		
Descrição: Permite que o usuário cadastre um login e senha no sistema, a partir de um código de verificação obtido com a compra da Smart Plug.		
Prioridade: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa		
Status: <input type="checkbox"/> Proposto <input type="checkbox"/> Aprovado <input type="checkbox"/> Incorporado <input checked="" type="checkbox"/> Validado		
Estabilidade: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa		
Origem: <input checked="" type="checkbox"/> Usuário <input type="checkbox"/> Cliente <input type="checkbox"/> Competidor <input type="checkbox"/> Interna		
Requisitos associados: 1, 3		

No: 3	<input type="checkbox"/> Funcional	<input type="checkbox"/> Não funcional
Requisito: Permitir o cadastro de várias centrais domésticas (casas).		
Descrição: Permite que o várias casas sejam cadastradas no sistema.		
Prioridade: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa		
Status: <input type="checkbox"/> Proposto <input type="checkbox"/> Aprovado <input type="checkbox"/> Incorporado <input checked="" type="checkbox"/> Validado		
Estabilidade: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa		
Origem: <input checked="" type="checkbox"/> Usuário <input type="checkbox"/> Cliente <input type="checkbox"/> Competidor <input type="checkbox"/> Interna		
Requisitos associados: 1, 2, 3		

No: 4	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não funcional
Requisito: Visualização do Estado das Tomadas		
Descrição: Permite que o usuário visualize o estado das tomadas ligadas ao sistemas, isto é, saber se o circuito interno da smart plug está aberto ou fechado, o que implica em funcionamento ou não de algum eletrodoméstico ligado à ela.		
Prioridade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média
Status:	<input type="checkbox"/> Proposto	<input type="checkbox"/> Aprovado
Estabilidade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média
Origem:	<input checked="" type="checkbox"/> Usuário	<input type="checkbox"/> Cliente
Requisitos associados: 1, 3 e 5		

No: 5	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não funcional
Requisito: Ligar e Desligar Tomadas		
Descrição: Permite que o usuário ligue e desligue as tomadas ligadas ao sistema remotamente. A partir do site da SmartPlug, o qual ele deve estar logado.		
Prioridade:	<input type="checkbox"/> Alta	<input type="checkbox"/> Média
Status:	<input type="checkbox"/> Proposto	<input type="checkbox"/> Aprovado
Estabilidade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média
Origem:	<input checked="" type="checkbox"/> Usuário	<input type="checkbox"/> Cliente
Requisitos associados: 1,3,4, e 6		

No: 6	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não funcional
Requisito: Agendamento de Comando de Ligar e Desligar		
Descrição: Permite ao o usuário que agende os instantes em que ele deseja que seus equipamentos sejam ligados ou desligados, automaticamente.		
Prioridade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média
Status:	<input type="checkbox"/> Proposto	<input type="checkbox"/> Aprovado
Estabilidade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média
Origem:	<input checked="" type="checkbox"/> Usuário	<input type="checkbox"/> Cliente
Requisitos associados: 1,3,5		

No: 7	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não funcional
Requisito: Armazenar dados de consumo		
Descrição: O sistema deve ser capaz de armazenar os dados de consumo recebidos do Hardware da SmartPlug.		
Prioridade:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média
Status:	<input type="checkbox"/> Proposto	<input type="checkbox"/> Aprovado
Estabilidade:	<input type="checkbox"/> Alta	<input type="checkbox"/> Média
Origem:	<input type="checkbox"/> Usuário	<input type="checkbox"/> Cliente
Requisitos associados: 1, 3 8, 9, 10		

No: 8	(<input checked="" type="checkbox"/>) Funcional	(<input type="checkbox"/>) Não funcional
Requisito: Vizualização de Dados de Consumo		
Descrição: Permite que o usuário vizualize os dados de consumo de energia elétrica de seus equipamentos.		
Prioridade: (<input checked="" type="checkbox"/>) Alta (<input type="checkbox"/>) Média (<input type="checkbox"/>) Baixa		
Status: (<input type="checkbox"/>) Proposto (<input type="checkbox"/>) Aprovado (<input type="checkbox"/>) Incorporado (<input checked="" type="checkbox"/>) Validado		
Estabilidade: (<input checked="" type="checkbox"/>) Alta (<input type="checkbox"/>) Média (<input type="checkbox"/>) Baixa		
Origem: (<input checked="" type="checkbox"/>) Usuário (<input type="checkbox"/>) Cliente (<input type="checkbox"/>) Competidor (<input type="checkbox"/>) Interna		
Requisitos associados: 1,3, 7,9 e 10		

No: 9	(<input checked="" type="checkbox"/>) Funcional	(<input type="checkbox"/>) Não funcional
Requisito: Geração de Gráficos kWh x Tempo		
Descrição: O sistema deve ser capaz de gerar gráficos a partir dos dados recebidos do hardware da tomada. Os gráficos devem mostrar o consumo de energia elétrica em um período de 1 dia, e no período de 1 mês.		
Prioridade: (<input checked="" type="checkbox"/>) Alta (<input type="checkbox"/>) Média (<input type="checkbox"/>) Baixa		
Status: (<input type="checkbox"/>) Proposto (<input type="checkbox"/>) Aprovado (<input type="checkbox"/>) Incorporado (<input checked="" type="checkbox"/>) Validado		
Estabilidade: (<input checked="" type="checkbox"/>) Alta (<input type="checkbox"/>) Média (<input type="checkbox"/>) Baixa		
Origem: (<input checked="" type="checkbox"/>) Usuário (<input type="checkbox"/>) Cliente (<input type="checkbox"/>) Competidor (<input type="checkbox"/>) Interna		
Requisitos associados: 7,8, 11		

No: 10	(<input checked="" type="checkbox"/>) Funcional	(<input type="checkbox"/>) Não funcional
Requisito: Cadastro/Alteração de Tomadas		
Descrição: Permite que o usuário cadastre o nome e os valores nominais dos equipamentos ligados à uma certa tomada. Também permite ao usuário fazer alterações neste cadastro.		
Prioridade: (<input checked="" type="checkbox"/>) Alta (<input type="checkbox"/>) Média (<input type="checkbox"/>) Baixa		
Status: (<input type="checkbox"/>) Proposto (<input type="checkbox"/>) Aprovado (<input type="checkbox"/>) Incorporado (<input checked="" type="checkbox"/>) Validado		
Estabilidade: (<input checked="" type="checkbox"/>) Alta (<input type="checkbox"/>) Média (<input type="checkbox"/>) Baixa		
Origem: (<input checked="" type="checkbox"/>) Usuário (<input type="checkbox"/>) Cliente (<input type="checkbox"/>) Competidor (<input type="checkbox"/>) Interna		
Requisitos associados: 4, 5, 6, 7, 8		

No: 11	(<input checked="" type="checkbox"/>) Funcional	(<input type="checkbox"/>) Não funcional
Requisito: Comparação de Gasto de Energia		
Descrição: Permite que o usuário compare os valores nominais de um certo equipamento com os dados reais medidos pela tomada.		
Prioridade: (<input checked="" type="checkbox"/>) Alta (<input type="checkbox"/>) Média (<input type="checkbox"/>) Baixa		
Status: (<input type="checkbox"/>) Proposto (<input type="checkbox"/>) Aprovado (<input type="checkbox"/>) Incorporado (<input checked="" type="checkbox"/>) Validado		
Estabilidade: (<input checked="" type="checkbox"/>) Alta (<input type="checkbox"/>) Média (<input type="checkbox"/>) Baixa		
Origem: (<input checked="" type="checkbox"/>) Usuário (<input type="checkbox"/>) Cliente (<input type="checkbox"/>) Competidor (<input type="checkbox"/>) Interna		
Requisitos associados: 7, 8		

No: 12	(<input checked="" type="checkbox"/>) Funcional	(<input type="checkbox"/>) Não funcional
Requisito: Funcionar como um servidor WEB		
Descrição: O sistema deve poder ser acessado através da WEB.		
Prioridade: (<input checked="" type="checkbox"/>) Alta (<input type="checkbox"/>) Média (<input type="checkbox"/>) Baixa		
Status: (<input type="checkbox"/>) Proposto (<input type="checkbox"/>) Aprovado (<input type="checkbox"/>) Incorporado (<input checked="" type="checkbox"/>) Validado		
Estabilidade: (<input checked="" type="checkbox"/>) Alta (<input type="checkbox"/>) Média (<input type="checkbox"/>) Baixa		
Origem: (<input type="checkbox"/>) Usuário (<input type="checkbox"/>) Cliente (<input type="checkbox"/>) Competidor (<input checked="" type="checkbox"/>) Interna		
Requisitos associados: 1		

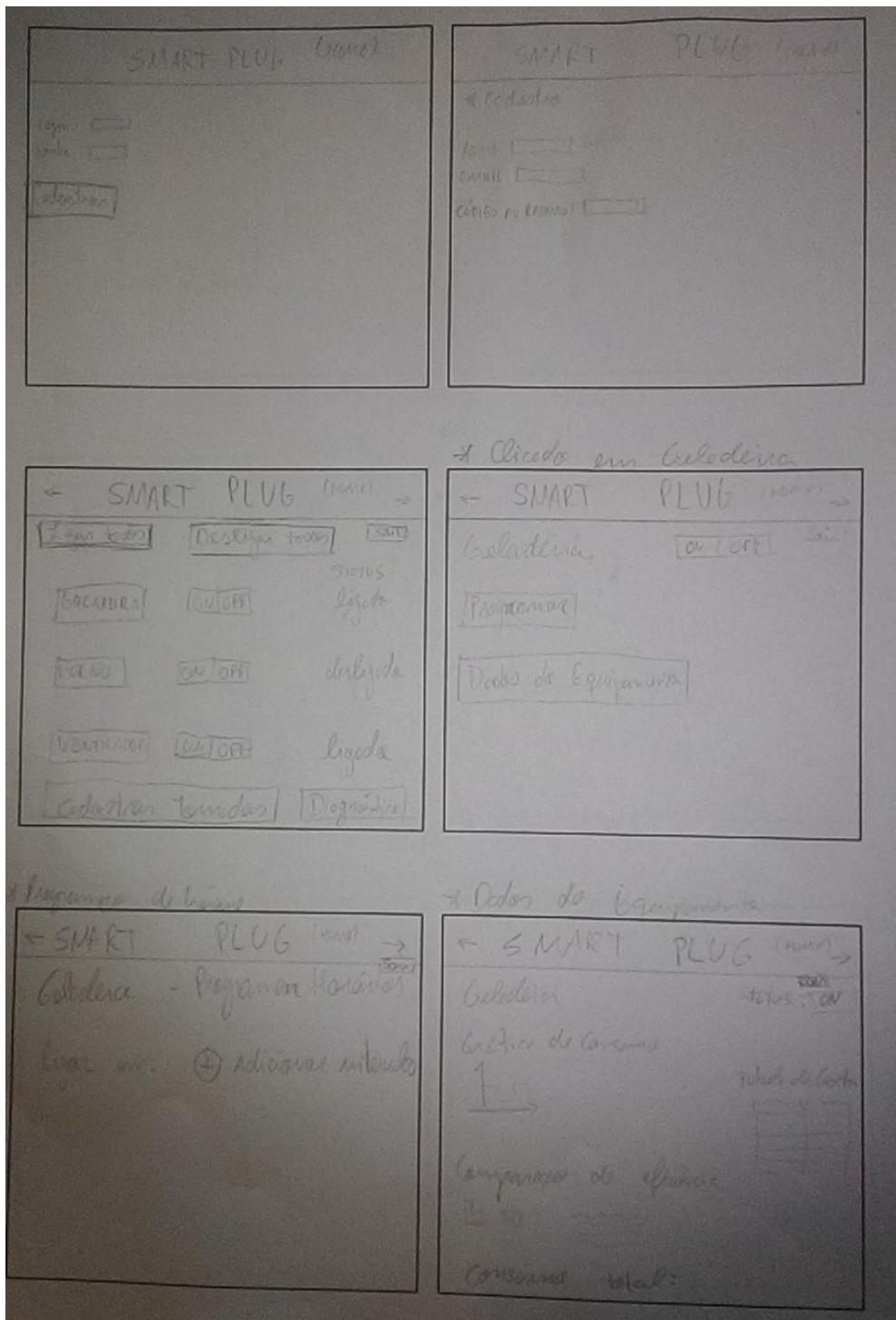
No: 13	(<input checked="" type="checkbox"/>) Funcional	(<input type="checkbox"/>) Não funcional
Requisito: Funcionar em um Browser --- Chrome		
Descrição: O sistema deve estar operante quando acessado através do navegador Chrome.		
Prioridade: (<input checked="" type="checkbox"/>) Alta (<input type="checkbox"/>) Média (<input type="checkbox"/>) Baixa		
Status: (<input type="checkbox"/>) Proposto (<input type="checkbox"/>) Aprovado (<input type="checkbox"/>) Incorporado (<input checked="" type="checkbox"/>) Validado		
Estabilidade: (<input checked="" type="checkbox"/>) Alta (<input type="checkbox"/>) Média (<input type="checkbox"/>) Baixa		
Origem: (<input checked="" type="checkbox"/>) Usuário (<input type="checkbox"/>) Cliente (<input type="checkbox"/>) Competidor (<input type="checkbox"/>) Interna		
Requisitos associados: 13		

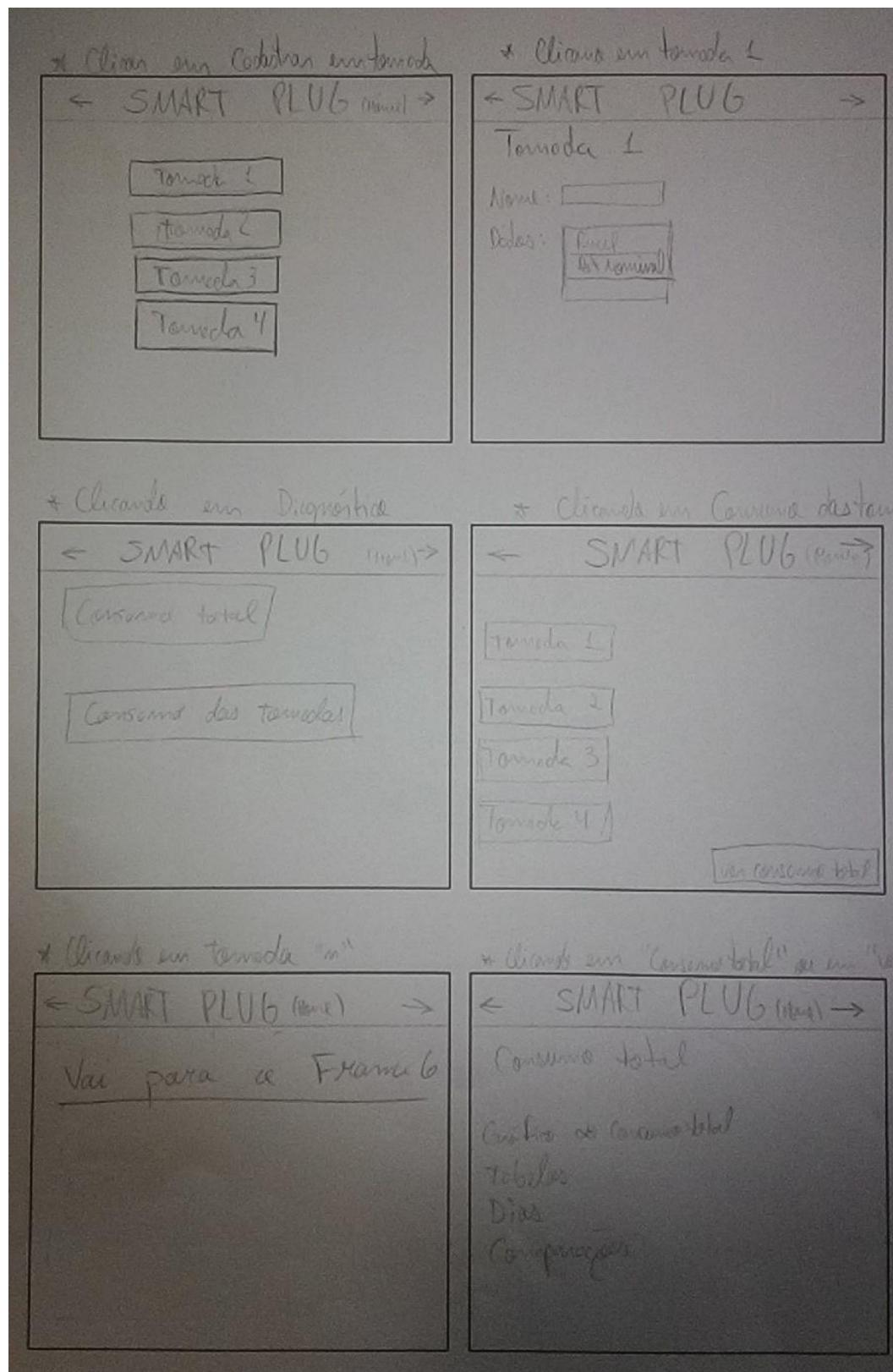
No: 14	(<input type="checkbox"/>) Funcional	(<input checked="" type="checkbox"/>) Não funcional
Requisito: Interface APP		
Descrição: O sistema pode estar acessível através de um APP.		
Prioridade: (<input type="checkbox"/>) Alta (<input type="checkbox"/>) Média (<input checked="" type="checkbox"/>) Baixa		
Status: (<input checked="" type="checkbox"/>) Proposto (<input type="checkbox"/>) Aprovado (<input type="checkbox"/>) Incorporado (<input type="checkbox"/>) Validado		
Estabilidade: (<input checked="" type="checkbox"/>) Alta (<input type="checkbox"/>) Média (<input type="checkbox"/>) Baixa		
Origem: (<input checked="" type="checkbox"/>) Usuário (<input type="checkbox"/>) Cliente (<input type="checkbox"/>) Competidor (<input type="checkbox"/>) Interna		
Requisitos associados: 12		

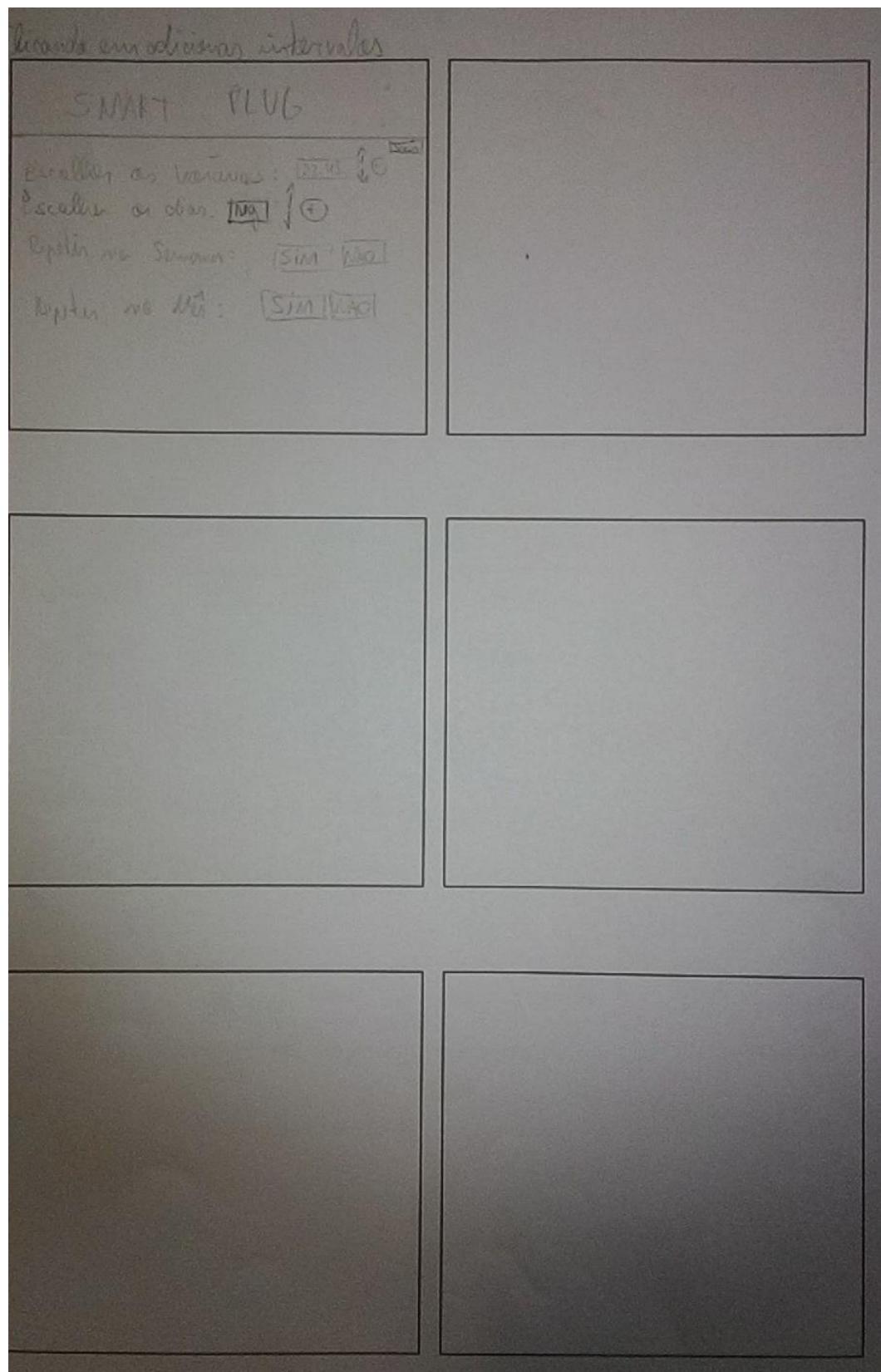
14. APÊNDICE B – WIRE FRAME

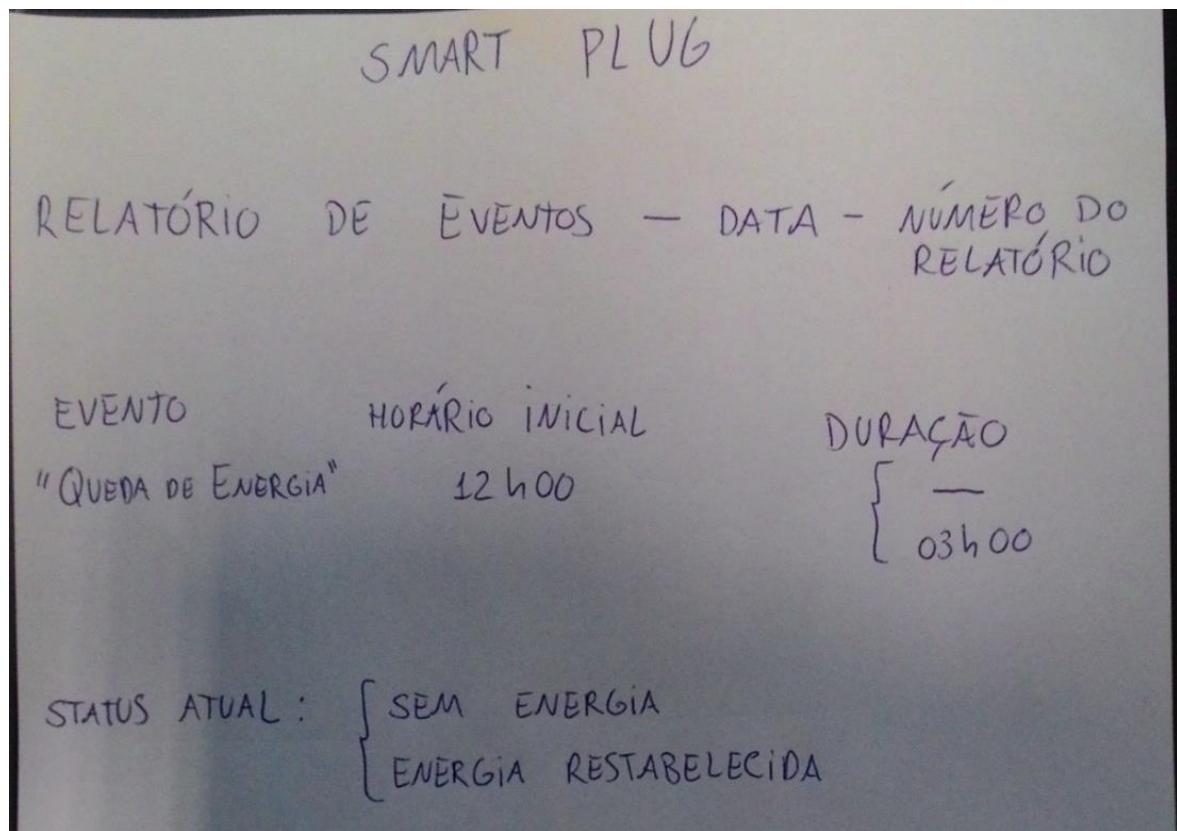
Abaixo o esboço do Wire Frame criado para o desenvolvimento da interface da Smart Plug:

14.1. Wire Frame das telas iniciais







14.2. Wire Frame dos relatórios

SMART PLUG

RELATÓRIO DE CONSUMO - DATA - NÚMERO DO
TIPO: DIÁRIO RELATÓRIO

CONSUMO TOTAL NO DIA: —

GRÁFICO DE CONSUMO NO DIA

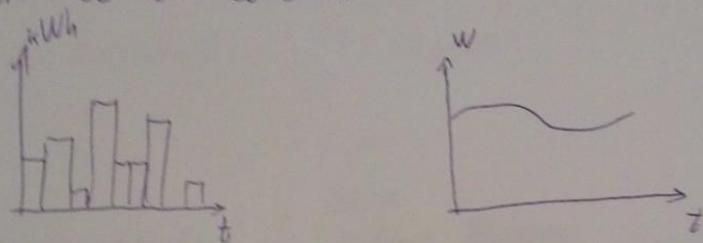


GRÁFICO DE CONSUMO DAS CÓMODOS

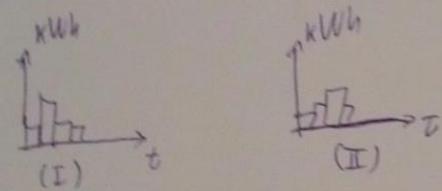
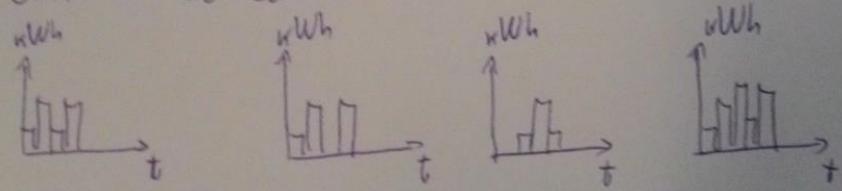


GRÁFICO DE CONSUMO DAS TOMADAS



SMART PLUG

RELATÓRIO DE CONSUMO - DATA - NÚMERO DO RELATÓRIO

TIPO: MENSAL

CONSUMO TOTAL NO MÊS: _____

CONSUMO TOTAL NO MÊS ANTERIOR: _____

VARIAÇÃO DE CONSUMO: _____

GRÁFICO DE CONSUMO TOTAL — MÊS ATUAL

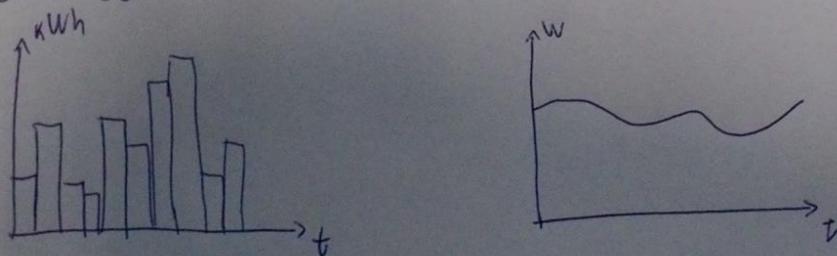


GRÁFICO DE CONSUMO DOS COMODOS

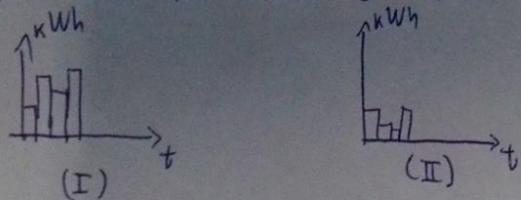
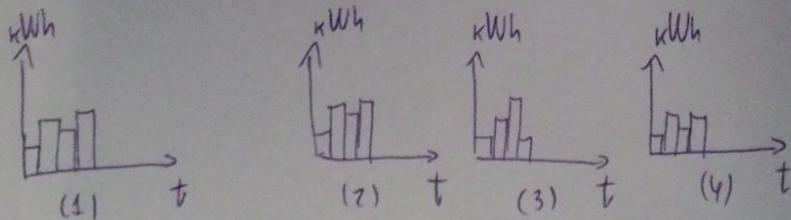


GRÁFICO DE CONSUMO DAS TOMADAS



OBSERVAÇÕES: "Houve variação de $\pm 10\%$ no consumo total em comparação com o mês anterior."

15. APÊNDICE C – INFORMAÇÕES SOBRE O CD

No CD há uma pasta com nome PCS2502_GRUPO_12_Monografia e dentro desta pasta está contido o arquivo com a monografia.