SMRE - Sistema de Economia de Recursos Elétricos

Dante José Lacerda Santos Segundo *
Instituto Federal da Bahia
Rua Emídio dos Santos, S/N
Barbalho, Salvador Bahia
dantelacerda01@gmail.com

Manoel C. M. Neto †
Instituto Federal da Bahia
Rua Emídio dos Santos, S/N
Barbalho, Salvador Bahia
manoelnetom@ifba.edu.br

ABSTRACT
This paper presents the development of SMRE, a project which aims at saving energy with the aid of a low cost microcontroller that performs measurements. The user will have available a web system to monitor your energy points and view information in graph form.

Keywords
Energy, Low Cost, Microcontroller, Graph

1. INTRODUÇÃO
O uso abusivo da energia elétrica é um assunto muito delicado. A produção de energia a partir de recursos fósseis é a principal fonte de emissões de gases que provocam o efeito estufa e consequentemente o aquecimento global.

A sociedade contemporânea vive dependente do consumo energético tratando a energia elétrica em seu cotidiano como um recurso infinito, suprindo as necessidades diárias sem se preocupar com as consequências geradas em longo prazo. Porém, se não utilizada com cuidado, a energia pode terminar e muitos poderão sofrer as consequências dos apagões que já atingem cidades brasileiras, a concentração de gases poluentes nos aglomerados urbanos e as mudanças climáticas advindas do aquecimento global.

Atualmente, 35,9 por cento da energia elétrica produzida no Brasil é utilizada pela indústria; 30,1 por cento pelo setor de transportes; 9,4 por cento pelas residências; 4,4 por cento pelo setor comercial e de serviços; 4 por cento pelo setor agronômico e 8,9 por cento são consumidos pelas próprias empresas geradoras de energia. Ou seja, a economia de energia doméstica deve ser uma em muitas atitudes que devem ser repensadas.

*Aluno do curso de Análise e Desenvolvimento de Sistemas (ADS)

†Doutor em Ciência da Computação e Professor do Curso de Análise e Desenvolvimento de Sistemas

É pensando nestes problemas que essa monografia foi desenvolvida. O SMRE é uma solução de baixo custo que utiliza Arduino para realizar medições de energia que irão ajudar as pessoas a se conscientizarem do gasto desnecessário que estão tendo. O SMRE dispõe também de um sistema *web* que será utilizado para mostrar em formas de gráfico como está o consumo de um ponto específico medido.

Este artigo será organizado em seções e apresentado na seguinte ordem:

1. Na seção 2 será explicado o referencial teórico.
2. Na seção 3 serão apresentados os trabalhos relacionados.
3. Na seção 4 será mostrada a tecnologia utilizada para a realização deste trabalho.
4. Na seção 5 será mostrada a metodologia de desenvolvimento do projeto.
5. Na seção 6 será detalhado o módulo e mostradas as principais funcionalidades.
6. Nas seções 7 e 8 será abordadas as metodologias de testes utilizadas e os resultados obtidos durante a validação do sistema.
7. Nas seções 9 à 11 serão apresentadas as conclusões, os trabalhos futuros e as referências utilizadas.

2. REFERENCIAL TEÓRICO
Antes de desenvolver o SMRE, foram feitas diversas pesquisas para entender um pouco mais o problema que enfrentaríamos e como seria possível sana-los de uma forma prática e sem muitos gastos. Para isso foram levantados alguns assuntos importantes que serão mostrados e explicados nas subseções a seguir.

2.1 Energia no mundo
O gasto de energia no Brasil vem crescendo muito com o passar dos anos. Em meio à era da tecnologia, tudo novo necessita de energia. Se analisarmos como o mundo vem se desenvolvendo rápido, surgindo novos produtos sempre, chegaremos a uma conclusão lógica de que o gasto será muito maior com o passar dos anos. Mais de 90 por cento da energia é produzida nas hidrelétricas, que dependem de água em níveis adequados em seus reservatórios para gerar energia.

Mas, em anos que há ausência de chuvas, prejudica a oferta de energia. Por isso a meta é sempre para que os consumidores reduzam o consumo de energia (principalmente quando é desnecessário).

A geração de energia depende de reservatórios de água e eles devem term níveis adequados. As regiões Sudeste/Centro-Oeste e Nordeste estão se mantendo dentro das expectativas do Operador Nacional do Sistema Elétrico (ONS) para que o País sempre tenha o mínimo de água que garanta segurança para a geração de energia necessária.

A má distribuição da água, desmatamento, desperdício e conflitos de uso são alguns dos problemas que tornam a escassez de água, não mais uma preocupação de ambientalistas e técnicos alarmistas, mas reconhecida a nível nacional. Em seu pronunciamento sobre o Dia Mundial do Meio Ambiente, ele alertou sobre a necessidade de preservação das fontes naturais.

Essa situação é resultado de um padrão de desenvolvimento sem planejamento, que consome muita água e energia e não protege os mananciais. A distribuição dos recursos hídricos no país maximiza o problema, já que 68 por cento da água brasileira está na região Norte, onde vive apenas 7 por cento da população. Ao contrário, as regiões Sudeste e Nordeste concentram 43 por cento e 29 por cento da população, respectivamente, mas contam com somente 7 por cento (SE) e 3 por cento (NE) da disponibilidade hídrica.

Junto com esses dados sobre a distribuição dos recursos hídricos é importante fazer uma comparação com um gráfico de consumo de energia dividido por regiões mostrado na Figura 1:

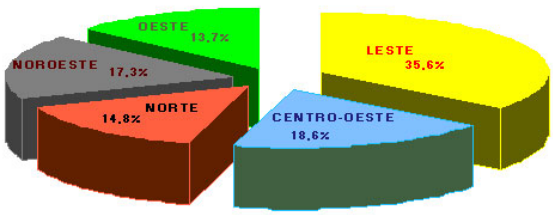


Figura 1: Consumo por região do país

Com tudo isso, temos risco de sermos pegos por apagões. Desde o começo dos anos 70, o Brasil convive com o risco de apagões. Nesses trinta anos, já foram registrados mais de cinqüenta blecautes. O mais importante foi o de 1999, quando faltou luz durante quatro horas. A falta de energia por alguns minutos no ano é normal em todo o mundo. Em média, cada casa fica até dez horas sem luz por ano no Brasil.

Diante da crise energética que estamos passando, não nos resta outra alternativa a não ser economizarmos os 20 por cento de gastos com energia estabelecidos pelo governo ou então, correremos o risco de ficarmos totalmente no escuro.

O povo brasileiro tem um gasto de energia elevado por conta do mal uso de seus aparelhos ou má organização da casa. É importante salientar a forma que consumimos mais energia em nossas casas e podemos ver isso de forma clara na Figura 2:

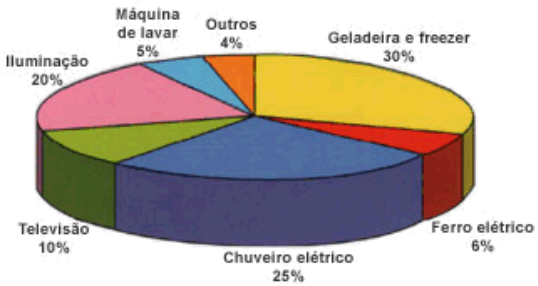


Figura 2: Gastos de energia em casa

Existem outras fontes energéticas inexploradas que poderiam sanar os problemas. Uma dessas fontes é a energia geotérmica. O potencial de energia geotérmica é quase ilimitado e pode facilmente fornecer energia suficiente para todas as necessidades mundiais. Mesmo se aproveitarmos apenas um por cento da energia geotérmica do exterior da crosta terrestre, teríamos disponível aproximadamente quinhentas vezes a quantidade de energia contida em todas as reservas de petróleo e gás no mundo. Esta fonte de energia emite pouco ou nenhum enxofre em comparação com os combustíveis fósseis usados em usinas e não emite óxido de nitrogênio. Além disso, instalações geotérmicas exigem muito pouco terreno em comparação a outras centrais elétricas. A perfuração de poços geotérmicos tem um impacto ambiental muito menor do que outros recursos energéticos e não há necessidade de abertura de minas, túneis ou armazenamento de resíduos.

2.2 Microcontroladores

Os microcontroladores são microprocessadores que podem ser programados para funções específicas. Em geral, eles são usados para controlar circuitos e, por isso, são comumente encontrados dentro de outros dispositivos, sendo conhecidos como "controladores embutidos". A estrutura interna de um microcontrolador apresenta um processador bem como circuitos de memória e periféricos de entrada e saída.

Cerca de 50 por cento dos microcontroladores vendidos são controladores "simples", outros 20 por cento são processadores de sinais digitais (DSPs) mais especializados. Os microcontroladores podem ser encontrados em praticamente todos os dispositivos eletrônicos digitais que nos cercam: teclado do computador, dentro do monitor, disco rígido, relógio de pulso, rádio relógio, máquinas de lavar, forno de micro-ondas, telefone, etc. Certamente eles foram tão ou mais importantes para a revolução dos produtos eletrônicos que os computadores. Eles permitiram a evolução de equipamentos que há anos não evoluíam como os motores a combustão, que agora com o novo controle eletrônico podem funcionar com sistema biocombustível, poluindo menos e as máquinas fotográficas, que migraram de processos químico/mecânico a circuitos com microcontroladores+Sensores Digitais+Memória.

Com freqüências de *clock* de poucos *MHz (Megahertz)* ou talvez menos, os microcontroladores operam a uma freqüência muito baixa se comparados com os microprocessadores atuais, no entanto são adequados para a maioria das aplicações. O seu consumo em geral é relativamente pequeno, normalmente na casa dos *miliwatts* e possuem geralmente habilidade para entrar em modo de espera (*Sleep* ou *Wait*) aguardando por uma interrupção ou evento externo, como por exemplo, o acionamento de uma tecla, ou um sinal que chega via uma interface de dados. O consumo destes microcontroladores em modo de espera pode chegar à casa dos *nanowatts*, tornando-os ideais para aplicações onde a exigência de baixo consumo de energia é um fator decisivo para o sucesso do projeto.

Os microcontroladores se diferenciam dos processadores, pois além dos componentes lógicos e aritméticos usuais de um microprocessador de uso geral, o microcontrolador integra elementos adicionais em sua estrutura interna, como memória de leitura e escrita para armazenamento de dados, memória somente de leitura para armazenamento de programas, EEPROM para armazenamento permanente de dados, dispositivos periféricos como conversores analógico/digitais (ADC), conversores digitais/analógicos (DAC), Portas de entrada e Saída digitais(I/O) para propósito geral.

Podemos ver a evolução dos microcontroladores e microprocessadores na Figura 3. Apenas os modelos mais comuns foram colocados, mas existem milhares de famílias no mercado.

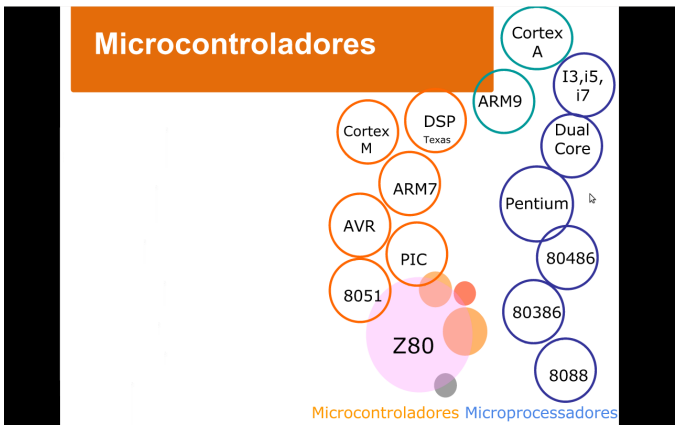


Figura 3: Evolução dos Microcontroladores e Microprocessadores

Um microcontrolador geralmente é pequeno e barato[2]. Os componentes são escolhidos para minimizar o tamanho e serem os mais econômicos possíveis. Um microcontrolador geralmente é feito para ser mais robusto de alguma forma. O microcontrolador que controla um motor de carro, por exemplo, tem que trabalhar em temperaturas extremas que um computador normalmente não suporta. Um microcontrolador de carro no Alaska tem que funcionar bem em temperaturas de -34°C, enquanto o mesmo microcontrolador no Rio de Janeiro pode ter de operar a 42°C. Quando se adiciona o calor que é gerado naturalmente pelo motor, a temperatura pode atingir de 65 a 80°C no compartimento do motor. Por outro lado, um microcontrolador embutido dentro de um VCR não precisa ser tão resistente assim.

2.3 Arduino

O Arduino é uma placa de controle I/O baseada no microcontrolador Atmega (Atmel), que serve de controle para diversos outros sistemas, porém o diferencial desta placa é que ela é desenvolvida e aperfeiçoada por uma comunidade que divulga as suas placas e seus códigos de aplicação, pois a concepção da placa é open-source.

Esta placa foi projetada com finalidades educativas tendo como intuito ajudar os designers e artistas a criarem obras interativas sem terem muitos conhecimentos em eletrônica, mas pelo fato de ter o seu esquema e software de programação *open-source*, ela acabou chamando a atenção dos técnicos de eletrônica, que começaram a aperfeiçoá-la e a criar aplicações mais complexas. A idéia básica de uso é introduzir sensores ou chaves nos conectores de entrada, e LEDs, displays, relés, motores e outros dispositivos que possam ser acionados pelos sinais de saída. Depois, é só programar o microcontrolador utilizando o *software open-source* desenvolvido pela equipe do Arduino e fazer os testes.

Sua placa consiste em um microcontrolador *Atmel AVR de 8 bits*, com componentes complementares para facilitar a programação e incorporação para outros circuitos. Um importante aspecto é a maneira padrão que os conectores são expostos, permitindo o CPU ser interligado a outros módulos expansivos, conhecidos como *shields*. Os Arduinos originais utilizam a série de *chips megaAVR*, especialmente os ATmega8, ATmega168, ATmega328 e a ATmega1280; porém muitos outros processadores foram utilizados por clones deles. [1]

Com ele é possível fazer algumas automações que não excedam o número das portas disponíveis. A quantidade de aplicações encontradas na internet utilizando o Arduino é enorme.

A possibilidade de escrever e apagar o código do microcontrolador faz do Arduino uma ótima ferramenta de teste ou para usos específicos de curta duração ou futuras modificações, ou até mesmo para fazer atualizações em um determinado produto, uma vez que é só trocar o microcontrolador em funcionamento por um novo microcontrolador com a programação nova.

Como o Arduino é *open-source*, a documentação também é colaborativa, ou seja, o leitor pode obter a literatura necessária no próprio site da equipe criadora do software ou procurar a documentação das variações do Arduino e suas respectivas informações. Por isto é sempre bom escolher bem o tipo de variação do Arduino, utilizando como critério a documentação, pois sendo uma comunidade aberta a documentação é praticamente necessária para cada projeto criado.

Além disso, se você quiser, pode criar sua própria placa arduino. Os projetos de referência do Arduino estão disponíveis na página de *hardware*. Eles são licenciados sob *Creative Commons Attribution Share-Alike*, portanto você é livre para usá-los e adaptá-los às suas necessidades próprias sem precisar pedir autorização ou pagar taxas. Na Figura 4[10]

podemos ver o esquema disponibilizado para criar a placa.

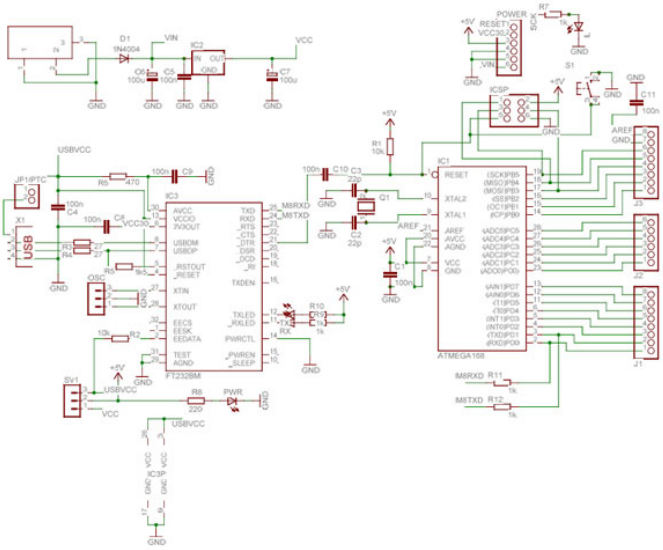


Figura 4: Esquema de montagem da placa arduino

Existem alguns tipos de Arduino e eles serão explicados nos tópicos a seguir[11]:

2.3.1 *Arduino Mega*

O Arduino Mega é uma placa baseada no microcontrolador ATmega1280 . Ele tem 54 digitais de entrada / saída pinos (dos quais 14 podem ser usados como saídas PWM), 16 entradas analógicas, 4 UARTs (portas seriais de hardware), um oscilador de cristal 16 MHz, uma conexão USB, uma tomada de poder, um cabeçalho ICSP, e um botão de reset. Ele contém tudo o necessário para apoiar o microcontrolador, basta conectá-lo a um computador com um cabo USB ou ligá-lo com um adaptador AC para DC ou bateria para começar. A Mega é compatível com a maioria dos escudos projetado para o Arduino Duemilanove ou Diecimila. Na Figura 5 temos o Mega.

Caracteristicas

Microcontrolador	ATmega1280
Tensão operacional	5V
Tensão de entrada (recomendado)	7-12V
Tensão de entrada (limites)	6-20V
Digital I / O Pinos	54 (dos quais 14 oferecem saída PWM)
Analógica dos Pinos de Entrada	16
DC Corrente por I / O Pin	40 mA
Corrente DC 3.3V para Pin	50 mA
Memória Flash	128 KB, dos quais 4 KB utilizado pelo gerenciador de boot
SRAM	8 KB
EEPROM	4 KB
Velocidade do relógio	16 MHz

Arduino Mega

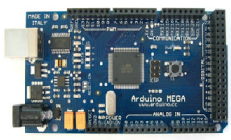


Figura 5: Características do Arduino Mega

2.3.2 *Arduino Dumilanove*

O Duemilanove Arduino (”2009”) é uma placa baseada no microcontrolador ATmega168 ou ATmega328 . Tem 14 digitais de entrada / saída pinos (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um oscilador de cristal 16 MHz, uma conexão USB, uma tomada de poder, um cabeçalho ICSP, e um botão de *reset*. Ele contém

Caracteristicas

Microcontrolador	ATmega168
Tensão operacional	5V
Tensão de entrada (recomendado)	7-12V
Tensão de entrada (limites)	6-20V
Digital I / O Pinos	14 (dos quais 6 oferecem saída PWM)
Analógica dos Pinos de Entrada	6
DC Corrente por I / O Pin	40 mA
Corrente DC 3.3V para Pin	50 mA
Memória Flash	16 KB (ATmega168) ou 32 KB (ATmega328), dos quais 2 KB utilizado pelo gerenciador de boot
SRAM	1 KB (ATmega168) ou 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) ou 1 KB (ATmega328)
Velocidade do relógio	16 MHz

Arduino Dumilanove

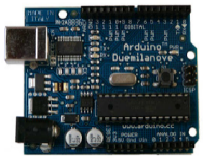


Figura 6: Características do Arduino Dumilanove

tudo que é necessário para apoiar o microcontrolador, basta conectá-lo a um computador com um cabo USB ou ligá-lo com um adaptador AC para DC ou bateria para começar a utilizar 6 temos o Dumilanove.

2.3.3 *Arduino UNO*

Para o desenvolvimento do SMRE foi utilizado um Arduino do tipo UNO.

O Uno Arduino é uma placa baseada no microcontrolador ATmega328 (folha de dados). Tem 14 digitais de entrada / saída pinos (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um oscilador de cristal 16 MHz, uma conexão USB, uma tomada de poder, um cabeçalho ICSP, e um botão de reset. Ele contém tudo que é necessário para apoiar o microcontrolador, basta conectá-lo a um computador com um cabo *USB* ou ligá-lo com um adaptador AC para DC ou bateria para começar. O Uno difere de todas as placas anteriores na medida em que não usa o chip de *FTDI driver USB-to-serial*. Em vez disso, ele apresenta o Atmega8U2 programado como um conversor *USB-to-serial*. Revisão 2 do conselho Uno tem um resistor de puxar a linha HWB 8U2 para a terra, tornando mais fácil de colocar em modo DFU. Na Figura 7 temos o UNO.

Caracteristicas

Microcontrolador	ATmega328
Tensão operacional	5V
Tensão de entrada (recomendado)	7-12V
Tensão de entrada (limites)	6-20V
Digital I / O Pinos	14 (dos quais 6 oferecem saída PWM)
Analógica dos Pinos de Entrada	6
DC Corrente por I / O Pin	40 mA
Corrente DC 3.3V para Pin	50 mA
Memória Flash	32 KB (ATmega328), dos quais 0,5 KB utilizado pelo gerenciador de boot
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
	16 MHz

Arduino UNO

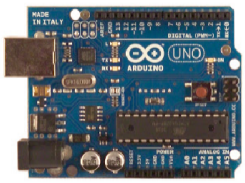


Figura 7: Características do Arduino UNO

2.3.4 *Arduino severino*

Esse é o arduino para quem não tem muita condição para comprar um desses citados acima pois essa é a versão mais barata do arduino, podendo até ser montado por quem não tem muita experiencia em eletronica. Essa é a versão serial do arduino. Na Figura 8 temos o Severino.

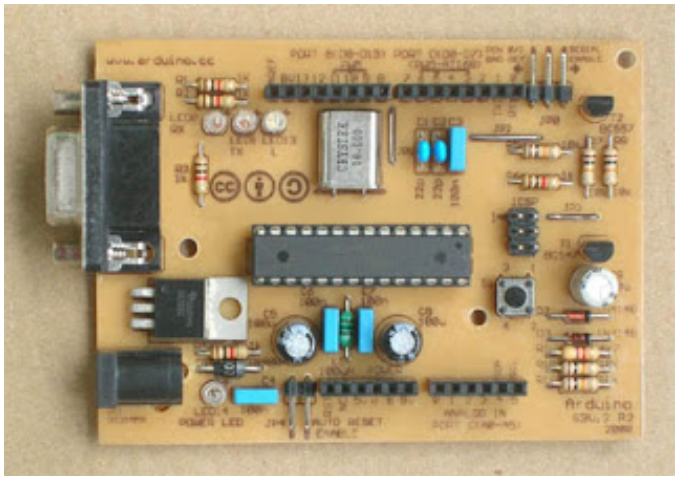


Figura 8: Arduino Severino

2.4 Computação Ubíqua

A computação ubíqua abrange um modelo de computação no qual usuários móveis, serviços e recursos são capazes de descobrir outros usuários, serviços e recursos. A idéia básica é que a computação mova-se para fora das estações de trabalho e computadores pessoais e torne-se pervasiva em nossa vida cotidiana. Uma série de princípios e tecnologias que estão por trás desse paradigma computacional devem ser mantidos em mente para o desenvolvimento de aplicações e dispositivos ubíquos.[4]

Nos dias atuais, com a convergência das tecnologias de rádio, dos microprocessadores e dos dispositivos eletrônicos digitais de uso pessoal, tornou-se possível fazer com que os dispositivos com certo poder de processamento computacional, tanto móveis quanto estacionários, cooperem entre si para fornecer ao usuário acesso instantâneo a novos serviços de forma transparente. Essa característica levou ao surgimento do conceito de ubiquidade na computação. A computação ubíqua abrange um modelo de computação no qual usuários móveis, serviços e recursos são capazes de descobrir outros usuários, serviços e recursos. Nesse paradigma computacional, as responsabilidades pela execução de determinado serviço necessário para o usuário são distribuídas entre uma variedade de dispositivos, cada qual com suas tarefas e funcionalidades específicas.

A computação ubíqua, por sua vez, integra mobilidade em-larga escala com a funcionalidade da computação pervasiva, isto é, qualquer dispositivo computacional sob posse de um usuário em movimento, pode construir, dinamicamente, modelos computacionais do ambiente em que está inserido e configurar seus serviços dependendo da necessidade. A idéia básica é que a computação move-se para fora das estações de trabalho e PCs e torna-se pervasiva no cotidiano dos indivíduos. Marc Weiser, idealizador da computação ubíqua visionou há um pouco mais de uma década que, no futuro, computadores estariam presentes nos mais comuns objetos do dia-a-dia. Neste mundo de Weiser, é necessário aprender a conviver com computadores, e não somente interagir com eles.

O sistema para gerenciar a computação ubíqua pode ser de pequeno porte, apenas com conexão de rede sem fio a um computador de grande capacidade. Um exemplo prático desta funcionalidade são casas que podem ser controladas por meio da tecnologia ubíqua; iluminação pode ser acionada, ligar ou desligar televisores e equipamentos eletroeletrônicos, até o monitoramento de pessoas acamadas ou controlar os itens da dispensa pelo prazo de validade, entre outros.

Na Figura 9 podemos ver o posicionamento de cada um dos tipos[6]:

Figura 9: Relação entre os tipos de computação

Além disso, na Figura 10 podemos ver as dimensões dos 3 tipos de computação apresentados. O grau de embarcamento refere-se, de forma geral, o grau de inteligência dos computadores, embutidos em um ambiente pervasivo, para detectar, explorar e construir dinamicamente modelos computacionais de seus ambientes.

	Computação Pervasiva	Computação Móvel	Computação Ubíqua
Mobilidade	Baixa	Alta	Alta
Grau de "embarcamento"	Alto	Baixo	Alto

Figura 10: Dimensões das computações

O futuro breve da Computação Ubíqua é a interação total, não somente de celulares, computadores, televisores ou e-books, mas também dispositivos como mesa, cadeira, chaveiro, geladeira, caneta, agenda e outros itens que não pertencem a informática. Através da função *cloud computing* (computação em nuvem) e interligação de internet, todos os dispositivos utilizados no dia-a-dia poderão ser interligados e adaptados a necessidade de cada pessoa.

3. TRABALHOS RELACIONADOS

Nesta seção será mostrado trabalhos relacionados ou que sejam parecidos com o SMRE. A ideia de economizar energia é muito comum, porém existem diversos projetos diferentes com um objetivo comum. A ideia principal deste projeto aqui é ter baixo custo, que é uma ótima característica que não é vista em outros projetos.

3.1 Economia IF-SC

Existe um projeto[9] que foi utilizado em um TCC no IF-SC que também utiliza economia de energia. A diferença é que ao invés de utilizar arduino, foi usado um transdutor de potência elétrica. Neste equipamento o valor da potência ativa de entrada será convertido em um sinal de corrente analógico, proporcional à potência medida.

O sistema do IF-SC tem como objetivo o controle e o gerenciamento do consumo. Através de dados históricos do sistema é possível compreender o perfil de consumo de uma edificação, tornando-se uma ferramenta útil no planejamento da manutenção, na previsão de ampliação da demanda e na avaliação do impacto de ações de economia de energia da instalação. O campus São José do IFSC tem uma despesa média mensal em energia elétrica em torno de 15.000 reais. Avaliando as contas dos últimos doze meses pode ser verificado que em alguns períodos de férias escolares o consumo de energia é da mesma ordem dos meses letivos. Além disso, percebe-se que a gestão do comando de cargas, como iluminação e ar condicionado não é realizada de maneira adequada. Para o desenvolvimento do projeto propõe-se a instalação de um transdutor de potência elétrica na subestação de energia do campus. Neste equipamento o valor da potência ativa de entrada será convertido em um sinal de corrente analógico, proporcional à potência medida. O sinal de saída do transdutor de potência será transmitido em valores digitais para um computador. No computador um programa será elaborado para, a partir dos sinais de medição, gerar os gráficos da energia elétrica consumida ao longo do tempo. Uma estrutura de banco de dados foi avaliada para aplicação no armazenamento ordenado do histórico do consumo. Foi desenvolvido um portal na web para visualização dos gráficos de demanda e geração de relatórios como em um sistema supervisório comercial. Este portal poderá ser acessado de qualquer ponto conectado à Internet.

A seguir são descritos os principais aspectos destes temas. Os transdutores analógicos de potência têm por finalidade converter a potência ativa de um sistema em um sinal contínuo (de tensão ou corrente) e isolado galvanicamente das entradas. Neste equipamento o valor da potência ativa de entrada será convertido em um sinal de corrente analógico entre 4mA a 20mA, proporcional à potência medida. Este sinal de saída do transdutor será transmitido em valores digitais para um computador. Há várias formas de realizar a transmissão do sinal: cabo coaxial ou par trançado, *wireless*, rede elétrica etc., as quais serão avaliadas no início do desenvolvimento do projeto. A tecnologia de comunicação sem fio ZigBee seria uma alternativa. O transdutor de potência teria ligação com o módulo Xbee via cabo serial, depois o sinal seria transmitido via wireless para outro módulo Xbee conectado ao computador. Porém, esta possibilidade deverá ser testada, uma vez que a tecnologia Zigbee não foi projetada para operar em ambientes expostos a altos níveis de interferência eletromagnética.

Na Figura 11 podemos ver um transdutor:

3.2 Inversor de frequência de energia

Neste projeto[16] é importante observar as máquinas de Fluxo, que são equipamentos destinados a movimentação de fluidos como líquidos e gases. As principais máquinas de fluxo são

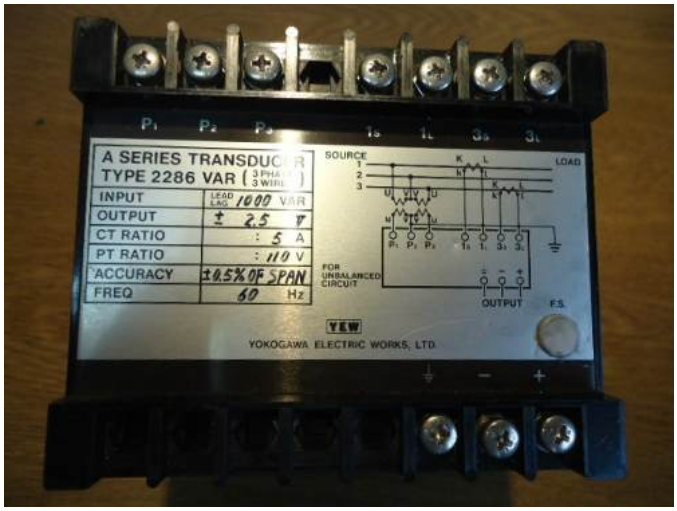


Figura 11: Transdutor de potência elétrica

ventiladores ou exaustores e bombas centrífugas. A redução no consumo de energia elétrica nas aplicações de bombeamento ou de pressurização em caso de ventiladores é realizada através da redução na velocidade do motor, ao invés de se utilizar válvulas ou dampers para restrição de vazão e/ou pressão.

As bombas centrífugas usadas no bombeamento de água e/ou ventiladores, são equipamento de torque variável, ou seja, o torque necessário para a movimentação do eixo da bomba cai ao quadrado da redução da velocidade. No caso da potência absorvida da rede, a redução é muito maior. A potência cai ao " cubo " da redução de velocidade. Em aplicações onde é necessário variar a vazão de uma bomba é normal a utilização de válvulas a qual reduz a vazão restringindo a área de passagem do fluido. Nesse caso bomba continua trabalhando em rotação nominal e consumindo potência nominal, sendo que a válvula se encarregará de dissipar a energia excedente. No caso de um ventilador ou exaustor é normal a utilização de um damper para restrição de vazão mantendo o motor trabalhando em rotação nominal. Em ambos os casos a vazão é reduzida com a perda de energia nas válvulas ou no damper. Com a utilização de um inversor de freqüência podemos reduzir a velocidade do motor para redução da vazão com sem a utilização de válvulas e dampers, economizando assim a energia que era perdida nos mesmos.

Na Figura 12 , o retângulo amarelo representa a potência elétrica consumida pelo motor da bomba ou ventilador, quando operado em rotação nominal, o retângulo laranja representa a potência consumida em rotação reduzida. Podemos notar que com uma pequena redução na rotação temos uma grande redução na potência elétrica consumida pela máquina de fluxo. Esse efeito deve-se ao fato de a potência absorvida pela bomba ou ventilador variar ao cubo da rotação.

Existem algumas vantagens que foram citadas no trabalho com o inversor de fluxo, que são:

1. Economia de energia elétrica ao reduzir a rotação.

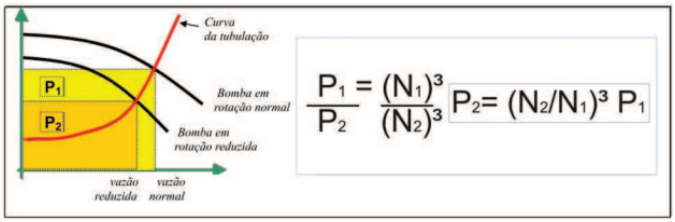


Figura 12: Representação do controle de fluxo de energia

2. Economia de energia elétrica durante as partidas do motor.
3. Economia adicional de energia com o software *"Energy Saver"*, que corrige a tensão do motor.
4. Partidas suaves reduzindo o desgaste na bomba.
5. Partidas com baixa corrente elétrica reduzindo o efeito de queda de tensão na instalação.
6. Paradas suaves sem "golpe de ariete", reduzindo desgastes da bomba e tubulação.
7. Elevação do fator de potência da instalação.

Podemos ver na Figura 13, um exemplo de inversor de frequência:

Figura 13: Inversor de frequência

3.3 Automação Residencial utilizando o Arduino

Os alunos da PUC de São Paulo, utilizando o arduino, desenvolveram um sistema que utiliza um sensor de infravermelho que ajudaria na economia de energia.[13]

O Arduino foi ligado a um computador, dois LEDs, um LDR, um sensor infravermelho composto por 2 transmissores e 2 receptores IR, um sensor de temperatura e alguns resistores. O circuito funciona com 5 Volts, fornecidos pelo próprio Arduino. O LDR aumenta ou diminui sua resistência de acordo com a incidência de luz, este foi usado para verificar se o local está escuro ou não. Funcionando junto com um sensor infravermelho, que tem o objetivo de detectar presença no local, o código no Arduino faz com que o LED do cômodo acenda se o ambiente estiver escuro e houver presença.

Além disso, o Arduino monitora a temperatura do local através de um sensor de temperatura (DS18B20), de acordo com a temperatura do ambiente, determinado LED acende (Verde se estiver normal e Vermelho se estiver muito quente).

O código basicamente verifica a incidência de luz sobre o LDR ao mesmo tempo que capta sinais do sensor infravermelho. Não basta estar escuro para que o LED acenda, também é preciso que haja alguém no cômodo. O sensor infravermelho envia continuamente um valor específico para o Arduino (valor obtido de acordo com distância entre o sensor e a parede), se este valor for alterado e a luz sobre o LDR for baixa, o LED acende. Qualquer outra condição não irá acende-lo. Através deste código, também monitoramos a temperatura do local. Um sensor capta a temperatura ambiente e envia esses dado para o Arduino, o código faz com que um LED verde permaneça aceso se a temperatura estiver adequada e um LED vermelho acenda se a temperatura estiver acima do normal. Na Figura 14 temos o sensor de infravermelho utilizado no projeto.

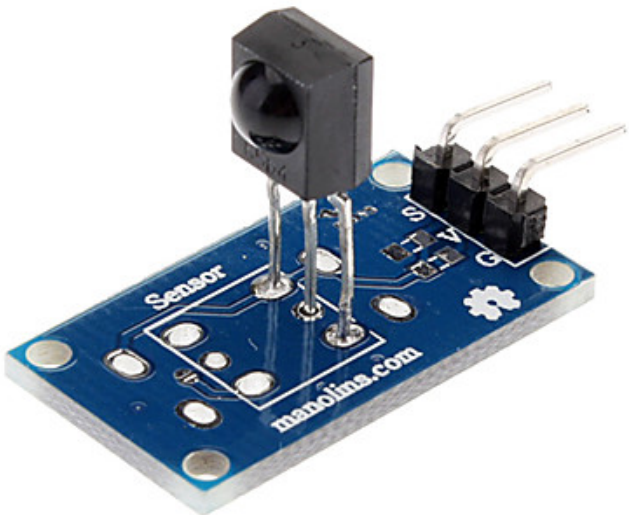


Figura 14: Sensor Infravermelho Arduino

3.4 OpenEnergyMonitor

OpenEnergyMonitor[15] é um projeto para desenvolver ferramentas *open-source* de monitoramento de energia para ajudar nos relacionamos com o uso da energia, os nossos sis-

temas de energia e o desafio da energia sustentável. Além disso ele é um sistema *end-to-end* que faz monitoramento utilizando Arduino.

O sistema OpenEnergyMonitor é composta de quatro partes principais: emonTx | emonGLCD | emonBase | emoncms. Estes podem ser montados e configurados para trabalhar em uma variedade de aplicações a partir de um monitor de energia em casa a energia solar PV de importação / exportação de monitoramento e muito mais.

Também é possível ir além de monitoramento e começar a controlar as coisas: desvio de superávit PV energia para água quente doméstica é um projeto de pesquisa openenergymonitor ativo.

Mas como construir o *OpenEnergyMonitor* ?

O sistema é composto por sensores sem fio que enviam dados em intervalos periódicos para a base-estação conectados à web. Neste momento é possível perceber: AC Eletricidade (potência aparente, corrente, tensão, potência real, fator de potência), temperatura, umidade, pulsos (de medidores de serviços públicos de saída de pulso), Elster IrDA (interface de metro utilidade direta) e energia solar PV. Ainda estão trabalhando na ampliação desta lista para incluir umidade, CO2 e outras medições de qualidade do ar.

Este guia tem os passos para a criação de um sistema básico. Você vai precisar de uma estação de base e pelo menos um nó sensor para fazê-lo funcionar.

Na Figura 15 temos uma visão geral do *OpenEnergyMonitor* que será explicada separatamente nas subseções a seguir:

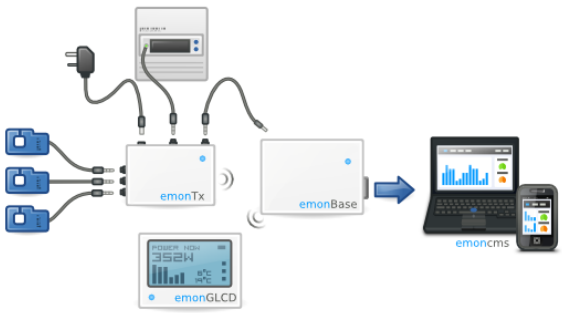


Figura 15: Visão geral do OpenEnergyMonitor

3.4.1 *Escolha e configuração de sua estação base*

É melhor começar pela configuraração da estação-base, de modo que fique pronto para receber dados de todos os nós de sensores como eles são adicionados.

A estação-base é um *Raspberry Pi* com uma placa de adaptador sem fio RFM12Pi ou NanodeRF (Arduino + *Ethernet clone*). Ambos são vendidos na loja OpenEnergyMonitor. O *Raspberry Pi* pode registrar os dados localmente para um disco rígido conectado e exibir os dados localmente (acessando-o como você acessa o seu roteador doméstico) ou pode ser configurado para encaminhar os dados para um servidor remoto, como emoncms.org. Outras opções de estação base, como a NanodeRF envia os dados diretamente para um servidor remoto externo, logo que os dados são recebidos a partir de um nó sensor.

Além do *Raspberry Pi* você vai precisar de uma pequena placa adaptadora que permite que o Pi receba dados de RF a partir dos nós de sensores sem fio, esta placa é chamada de RFM12Pi.

Na Figura 16 pode ser visto um *Raspberry Pi*

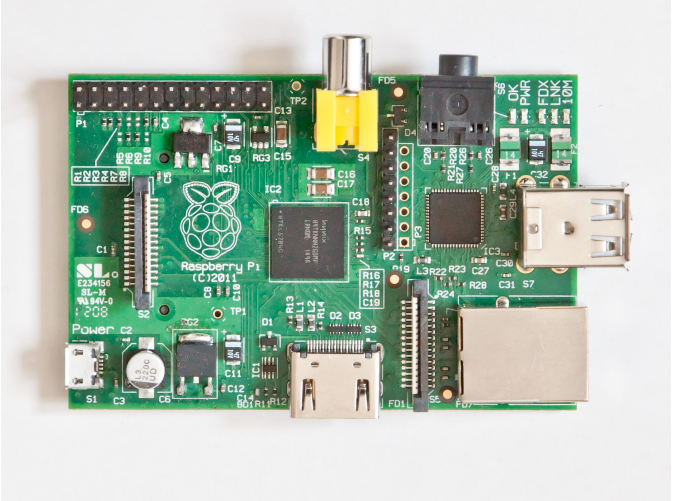


Figura 16: Raspberry Pi

3.4.2 *Escolha nós sem fio qual você gostaria de conectar-se*

Usando uma placa de hardware multiusos aberto chamado emonTx sua possível medir:

1. 4x Circuitos monofásicos CA com clipe em sensores CT. Ou um circuito de 3 fases
2. 1x Tensão AC com um adaptador de tensão AC-AC para o isolamento. Temperatura com sensores DS18B20.
3. 1x Medidor de pulso contando entrada e saída de pulso

Várias emonTx podem ser conectados a um único emonBase. Cada emonTx terá de ser dado um nó sem fio ID diferente. Se você já possui um Arduino você pode querer considerar o uso de um protetor emonTx.

emonTH - Monitoramento de Temperatura e Umidade: Alimentado por bateria de temperatura (interno) nó de monitoramento (interna / externa) e umidade.

emonGLCD - display LCD sem fio: O emonGLCD é um *open-source* de uso geral sem fio com display LCD queve pode ser adicionado ao sistema como qualquer outro nó.

Ele pode ser configurado para mostrar qualquer coisa que você gostaria de exibir a partir do consumo de energia e geração de temperatura ambiente. Ele tem dois LEDs indicadores que brilham vermelho ou verde, dependendo se você está importando ou exportando eletricidade. O emonGLCD foi construído em um sensor de temperatura; bem como receber e exibir dados de monitoramento do emonGLCD podendo transmitir a temperatura ambiente atual de volta para a estação base para a exploração com o emoncms.

O emonGLCD é realmente útil para fornecer em tempo real as informações.

Na Figura 17 pode ser visto o display LCD:

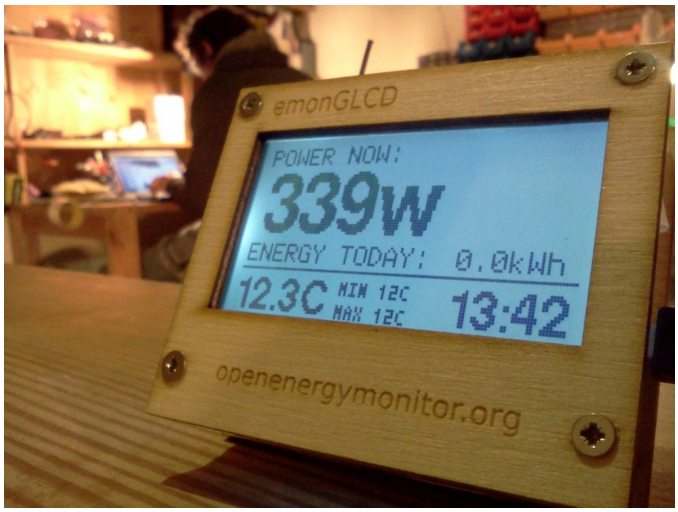


Figura 17: emonGLCD

3.4.3 Configure emoncms para registrar, processar e exibir dados monitorados

Agora que os valores de dados nó sensor são registrados na lista de entradas de emoncms o próximo passo é para processamento de entrada de configuração nessas entradas. Usando o processamento de entrada, podemos manipular os dados de entrada antes de armazená-lo no banco de dados emoncms.

Confira: O LED verde no RFM12Pi deve estar piscando (vagamente) cada vez que um pacote de RF é recebido.

4. TECNOLOGIA UTILIZADA

Nesta seção será descrita a tecnologia utilizada durante o desenvolvimento deste projeto. Por se tratar de um projeto grande, foram utilizadas diferentes tecnologias que juntas chegaram a um objetivo comum. Tais tecnologias são descritas a seguir:

1. Banco de dados PostgreSQL, que foi a melhor opção por ser um banco de dados gratuito.
2. CT sensor YHDC SCT-013-000
3. Placa Arduino UNO que foi utilizada para realizar as medições.
4. Wifi-Shield que foi utilizado no Arduino para realizar a comunicação com o servidor para salvar os dados obtidos pela placa.

5. SMRE

5.1 Levantamento de Requisitos

Durante o desenvolvimento do *software* existem diversas fases e atividades que são essenciais para a realização do seu objetivo maior: entregar o *software* funcionando corretamente dentro do orçamento e prazos previstos para o seu desenvolvimento[5].

Para chegar aos objetivos planejados, todas as atividades de desenvolvimento devem ser criteriosamente elaboradas e desenvolvidas, seja usando uma abordagem de desenvolvimento mais rica em documentação tais como o poderoso *UP (Unified Process)* ou as excelentes metodologias ágeis (XP, SCRUM, etc). Dessa forma, em qualquer uma delas encontraremos com maior ou menor rigor e formalização, atividades de análise de requisitos, design, definição de arquitetura, codificação e outras. Um trabalho consistente de análise dos requisitos, ou seja, identificar, quantificar, definir, priorizar e classificar os principais problemas que o futuro *software* deve resolver é a base de um projeto de *software* de sucesso.

O maior foco dos profissionais de T.I. são nas atividades de projeto e codificação. Isso ocorre pois na maioria das universidades o foco principal são nas matérias técnicas e científicas. Além disso, deve-se também ao perfil dos profissionais atuantes da área que se interessam pouco com assuntos administrativos. Mas Isso tem mudado bastante recentemente, pois os profissionais de T.I. têm percebido a importância da relação entre negócio e T.I. Entretanto, a atividade de levantamento de requisitos é de fundamental importância para que se construa o *software* certo, ou seja, antes de desenvolver o *software*, é preciso que todos os *stakeholders* saibam exatamente o que é esperado do aplicativo a ser construído. É muito importante também que eles saibam o que não se deve fazer também!

Um requisito é uma especificação de uma característica ou propriedade que um sistema deve possuir ou fazer, assim como suas restrições. Os requisitos podem ser definidos por diversas classificações tais como: requisitos de negócio, funcionais, não funcionais, etc.

Toda metodologia de desenvolvimento de software (MDS) propõe uma série de fases e atividades dentro do seu ciclo de vida e o encadeamento entre elas. Existem dois grupos de atividades referentes a requisitos e que podemos os chamar assim[14]:

1. Especificação de requisitos - São todas as atividades realizadas para identificar, analisar, especificar e definir as necessidades de negócio que um aplicativo deve

prover para solução do problema levantado. Requisitos que não refletem as reais necessidades dos usuários, incompletos e/ou inconsistentes, mudanças em requisitos que já foram previamente acordados e a dificuldade para se chegar a um acordo entre profissionais de T.I. e usuários são os maiores problemas enfrentados no grupo de atividades de especificação de requisitos.

2. Gestão de requisitos - Preocupa-se com a documentação, versionamento, controle de mudanças e qualidade dos requisitos levantados na fase de especificação de requisitos. Todo requisito apresenta um ciclo de vida único que acompanha a dinâmica dos negócios associados. Assim sendo, não se pode esperar que um requisito seja imutável ao longo do tempo, uma vez que o negócio do qual o requisito se desprende é dinâmico.

O gestor do projeto deve escolher os melhores analistas de sua equipe para realizar a atividade de levantamento de requisitos. Essa decisão será a mais importante pois ela tece um caminho consistente para o seguimento do projeto. É importante que o gerente do projeto envolva-se ou designe alguém capacitado para a realização da gestão de escopo do projeto, inclusive criando um método criterioso de controle de mudanças. Muitas empresas cientes do problemas ocasionados por falta da correta gestão de escopo, tentam proteger os requisitos do projeto, tornando qualquer alteração difícil, cercada de documentos e assinaturas que muitas vezes provocam o efeito contrário ao desejado. É importante que todos os *stakeholders* busquem harmonizar seus interesses para uma execução correta do projeto.

Por fim, podemos dizer que a análise de requisitos é uma das etapas mais importantes no desenvolvimento de um projeto, pois ela traça exatamente o caminho que deve ser seguido e o que não deve ser seguido e portanto será mostrado os requisitos funcionais e os não funcionais do sistema nas Figuras 18 e 19.

ID	Requisito Funcional	Ator
RF1	A aplicação deve permitir a geração de relatórios (gráficos) .	Usuário
RF2	A aplicação deve permitir, antes da geração do gráfico, escolher o tipo de relatório que quer.	Usuário
RF3	A aplicação deve obrigar o usuário a estar logado para realizar as consultas.	Usuário
RF 4	A aplicação deve mostrar em forma de gráfico os dados de energia coletados.	Usuário

Figura 18: Requisitos Funcionais do sistema

5.2 Arquitetura do Sistema

O SMRE foi dividida em 3 camadas, que seguem o padrão *Model-View-Controller (MVC)*. O padrão MVC separa a representacão da informacão da interacão do usuário com a mesma. A camada de modelo deve consistir de dados da aplicacão, regras de negócio, lógica e funções. Uma visão pode ser qualquer sada que represente dados, como os gráficos gerados na aplicação, diagramas ou páginas web. O padrão possibilita a existência de multiplas visões para um mesmo leque de dados. A camada de controle gerencia a

ID	Requisito Não-Funcional	Categoria
RNF1	Os dados mostrados na aplicação devem ser fiéis ao valor de energia passada na corrente.	Confiança
RNF2	A aplicação deverá ser acessada por um navegador moderno.	Usabilidade
RNF3	Os usuários não devem ter acessos às informações de energia de outros clientes.	Segurança
RNF 4	A aplicação será desenvolvida utilizando a linguagem Java e o banco de dados PostgreSQL.	Software

Figura 19: Requisitos Não-Funcionais do sistema

entrada, convertendo-a em comandos para camada de modelos ou de visão ou outros subsistemas. Este padrão tem como objetivos centrais a reutilização de codigo e separação de responsabilidades.[7]

O estilo arquitetural do SMRE utiliza os estilos *Object-Oriented* (Orientado a objetos - OO) e utiliza como conectores *procedures calls* e *events*. Os conectores do tipo *Procedure Call* modelam o fluxo de controle através de técnicas de invocação e realizam a transferência de dados entre os componentes envolvidos através do uso de parâmetros. Exemplos de conectores *Procedure Call* incluem funções, procedimentos, metodos da orientação a objetos, chamadas de *callback* e *system calls*[8].

A camada de visão é representada pela interface gráfica e ela disponibiliza as telas para responder o questionário e gerar os gráficos e relatórios. As ações feitas na camada de visão são encaminhadas para a camada de controle, que é responsável por gerir os pedidos enviados e acionar as funcionalidades programadas para cada situação. A camada de modelo executa sas ações requisitadas e retornam o resultado para a camada de controle.

A camada de modelo tem como função notificar suas visões e controladores associados quando há uma mudança em seu estado. Esta notificação permite que as visões produzam saídas atualizadas e que os controladores alterem o conjunto de comandos disponíveis. Uma implementação passiva do MVC monta estas notificações, devido a aplicação não necessitar delas ou a plataforma de *software* não suportá-las.

A Figura 20[3] representa o padrão utilizado (MVC) no sistema com suas funções escritas em cada uma das camadas:

5.3 Montagem da placa

Nesta seção será mostrado os componentes que foram utilizados para montar a placa que ficou responsável por realizar as medições. Para realizar a montagem da placa foram necessário os seguintes componentes:

1. 1 Placa Arduino (neste projeto foi utilizada uma do modelo UNO).

2. 1 Placa de Cobre para o desenho da placa.

3. Percloreto de ferro, para corroer o cobre e terminar a produção da placa.

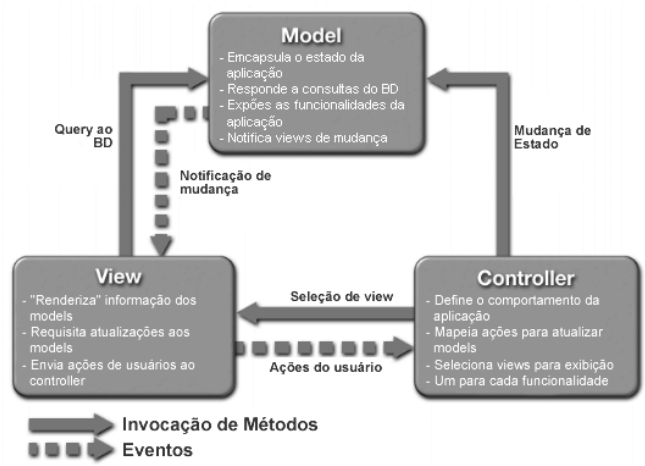


Figura 20: Modelo do padrão do sistema

4. 1 CT sensor YHDC SCT-013-000, que é o responsável pelas medições.
5. 1 Burden resistor 18 Ohms se a voltagem for 3.3V, ou 33 Ohms se a voltagem for 5V.
6. 2 10k Ohm resistors (ou qualquer resistor com valor par até 470k Ohm).
7. 1 10uF capacitor.
8. 1 Wi-Fly shield, necessário para fazer a comunicação com o servidor e enviar os dados para serem salvos.

Com estes componentes, o esquema que será mostrado a seguir pode ser seguido e podemos começar a extração dos dados de qualquer ponto de energia. A Figura 21 está o esquema de montagem que foi desenhado para a confecção da placa.

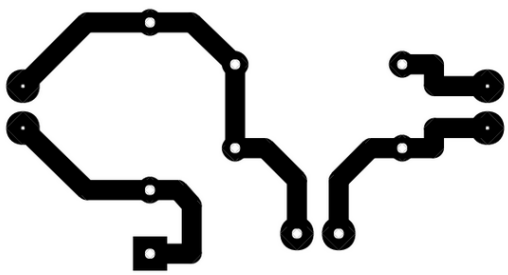


Figura 21: Esquema de montagem da placa

Na Figura 22 é mostrado o projeto em seu estado final já conectado ao arduino . A placa possui uma entrada para conectarmos o sensor responsável por realizar as medições (YHDC) em um dos lados e três fios no outro lado, que serão conectados no Arduino nas saídas de 5V, terra e sinal. Este sensor é conectado ao Arduino (com a placa *Wi-fly*, que pode ser vista na figura 23. O projeto não precisou utilizar mais a protoboard, pois foi desenvolvida manualmente e o tamanho ficou menor, fornecendo assim uma facilidade maior de se instalar nos locais.

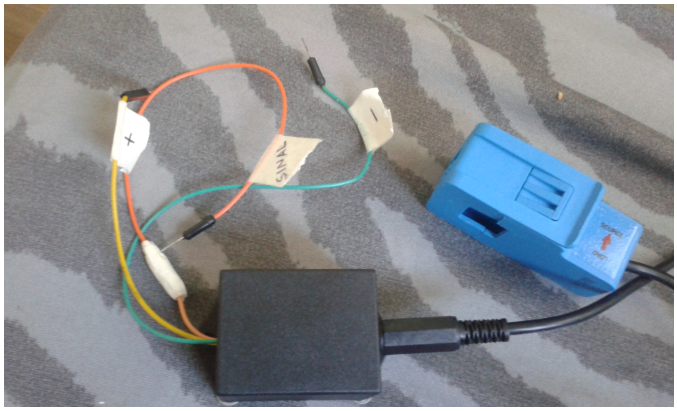


Figura 22: Projeto desenvolvido final

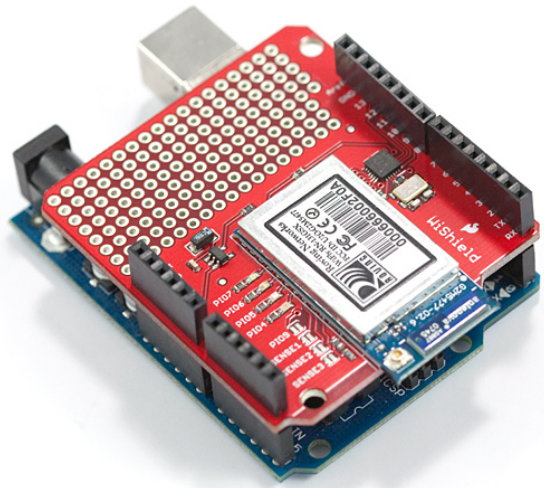


Figura 23: Wi-Fly shield (já conectado no arduino)

5.4 sensor YHDC SCT-013-000

O transformador de corrente Yhdc[12](Figura 24) é fabricado pela Beijing YaoHuadechang Electronic Co., Ltd e consignado por Seeed Estúdio de Shenzhen, China (www.seeedstudio.com) como sensor de corrente AC não-invasiva (100A max), modelo SCT-013-000, SKU THM105C4B.

Não tem qualquer resistência de carga interna, mas pode limitar a tensão de saída no caso de desligamento acidental. Ele é capaz de desenvolver a tensão suficiente para acionar totalmente uma entrada 5 V.

5.4.1 Testes com o sensor

Para correntes de teste de até 150 A, o principal CT consiste em de 1 a 30 fios de arame 16/0.2mm isolado, e a maioria dos testes foram realizados a 5 A - portanto, a corrente do primário visto pela CT pode ser ajustada em passos de 5

Figura 24: sensor YHDC SCT-013-00

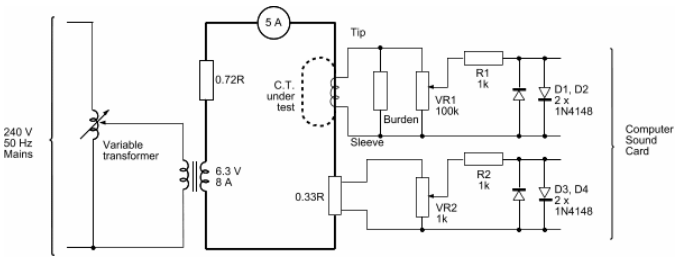


Figura 25: Esquema de testes

A, colocando um número variável de voltas no interior do núcleo. Para os ensaios de saturação superior a 250 A, o principal consiste em 50 passagens de fio de cobre esmaltado, a corrente sendo ajustada nesse caso.

(Nota: A corrente excede a classificação do fio utilizado para o principal, mas como a bobina é vagamente agrupados exceto cada ensaio é de duração relativamente curta, o aquecimento não é um problema).

A corrente primária foi monitorada a 0.33 amperes. Os potenciômetros, resistor limitador de corrente e diodos tanto na derivação e as saídas do CT são para proteger a placa de som do computador a partir de sobre-tensão e transientes de comutação, os potenciômetros foram ajustadas de forma que a tensão não ultrapassava 400 mV no pico e nesta tensão, os diodos não afetam o monitoramento.

A forma da onda de tensão e corrente primária do TC foram gravados usando um osciloscópio de software e a forma de onda registrada e posteriormente calibrado contra a tensão real, usando um multímetro ou um osciloscópio real, conectado diretamente em frente a saída do TC.

obs: A Figura 25 representa o esquema de testes.

5.4.2 O Transformador de Corrente Yhdc

Na Figura 26 temos os componentes internos e na Figura 27 temos o diagrama de circuitos do YHDC. A seta de corrente representa a corrente que flui para fora do transformador

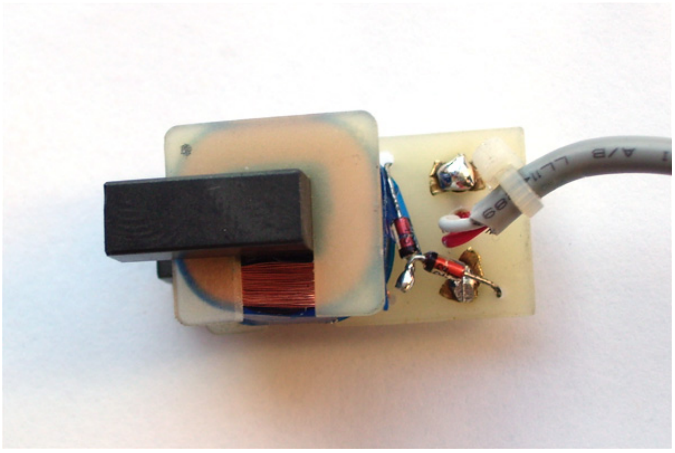


Figura 26: Componentes internos do YHDC

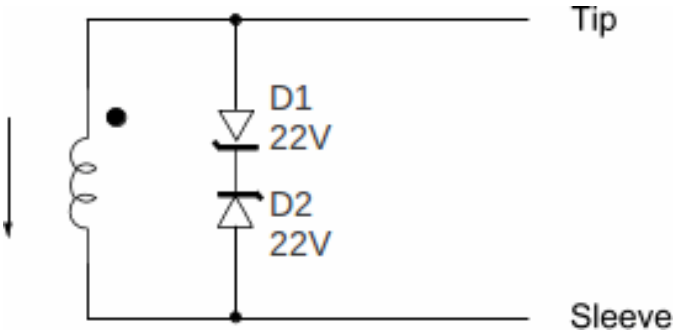


Figura 27: Diagrama de circuitos do YHDC

rotulado "SCT-013-000", em seguida, a flecha da ponta é positiva em relação ao *sleeve*. O fio branco é o principal e o vermelho é o *sleeve*.

O objetivo dos dois diodos zener 22 V é limitar a tensão que pode aparecer e através dos enrolamentos para um valor seguro.

5.5 Modelagem do sistema

Nesta seção será mostrada a modelagem do SMRE. Como por ser visto na Figura 28, o sistema possui 7 classes, sendo que 4 delas são relacionadas diretamente com o as medições e outras 3 relacionadas com a segurança do sistema. A tabela principal é a de MedicaoEnergia, onde são armazenados os dados de medição (valor, data e hora) e o ponto de medição utilizado.

O sistema possui 2 fluxos importantes, mostrados nos diagramas de sequência a seguir:

1. O fluxo mostrado na Figura 29 é o responsável pela inserção de dados no banco, onde os dados saem do Arduino, passam pela camada de negócios e finalmente vai à camada de persistência.

2. O fluxo mostrado na Figura 30 é o responsável por gerar os gráficos, onde o Cliente faz a solicitação do gráfico, a solicitação passa pela camada de apresenta-

ção, é buscada na camada de persistência e mostrada de volta em forma de gráfico.

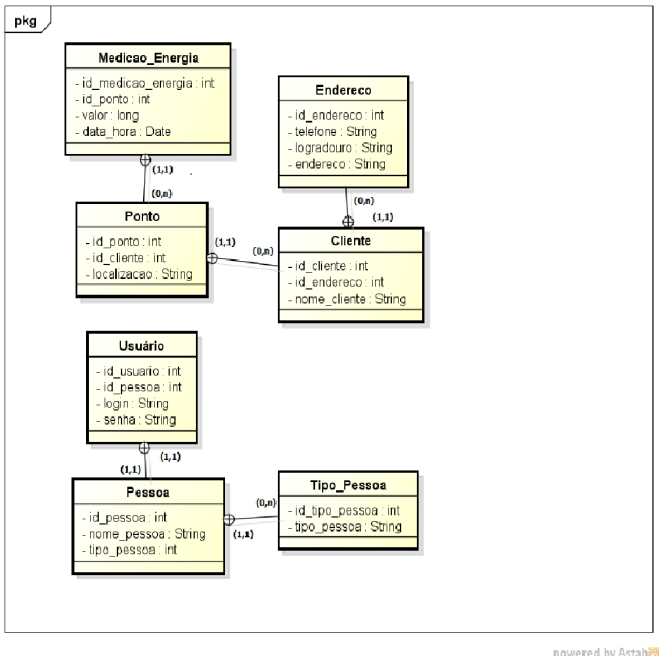


Figura 28: Diagrama de classes do sistema

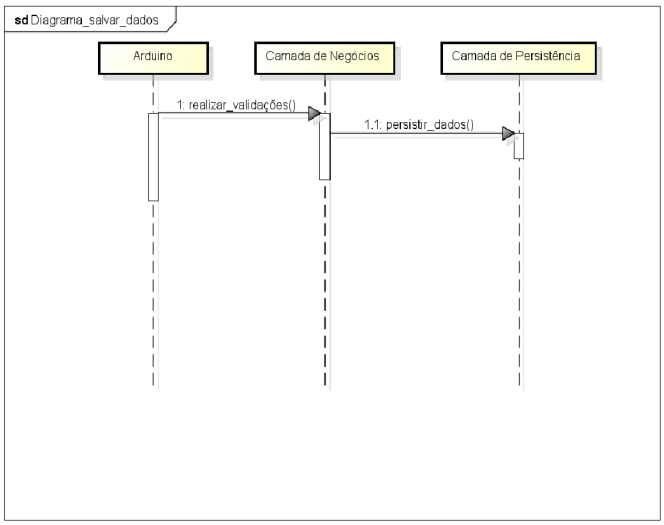


Figura 29: Diagrama de sequência (salvar dados)

6. PRINCIPAIS FUNCIONALIDADES

O SMRE deve ser acessado através de um navegador *web* moderno para a visualização de informações úteis em forma de gráficos. A tela de filtro dos gráficos é composta por:

1. Uma lista de Clientes que é filtrada de acordo com o perfil do usuário logado.

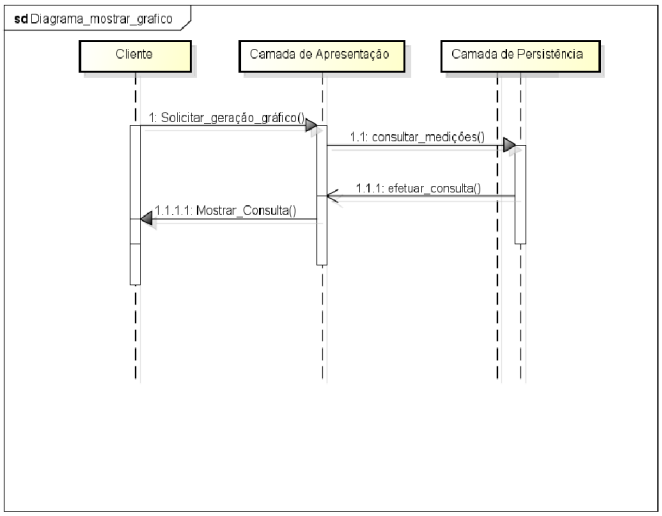


Figura 30: Diagrama de sequência (gerar gráficos)

2. Uma lista de Pontos de Medição que é filtrada de acordo com o cliente selecionado pelo usuário.
3. Dois campos de data que representam o período do filtro para gerar o gráfico.
4. Um Radiogroup com as opções de tipos de gráfico, que pode ser: Por ponto, por cliente, apenas as últimas 24 horas e média de consumo dos pontos por cliente. ps: Cada um dos tipos de gráficos selecionado irá gerar uma visão diferente para o usuário.
5. Um botão de Consultar Medições que é responsável por gerar o gráfico.

O fluxo que essa tela segue será descrito a seguir:

1. Após entrar na tela, o usuário deverá selecionar um dos clientes associados ao seu perfil.
2. De acordo com o tipo de gráfico, torna-se necessário informar um Ponto de Medição ou não.
3. Se a opção de tipo de gráfico selecionada foi: Consumo por ponto ou Últimas 24 horas, uma lista de Pontos de Medição é populada baseada no cliente selecionado e o usuário deverá selecionar qual ponto ele deseja obter as informações.
4. Se a opção de tipo de gráfico selecionada foi: Consumo por cliente ou Média de consumo por cliente, a lista de Pontos de Medição é bloqueada e o usuário só precisa informar o Cliente.
5. Se a opção de tipo de gráfico selecionada foi: Últimas 24 horas, automaticamente os campos de período são atualizados e bloqueados com as datas das últimas 24 horas.
6. Se a opção de tipo de gráfico selecionada não foi: Últimas 24 horas, o usuário deverá informar o período desejado para filtrar os dados do gráfico.

Figura 31: Gráfico de consumo por cliente

7. Após preencher todos os campos necessários o usuário deverá clicar no botão de Consultar medições para gerar o gráfico.

Como pode ser visto, de acordo com o tipo de gráfico selecionado, a tela e a funcionalidade pode mudar completamente. O sistema permite o usuário ter uma visão tanto geral como específica dos dados de medição de energia de uma forma muito clara. Todos os gráficos serão mostrados com as informações da energia corrente no momento da medição e seus pontos serão demarcados pela data e hora que foi feita tal medição. A seguir será mostrado o resultado dos gráficos gerados pelo cliente:

Na Figura 31 o cliente IFBA foi selecionado, selecionado o tipo de gráfico Consumo por cliente, um período informado, gerando assim um gráfico com todas as medições do cliente selecionado.

Na Figura 32 o cliente do IFBA foi selecionado, filtrando os seus pontos de medição, o ponto de medição Portaria foi selecionado, selecionado o tipo de gráfico Consumo por ponto e um período foi informado, gerando assim um gráfico com todas as medições do ponto selecionado.

Na Figura 33 o cliente do IFBA foi selecionado, selecionado o tipo de gráfico Média de consumo por cliente, um período foi informado, gerando assim um gráfico com a média de consumo dos pontos do cliente selecionado.

Os dados mostrados no gráfico de Últimas 24 horas são os mesmos mostrados nos gráficos de consumo por cliente e consumo por ponto, tendo como diferencial uma visão mais rápida e prática pelo gestor que só quiser verificar dados recentes (24 horas no caso).

7. PROCESSO DE TESTES

O processo de testes foi uma fase muito importante e necessitou da integração direta do usuário para auxiliar no teste. Nesta seçõ será apresentada a metodologia utilizada para testar o SMRE. Como o projeto foi dividido em duas partes (O desenvolvimento da aplicação *web* e a montagem do microcontrolador junto com a persistência dos dados), tivemos testes em ambientes e locais diferentes.

Figura 32: Gráfico de consumo por ponto de medição

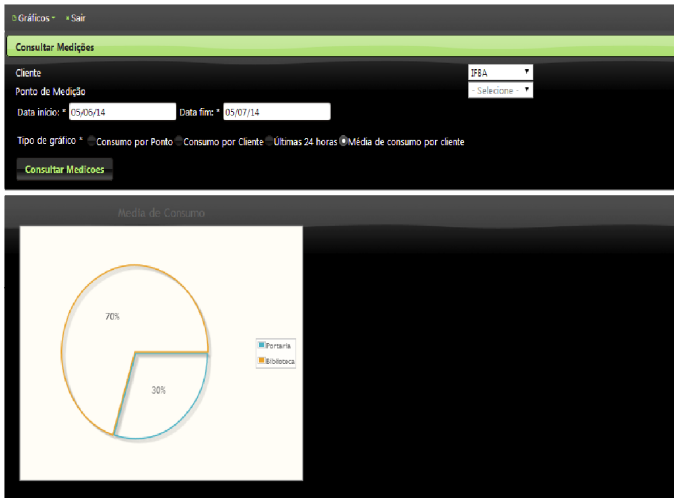


Figura 33: Gráfico de média de consumo por cliente

Testes unitários também conhecidos como *Unit test*, são os primeiros testes que a maioria dos sistemas passam, pois ele pode ser feito até durante o desenvolvimento da aplicação. O objetivo dos testes unitário é testar a menor funcionalidade existente do *software*, ou seja, isolar parte dos códigos e métodos, e analisar se essas funcionalidades tenham o retorno esperado mediando a um valor informado. Na aplicação *web* por exemplo, existe uma classe específica para fazer os testes dos módulos descritos.

Os testes automatizados servem para verificar se o *software* segue o fluxo correto que está especificado para o projeto. Como esse é um projeto grande, foi essencial realizar estes testes, pois ele é dividido em duas partes e a parte *web* depende totalmente do perfeito funcionamento do microcontrolador.

Para testar o Microcontrolador, foi necessário após a montagem do mesmo procurar por pontos em que pudesse fazer a medição de forma fácil e que a corrente elétrica passasse de forma constante. Para tal teste utilizei a energia de Ar condicionados em 3 locais diferentes, comparando a corrente que estava sendo mandada para a camada de persistência com a corrente que era passada para o ar condicionado no momento. O microcontrolador se conecta a um servidor, fica enviando os valores da corrente obtida dentro de uma faixa de tempo pre-parametrizada e após este processo os dados são salvos em banco. Na aplicação *web* então, podemos gerar um gráfico para testar a integridade dos dados obtidos com o mostrado em tela.

8. TESTES COM USUÁRIOS

Por se tratar de um sistema que trabalha com medições, os dados devem ser fiéis e consistentes. O usuário que utilizar o sistema poderá ter certeza de que seus pontos de medição estão passando a corrente exata e nenhum de seus aparelhos eletrônicos estão em risco.

Seis pessoas disponibilizaram os ar-condicionados de suas casas para os testes serem realizados. Os seis participantes destes testes foram convidados a responder um questionário composto por perguntas de Sim e Não, que diz respeito ao sistema e seu uso. O questionário é composto por três questões pessoais para o participante (nome, idade e sexo), oito questões sobre o uso do sistema em geral e uma para sugestões.

Antes de realizarmos as medições foram feitas as seguintes perguntas:

1. Você acredita que sua rede elétrica esteja cem por cento boa? (SIM/NÃO) - Observou-se que 3(50 por cento) participantes responderam que sim e 3(50 por cento) responderam que não.

2. Você já teve algum de seus eletrodomésticos queimado? (SIM/NÃO) - Observou-se que 4 (67 por cento) participantes responderam que sim e 2(33 por cento) responderam que não.

Após realizar as perguntas, o SMRE realizou medições por 7 dias e com os dados, foi gerada uma tabela que possui o nome do aparelho medido, a voltagem da rede elétrica e a média de corrente passada durante os 7 dias. A tabela pode ser vista na Figura 34:

Modelo do Ar Condicionado	Consumo Ideal	Média de consumo obtido
ELECTROLUX TE12F/TI12F	3.516 W/h	3.726 W/h
SAMSUNG AQ24UWBUXAZ	7.03 W/h	6.99 W/h
SAMSUNG AQ18UWBUXAZ	5.275.2 W/h	5.289 W/h
ELECTROLUX TE/TI09F	2.63 W/h	2.65 W/h
ELECTROLUX TI30R TE30R	8.792 W/h	9.111 W/h
LG TS-H092YNW0	2.63 W/h	2.88 W/h

Figura 34: Tabela de consumo

Após verificar a tabela de testes com os resultados obtidos, foram feitas as seguintes perguntas para termos um *feedback* dos usuários:

1. O sistema foi útil para você? (SIM/NÃO) - Observou-se que 6(100 por cento) participantes disseram que sim e 0 disseram que não.

2. Os dados foram mostrados de forma simples e clara? (SIM/NÃO) - Observou-se que 6(100 por cento) participantes disseram que sim e 0 disseram que não.

3. Após os testes você acredita que precisará fazer uma revisão na parte elétrica de sua casa? (SIM/NÃO)- Observou-se que 3(50 por cento) participantes responderam que sim e 3(50 por cento) responderam que não.

4. Você recomendaria este sistema para outras pessoas utilizarem? - Observou-se que 6 (100 por cento) participantes disseram que sim e 0 disseram que não.

5. O sistema atendeu suas expectativas? (SIM/NÃO) - Observou-se que 6 (100 por cento) participantes disseram que sim e 0 disseram que não.

6. Você utilizaria e confiaria no sistema para monitorar sua rede elétrica em outras ocasiões? (SIM/NÃO) - Observou-se que 6(100 por cento) participantes disseram que sim e 0 disseram que não.

Os testes foram todos feitos com valores reais obtidos de pontos diferentes para garantir a coerência da informação. Com o final dos testes, observou-se que os dados persistidos em banco (enviados pelo microcontrolador) foram salvos de forma correta e o cliente pôde ter várias visões diferentes dos dados de forma clara. Algo interessante é a visão de metade dos participantes que achavam que sua rede elétrica estava totalmente boa, enquanto na verdade não estava. Segundo os participantes, o sistema os ajudou muito e os influenciou a ficarem atentos sempre quanto à segurança em casa. Alguns dos participantes fizeram alguns comentários e sugestões boas para o projeto que pode ser visto na tabela da Figura 35:

Comentários
"Achei um sistema ótimo! Antes dos testes acreditava não ter problemas, após os testes mudei minha opinião e chamei um eletricista na mesma hora! Com certeza me salvou de um problema futuro."
"Muito bom e fácil de usar! Seria bom poder medir a água gasta diretamente nos canos também."
"Seria bom se pudéssemos cadastrar manualmente o tempo de intervalo entre as medições, ou teremos muitos ou poucos dados dependendo da necessidade."
"Nunca mais acharei que tudo está bem. É ótimo existirem métodos de monitoramento que não custem muito e não nos dê trabalho."
"Gostei de poder ter os dados que me interessam de forma fácil e rápida sem muita enrolação."

Figura 35: Comentários relevantes de usuários

O resultado final do questionário pode ser visto na tabela da Figura 36 além disso gráficos sumarizando as respostas das questões podem ser vistos no final deste projeto.

Questões	Respostas	
	Sim %	Não %
Questão 1	50	50
Questão 2	67	33
Questão 3	100	0
Questão 4	100	0
Questão 5	50	50
Questão 6	100	0
Questão 7	100	0
Questão 8	100	0

Figura 36: Sintetização questões sim ou não

9. CONCLUSÃO

A conscientização dos gastos elevados de energia está acontecendo de forma lenta. Projetos como este são de extrema importância para a sustentabilidade do planeta. Muitas vezes, o próprio esquema de montagem da rede elétrica está com problema e nós não sabemos, por isso durante os testes foram realizadas perguntas antes e depois para o usuário. Muitas pessoas acreditavam estar com a rede elétrica em perfeitas condições mas na verdade estavam tendo gastos a mais do que o necessário e o sistema ajudou.

Pode-se dizer que o desenvolvimento do SMRE pode ser caracterizado como uma relevante contribuição para a sustentabilidade do mundo. A sua relevância se dá pois apesar de existirem outros meios de controlar e monitorar a energia, nenhum deles possui um baixo custo aliado a uma facilidade de manuseio do sistema.

Apesar dos objetivos deste trabalho terem sido alcançados, com a análise dos trabalhos relacionados junto com as sugestões dos usuários, podemos ver que existem modificações e melhorias a serem realizadas. Sugestões para tais melhorias e trabalhos futuros podem ser vistos na próxima seção.

10. TRABALHOS FUTUROS

Por ser um projeto ousado e com um grande potencial para o futuro, existem algumas ideias que poderiam vir a melhorar o projeto no futuro como:

1. Adicionar outras grandezas para serem medidas no sistema.
2. Desenvolver uma versão do sistema *web* para *tablets* e *smartphones* das plataformas *android* e *iOS*.
3. Novos tipos de gráfico de acordo com a exigência do cliente.
4. Criação de um *Dashboard* na tela inicial do sistema para controle e informações gerais.
5. Adicionar uma funcionalidade de configuração manual do intervalo entre medições.
6. Implementar função de históricos comparativos entre períodos de medição.

11. REFERENCIAS

[1] G. A. H. F. P. L. V. Bonadio. aug 2009. `http://projeto39.wordpress.com/o-arduino/`.

[2] M. Brain. first edition, oct 2013. `http://tecnologia.hsw.uol.com.br/microcontroladores1.htm`.

[3] M. Brizeno, mar 2012. `http://brizeno.wordpress.com/2012/03/12/mao-na-massa-model-view-controller/`.

[4] C. E. Cirilo. Computação ubíqua: definição, princípios etecnologias. *Departamento de Computação – Universidade Federal de São Carlos*, pages 2+, 2012. `http://www.academia.edu/1733697/Computacao_Ubiqua_definicao_principios_e_tecnologias`.

[5] R. de Freitas Góis. Análise de requisitos. *ICMC - USP*, may 2012. `http://pt.slideshare.net/sukikitsune/anlise-de-requisitos`.

[6] F. Domingues. Computação ubíqua. *http://www.infowizard.com.br*, pages 1+, jun 2011. `http://www.hardware.com.br/artigos/computacao-ubiqua/`.

[7] N. e. W. E. Rozanski. *Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives.*, volume 1. Pearson Education, 2011.

[8] A. W. Kiwelekar. *Architectural Connectors*. Department of Computer Science and Engineering Indian Institute of Technology, Bombay Mumbai, 2007. `http://www.cse.iitb.ac.in/~awk/aconnector.pdf`.

[9] D. K. F. A. R. M. S. Lins. Projetos com arduino (em português). *UFPA*, may 2010.

[10] E. S. Ltda. *Saber Eletrônica*. 435. Revista Saber Eletrônica, apr 2009. `http://www.sabereletronica.com.br/artigos/2096-a-onda-do-arduino`.

[11] Radson. mar 2011. `http://tudodearduino.blogspot.com.br/2011/08/tipos-de-arduino.html`.

[12] C. M. Robert Wall B.Sc. Yhdc sct-013-000 current transformer. `http://openenergymonitor.org/emon/buildingblocks/report-yhdc-sct-013-000-current-transformer`.

[13] . F. S. O. J. P. M. R. E. S. L. S. C. R. P. Roda. Automação residencial utilizando o arduino. *Pontifícia Universidade Católica de São Paulo PUC-SP*, pages 1+, 2009. `https://www.slideboom.com/presentations/download/369037/Poster`.

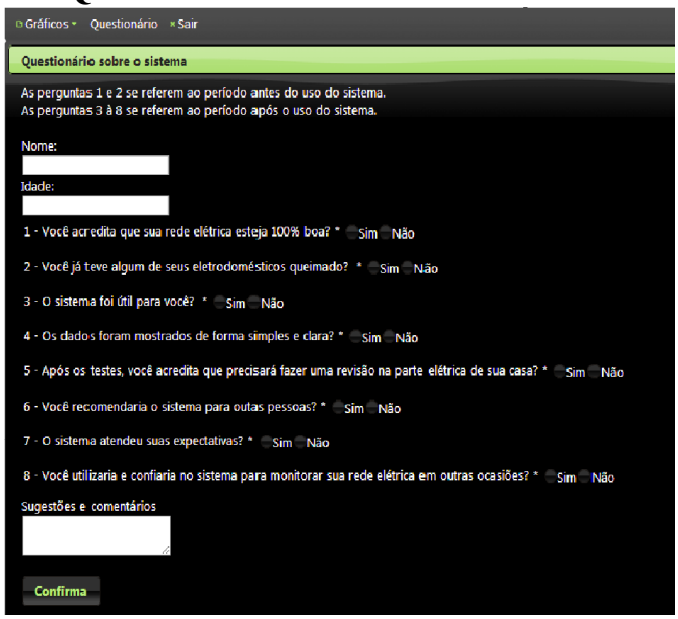
[14] M. A. Silva, 2014. `http://www.linhadecodigo.com.br/artigo/1685/a-importancia-do-levantamento-de-requisitos-no-sucesso-aspx`.

[15] E. Team, 2008+. `http://openenergymonitor.org/emon/`.

[16] Yaskawa, 2008. `http://www.yaskawa.com.br/wp-content/uploads/2013/07/Economia-de-Energia.pdf`.

Apendice

A. QUESTINÁRIO



Gráficos Questionário Sair

Questionário sobre o sistema

As perguntas 1 e 2 se referem ao período antes do uso do sistema.
As perguntas 3 à 8 se referem ao período após o uso do sistema.

Nome:

Idade:

1 - Você acredita que sua rede elétrica esteja 100% boa? * Sim Não

2 - Você já teve algum de seus eletrodomésticos queimado? * Sim Não

3 - O sistema foi útil para você? * Sim Não

4 - Os dados foram mostrados de forma simples e clara? * Sim Não

5 - Após os testes, você acredita que precisará fazer uma revisão na parte elétrica de sua casa? * Sim Não

6 - Você recomendaria o sistema para outas pessoas? * Sim Não

7 - O sistema atendeu suas expectativas? * Sim Não

8 - Você utilizaria e confiaria no sistema para monitorar sua rede elétrica em outras ocasiões? * Sim Não

Sugestões e comentários

Confirma

Figura 37: Questionário sobre o sistema

B. GRÁFICOS - TESTE COM USUÁRIOS

As Figuras exibidas nesta seção expõem graficamente os resultados obtidos pelo questionário aplicado durante a validação do SMRE com usuários.

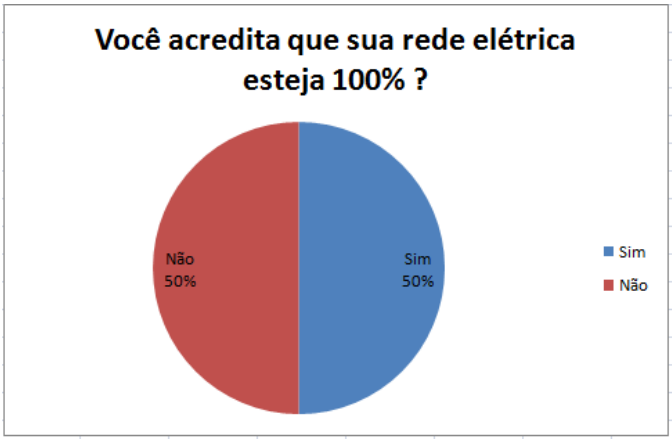


Figura 38: Questão 1

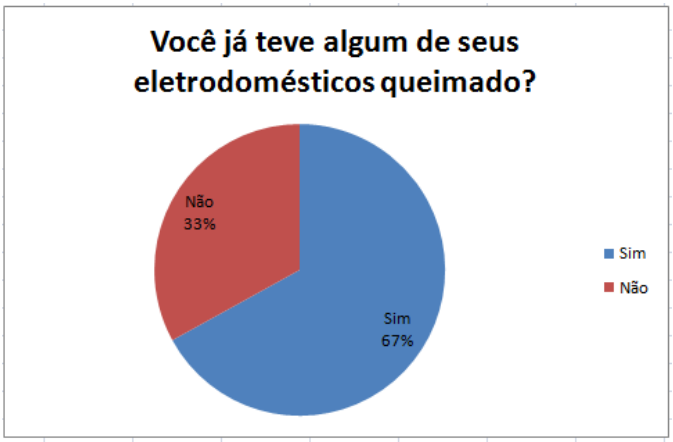


Figura 39: Questão 2

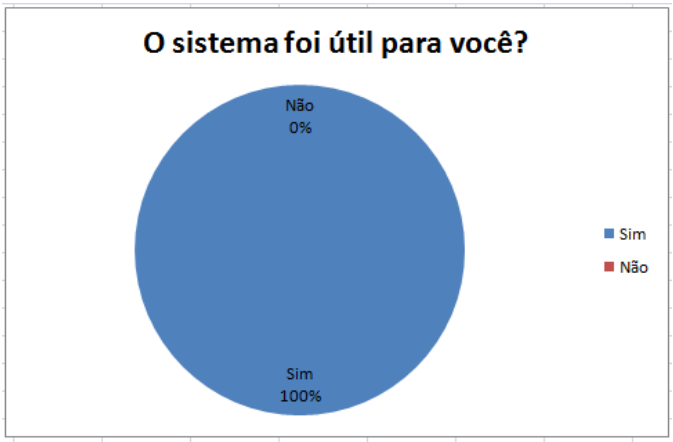


Figura 40: Questão 3

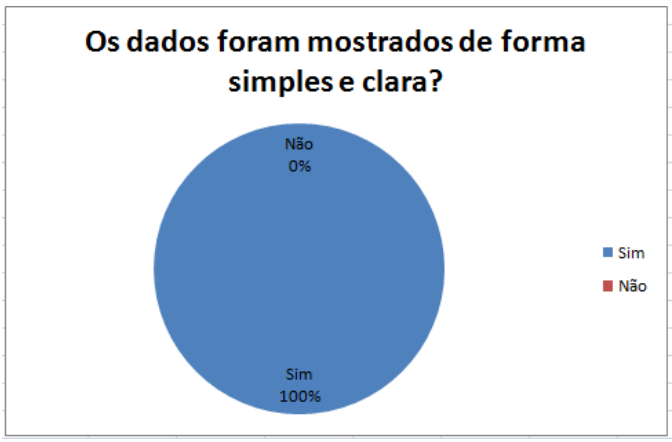


Figura 41: Questão 4

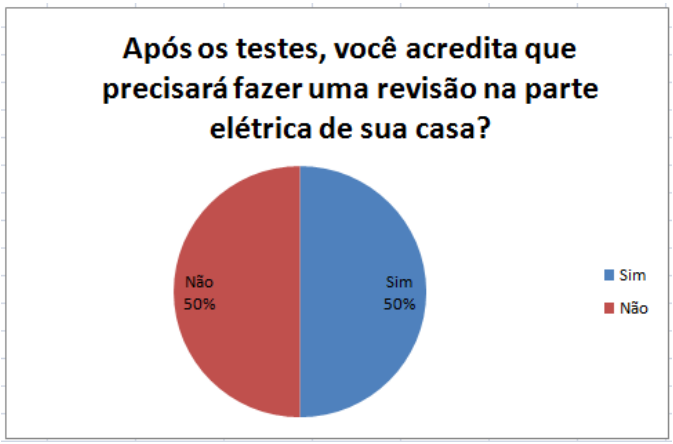


Figura 42: Questão 5

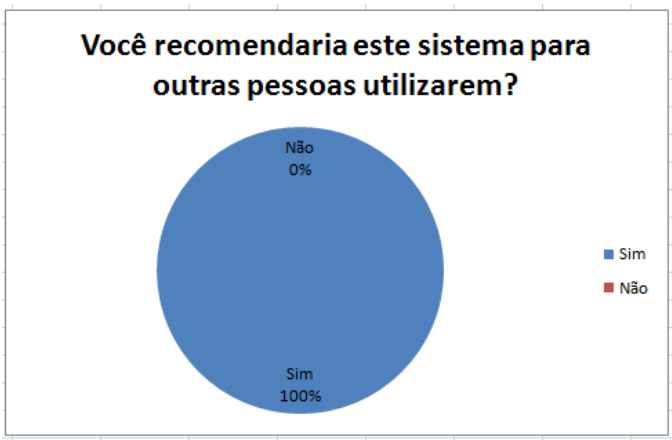


Figura 43: Questão 6

Figura 44: Questão 7

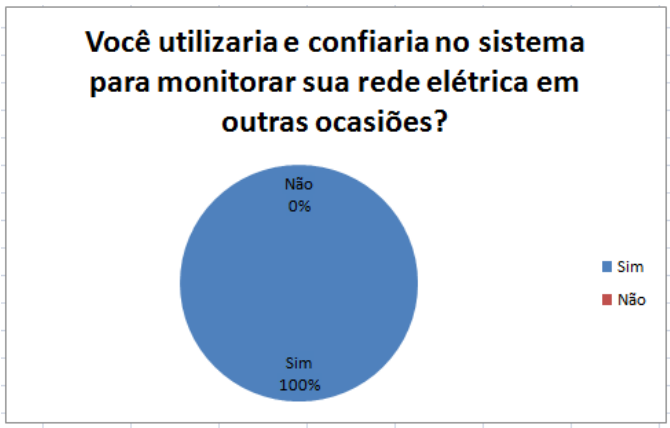


Figura 45: Questão 8