

Rodrigo Lisboa Della Rovere

**Protótipo de um Sistema Inteligente de
Monitoramento do Consumo de Energia
Elétrica de uma Residência**

São Carlos - SP

2016, 8 de novembro

Rodrigo Lisboa Della Rovere

Protótipo de um Sistema Inteligente de Monitoramento do Consumo de Energia Elétrica de uma Residência

Monografia apresentada ao Curso de Engenharia Elétrica, Ênfase Energia e Automação, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Universidade de São Paulo - USP

Escola de Engenharia de São Carlos - Departamento de Engenharia Elétrica

Orientadora: Prof.^a Dra. Vilma Alves de Oliveira

São Carlos - SP

2016, 8 de novembro

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Lisboa Della Rovere, Rodrigo
I873p Protótipo de um Sistema Inteligente de
Monitoramento do Consumo de Energia Elétrica de uma
Residência / Rodrigo Lisboa Della Rovere; orientadora
Vilma Alves de Oliveira. São Carlos, 2016.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Sistemas de Energia e Automação) -- Escola de
Engenharia de São Carlos da Universidade de São Paulo,
2016.

1. Internet das Coisas. 2. Monitoramento do consumo
de energia elétrica. 3. ESP8266. 4. Raspberry Pi. I.
Título.

FOLHA DE APROVAÇÃO

Nome: Rodrigo Lisboa Della Rovere

Título: "Protótipo de um sistema inteligente de monitoramento do consumo de energia elétrica de uma residência"

Trabalho de Conclusão de Curso defendido e aprovado
em 23 / 11 / 2016,

com NOTA 8,7 (OITO, SETE), pela Comissão Julgadora:

Profa. Titular Vilma Alves de Oliveira - Orientadora - SEL/EESC/USP

Mestre Heitor Vinicius Mercaldi - (Doutorando - SEL/EESC/USP)

Mestre Mauricio Eiji Nakai - (Doutorando - SEL/EESC/USP)

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado José Carlos de Melo Vieira Júnior

Dedico este trabalho à minha avó Maria (in memoriam) e ao meu afilhado Arthur.

Agradecimentos

À minha família, mãe Cicera, pai Cláudio e irmã Ravena, pelo amor e apoio em todos esses anos de faculdade.

À minha namorada Layna Amorim, pela incrível paciência, incentivo e total apoio nas horas difíceis, de desânimo e cansaço.

À minha orientadora Prof(a). Dra. Vilma Alves de Oliveira pela orientação e suporte, suas correções e incentivos.

Aos amigos que mais me ajudaram com o desenvolvimento desse projeto, Rafael Augusto Mariano e Pedro Arantes, sem eles, eu não teria conseguido.

Ao amigo Caio Martins, aos colegas do Laboratório de Controle e a todos amigos de São Carlos, pela amizade, troca de conhecimentos e por todos os momentos da minha vida acadêmica e pessoal.

Resumo

O projeto realizado é um protótipo de um sistema inteligente de monitoramento do consumo de energia elétrica de uma residência. O projeto é dividido em dois componentes principais: medidores pontuais e uma central de controle. O medidor é formado por um sensor (modelo AllegroTMACS712), que mede a corrente utilizada pelo aparelho conectado, e um microcontrolador ESP8266, com suporte à comunicação *WiFi*, para enviar os dados obtidos para a central de controle. A central é composta por outro ESP8266, que recebe os dados de todos os medidores, e um microcomputador Raspberry Pi, que processa esses dados e fornece ao usuário informações acerca do consumo de energia elétrica. Essas medidas são tomadas individualmente, podendo ser consultadas em quaisquer períodos de tempo escolhidos pelo usuário. As consultas ficam disponíveis na rede interna do usuário, podendo ser acessadas por meio de um *smartphone*, *tablet*, ou computador, via navegador *web*. Para a validação do sistema foram instalados módulos de aferimento em aparelhos domésticos e uma central na residência.

Palavras-chaves: Sistema inteligente, Medidor pontual, Monitoramento de energia elétrica, microcontrolador ESP8266, Raspberry Pi

Abstract

The undertaken project is a smart monitoring system prototype of a residence power consumption. The project was divided into two main components: a spot metering and a control unit. The spot metering consists of a sensor (model AllegroTM ACS712), which measures the current used by the connected device, and an ESP8266 microcontroller with support of WIFI communication, which sends the data to the control unit. The control unit is composed of another ESP8266, which receives data from all others meters, and a Raspberry Pi microcomputer which processes the data and provides to the user, information about electricity consumption. These measures are taken individually and may be consulted at any user-selected periods of time. The data queries are available in the internal user network and can be accessed through a smartphone, tablet, or computer via browser. To validate the system, metering modules were installed in home appliances and a control unit was installed in the residence.

Key-words: Smart Meter, Spot metering, Power monitoring, Microcontrolador ESP8266, Raspberry Pi.

Lista de ilustrações

Figura 1 – Conhecimento do consumo individual dos aparelhos.	19
Figura 2 – Interesse em saber o consumo individual dos aparelhos.	19
Figura 3 – Previsão de economia de energia elétrica.	19
Figura 4 – Material semiconductor - Fluxo de corrente linear. Fonte: Sensors (2002)	21
Figura 5 – Fluxo de corrente distorcido devido à presença de um campo magnético.	22
Figura 6 – Sensor de Corrente - Allegro™ ACS712.	23
Figura 7 – Logo WiFi.	23
Figura 8 – Módulo ESP8266-12E.	24
Figura 9 – NodeMCU ESP-12E.	24
Figura 10 – Raspberry Pi 1 - Modelo B	25
Figura 11 – Principais diagramas de blocos do sistema.	28
Figura 12 – Esquemático dos medidores.	29
Figura 13 – Instalação do pacote do NodeMCU na IDE Arduino - Passo 1	31
Figura 14 – Instalação do pacote do NodeMCU na IDE Arduino - Passo 2	32
Figura 15 – Instalação do pacote do NodeMCU na IDE Arduino - Passo 3	32
Figura 16 – Fluxogramas de toda programação.	36
Figura 17 – Fluxograma da análise final	37
Figura 18 – BLocos que compõem o circuito do Medidor	38
Figura 19 – Circuito do Medidor	38
Figura 20 – Circuito do Medidor com benjamim adaptado	39
Figura 21 – Visão interna do benjamim	39
Figura 22 – Conectores do equipamento, sensor e fonte de alimentação	40
Figura 23 – Protótipo final do Medidor	40
Figura 24 – Protótipo final da Central	41
Figura 25 – Interface com o Usuário - Página Final	41
Figura 26 – Medidas obtidas nos testes com os equipamentos	42
Figura 27 – Enquete: Pergunta 1	45
Figura 28 – Enquete: Pergunta 2	45
Figura 29 – Enquete: Pergunta 3	45
Figura 30 – Enquete: Pergunta 4	46
Figura 31 – Enquete: Pergunta 5	46
Figura 32 – Enquete: Pergunta 6	46

Sumário

1	INTRODUÇÃO	17
1.1	Internet das Coisas	17
1.2	Era da Informação	18
1.3	Consumidor Consciente	18
1.4	Medidores de Energia	20
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	Efeito <i>Hall</i>	21
2.1.1	Sensor de Corrente (Modelo Allegro™ACS712)	22
2.2	<i>Wireless</i> - <i>Wifi</i>	23
2.2.1	Módulo ESP8266	23
2.3	Microcomputadores - Raspberry Pi	25
3	ARQUITETURA E CONFIGURAÇÃO DO SISTEMA DE MONITORAMENTO DE CONSUMO DE ENERGIA	27
3.1	Sistema Inteligente de Monitoramento	27
3.1.1	Descrição	27
3.1.2	Central Inteligente	27
3.1.3	Medidores	28
4	TESTES REALIZADOS	31
5	RESULTADOS	37
5.1	Estimativa dos Custos do Projeto	42
6	CONSIDERAÇÕES FINAIS	43
	APÊNDICE A – PESQUISA REALIZADA	45
	APÊNDICE B – CÓDIGO <i>WIFI</i> SIMPLES	47
	APÊNDICE C – CÓDIGO <i>WIFI</i> SERVER	53
	APÊNDICE D – CÓDIGO <i>WIFI</i> CLIENT	57
	REFERÊNCIAS	61

1 Introdução

Nos dias de hoje, com a popularização dos meios de comunicação, a informação está cada vez mais ao alcance de todos. Isso se reflete em todos os lugares, porém a informação acerca do consumo de energia elétrica dos equipamentos eletro-eletrônicos é bem limitada. O consumidor tem uma visão geral dos gastos relacionados a seu consumo de energia, pois esse aferimento é realizado apenas no ponto de entrada de energia elétrica em sua residência, na maioria das vezes, por meio de medidores eletromecânicos, e essa falta de informação gera um consumo menos consciente, o que pode fazer grande diferença, tanto para o consumidor, quanto para o meio ambiente.

Com os atuais avanços no setor eletroeletrônico, é possível oferecer soluções automatizadas na área de medidores de energia. Conversores analógicos digitais, microcontroladores e sensores podem ser combinados para o desenvolvimento de um sistema que meça individualmente o consumo de cada aparelho, sendo facilmente instalado em um ponto de energia, como uma tomada, e fornecendo dados do consumo dos equipamentos conectados a esta.

Em um país como o Brasil, que possui mais de 200 milhões de habitantes, uma abordagem que permita o consumo consciente de energia elétrica nas residências é de grande interesse. Baseado nisso, o objetivo do presente trabalho foi o desenvolvimento de um medidor inteligente de energia elétrica, proporcionando mais clareza dos gastos diários e por consequência, auxiliando o usuário a agir de forma mais consciente, favorecendo a sustentabilidade e economia de energia elétrica.

1.1 Internet das Coisas

O que alguns anos atrás era apenas um sonho futurista, hoje em dia é algo amplamente discutido quando o assunto é tecnologia, senão o mais importante: a Internet das Coisas, do inglês *Internet of Thing*, IoT).

IoT é a “revolução silenciosa... cuja hora finalmente chegou”, foi a declaração dada por [Witchalls e Chambers \(2013\)](#), após realizar em conjunto com o grupo *The Economist Intelligence Unit*, EIU, uma pesquisa global com executivos e descobrir que 76% das empresas ligadas à tecnologia, estavam explorando ativamente ou utilizando a IoT. Ela é uma revolução tecnológica que representa a evolução da computação e tem como objetivo, estabelecer uma interação entre objetos inteligentes por meio da Internet. É, resumidamente, a possibilidade de comunicação entre todos os objetos que existem – enviando e recebendo dados e informações com o intuito de facilitar a vida das pessoas

([GE Look Ahead, 2014](#)).

A popularização dos sistemas embarcados microcontrolados possibilitou a automização de atividades cotidianas. A partir do uso das plataformas Arduino e Raspberry Pi, por exemplo, equipamentos residenciais podem ser aperfeiçoados com a integração da Internet e um monitoramento de dados mais precisos fornecidos por sensores acoplados.

1.2 Era da Informação

A Era da Informação é um termo frequentemente utilizado para designar o período advindo da Terceira Revolução Industrial inaugurado no final do século XX, e se refere à dinamização dos fluxos informacionais pelo mundo e a avanços tecnológicos, tais como: o microprocessador, a rede de computadores, a fibra óptica e o computador pessoal ([CASTELLS, 1999](#)).

Com o avanço dos *smartphones*, redes sociais, aplicativos de mensagens instantâneas, canais de vídeos, *sites* jornalísticos, *blogs* e tantos outros, a quantidade de informação disponível a todo momento e a qualquer pessoa, bastando ter acesso a *internet*, é imensurável. Isso vem estimulando as pessoas a se interessarem cada vez mais por assuntos que antes não recebiam devida importância, e hoje fazem parte do dia-a-dia do cidadão comum.

1.3 Consumidor Consciente

O consumidor consciente sabe que pode ser um agente transformador da sociedade por meio do seu ato de consumo. Sabe que os atos de consumo têm impacto e que, mesmo um único indivíduo, ao longo de sua vida, produzirá um impacto significativo na sociedade e no meio ambiente. Por meio de cada ato de consumo, o consumidor consciente busca o equilíbrio entre a sua satisfação pessoal e a sustentabilidade, maximizando as consequências positivas e minimizando as negativas de suas escolhas de consumo, não só para si mesmo, mas também para as relações sociais, a economia e a natureza ([Ministério do Meio Ambiente, 2014](#)).

Nesse âmbito, foi realizada uma pesquisa (Apêndice A), via redes sociais, com grupos de pessoas distintas, em sua maioria, jovens de 17 a 28 anos, durante os meses de junho a agosto de 2016, sobre o consumo consciente de energia elétrica residencial e a seguir tem-se alguns resultados dessa pesquisa. Os dados contidos nas Fig. 1 indicam que menos de 10% dos consumidores tem conhecimento dos gastos individuais de seus aparelhos eletro-eletrônicos, ou seja, qual o consumo do ar-condicionado, do chuveiro elétrico, da torradeira ou da televisão. Além disso, a Fig. 2 mostra que mais de 90% optaria por uma forma de aferimento que pudesse discriminar o consumo desses, e adicionalmente, quase

95% dos entrevistados disseram que agiriam de forma mais econômica, com relação ao consumo de energia elétrica, se esse fosse detalhado por equipamento (Fig. 3).

Você sabe quanto cada aparelho da sua casa consome de energia?

(116 respostas)

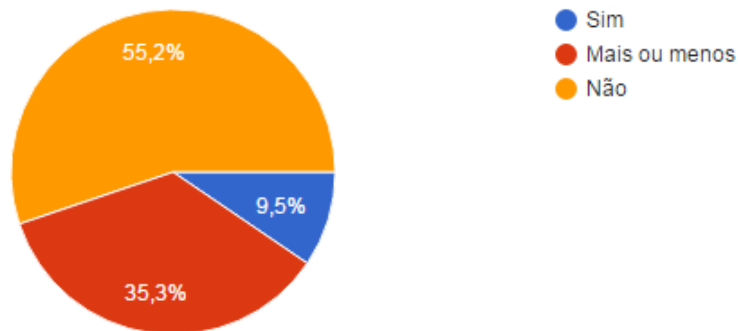


Figura 1 – Conhecimento do consumo individual dos aparelhos.

Você gostaria de saber quanto cada aparelho da sua casa gasta durante determinado uso?

(116 respostas)

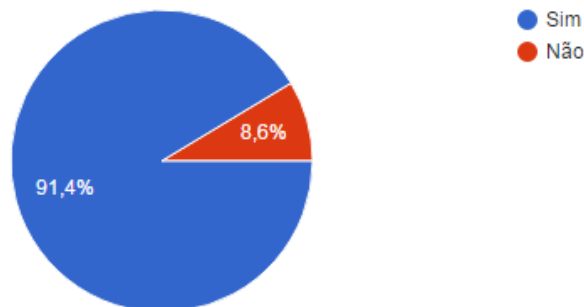


Figura 2 – Interesse em saber o consumo individual dos aparelhos.

Se você soubesse, você acha que conseguiria economizar energia?

(116 respostas)

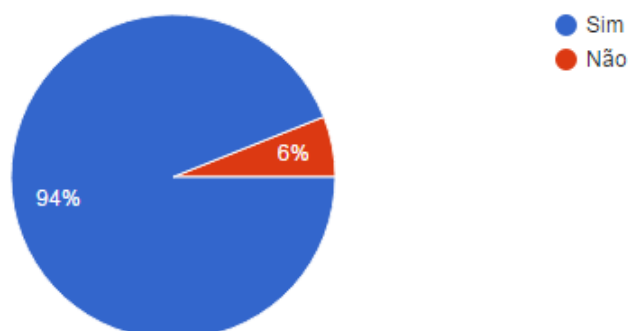


Figura 3 – Previsão de economia de energia elétrica.

Informações como essas, mostram que o consumidor se tornaria cada vez mais consciente sobre o uso da energia elétrica, caso soubesse exatamente seu consumo em determinado momento, buscando o equilíbrio entre a sua satisfação pessoal e a sustentabilidade.

Baseando-se nessas informações, este trabalho visou a construção de um protótipo para um sistema de medição inteligente do consumo de energia elétrica de uma maneira unitária, ou seja, será medido o consumo de cada aparelho individualmente. Essa medida pode parecer algo complexo para usuários leigos, porém, com auxílio dos métodos e avanços tecnológicos citados anteriormente, foi desenvolvido um aparelho de simples utilização.

1.4 Medidores de Energia

Os medidores eletrônicos vêm substituindo os medidores eletromecânicos há alguns anos. Contudo, estes ainda fornecem ao usuário, apenas um valor total do que foi consumido, sem distinção dos locais e aparelhos de consumo. Do ponto de vista das concessionárias de energia, essa medida é suficiente, pois é cobrado o total consumido pela residência, entretanto, visando a transparência dos valores que compõem esse total, medidores unitários e independentes têm sido disponibilizados aos usuários.

Já existem sistemas para a medição unitária do consumo de aparelhos eletroeletrônicos, porém, oferecem apenas medidas instantâneas. Esse mecanismo não fornece nenhum tipo de armazenamento desses dados, assim, nenhuma uma análise mais detalhada pode ser tomada acerca do consumo.

O objetivo deste protótipo foi realizar medidas em vários pontos de energia de maneira independente as quais são armazenadas em uma memória não-volátil (cartão SD), de forma que o usuário tenha acesso às informações do consumo de todos os aparelhos, podendo assim, realizar buscas e análises sobre o que for conveniente.

2 Fundamentação Teórica

Este capítulo tem por objetivo introduzir os conceitos teóricos que foram utilizados em todo o trabalho: Efeito *Hall*, tecnologia de comunicação *Wi-Fi* e microcomputadores.

2.1 Efeito *Hall*

O efeito *Hall* se caracteriza basicamente pelo aparecimento de um campo elétrico quando se mergulha um condutor, percorrido por uma corrente elétrica, em um campo magnético ([SENSORS, 2002](#)).

Edwin Hall realizava estudos e experiências buscando entender qual a influência de um campo magnético externo sob um fio condutor. Ele queria entender se a força devido a este campo externo atuaria sobre os portadores de corrente elétrica ou sobre o fio como um todo. *Hall* acreditava que essa força magnética atuaria sobre os portadores de carga fazendo com que a corrente se deslocasse para uma determinada região do fio, e portanto, a resistência do fio iria aumentar. Apesar de não observar tal aumento na resistência do fio em seus experimentos, *Hall* sabia que de alguma forma a corrente elétrica era alterada sem que a resistência fosse modificada. Ele propôs a presença de um estado de *stress* em uma determinada região do condutor, devido ao acúmulo de portadores de carga, que originaria uma diferença de potencial transversal, mais tarde conhecida como tensão de *Hall* ([WIKIPÉDIA, 2016](#)).

Na Fig. 4, tem-se um filme de material semiconductor percorrido por uma corrente elétrica ao longo do eixo x . A corrente é formada por um fluxo de portadores de carga (elétrons, íons ou lacunas) que seguem uma trajetória, representado pelas setas. Sem a presença de um campo magnético, a trajetória desse fluxo é linear e a distribuição de corrente sobre o semiconductor é uniforme. A diferença de potencial resultante é zero ([SENSORS, 2002](#)).

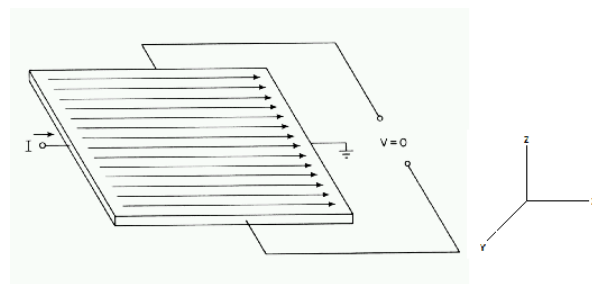


Figura 4 – Material semiconductor - Fluxo de corrente linear. Fonte: [Sensors \(2002\)](#)

Na presença de um campo magnético externo (Fig. 5), representado por B , aplicado

ao longo do eixo z , perpendicular ao semicondutor, os portadores de carga sofrem uma força magnética que causa uma deflexão em suas trajetórias na direção y , ou seja, o fluxo de corrente é distorcido. Essa mudança de trajetória gera um gradiente de cargas e consequentemente surge um campo elétrico na direção y , conhecido como campo de *Hall*. Devido às dimensões finitas do fio, haverá um acúmulo de cargas nas extremidades ao longo da direção y , resultando em uma diferença de potencial conhecida como Tensão *Hall*, V_{hall} (SENSORS, 2002).

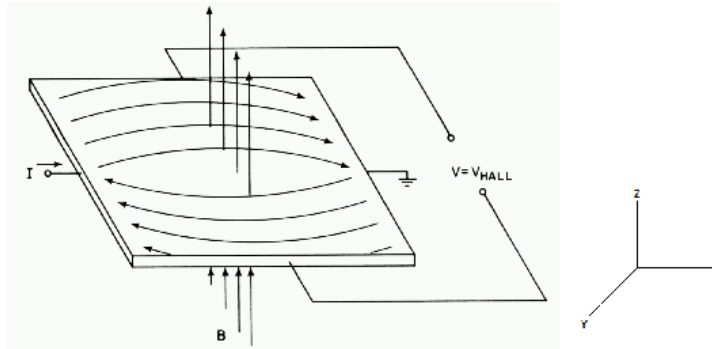


Figura 5 – Fluxo de corrente distorcido devido à presença de um campo magnético.

Fonte da imagem: SENSORS (2002)

Uma equação que descreve superficialmente a interação entre campo magnético, corrente e tensão *Hall*, dada em *Volts*, é:

$$V_{hall} = k * I * B * \sin(\theta) \quad (2.1)$$

em que:

- k : Constante definida pela geometria do elemento *Hall* e pela temperatura ambiente.
- I : Corrente que flui através do elemento, dada em amperes (A).
- $B * \sin(\theta)$: Componente do campo magnético perpendicular ao filme. dado em Tesla (T).

2.1.1 Sensor de Corrente (Modelo Allegro™ACS712)

Foi utilizado o sensor de corrente ACS712 (Fig. 6) do fabricante Allegro™. Esse sensor se destaca por ser uma solução precisa e econômica para detecção de corrente em sistemas industriais e residenciais, tanto para corrente contínua como corrente alternada. É disponibilizado em 3 versões que se diferem pela capacidade máxima de corrente que pode ser medida: 5A, 20A ou 30A. Foi escolhida a versão de 30A, pois muitos aparelhos domésticos ultrapassam os 5A e 20A máximos das versões inferiores (LLC, 2012).

Esse modelo de sensor de corrente foi escolhido por ser isolado galvanicamente do sistema, uma vez que o circuito *Hall* não entra em contato direto com o circuito da corrente

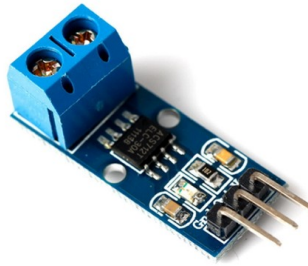


Figura 6 – Sensor de Corrente - AllegroTMACS712.

medida, e por possuir um tamanho muito menor do que transformadores, possibilitando assim, um medidor bastante compacto e de fácil manuseio.

2.2 Wireless - Wifi

Wireless, ou, rede sem fio, é uma tecnologia que permite a transmissão de dados e informações sem a necessidade do uso de cabos. Essa comunicação normalmente é feita através de equipamentos de radiofrequência ou de infravermelho. Essa infraestrutura tecnológica inclui desde o uso de transceptores de rádio (como *walkie-talkies*) até satélites artificiais no espaço sideral. Seu uso mais comum é em redes de computadores, servindo como meio de acesso à internet através de locais remotos, ou na comunicação entre dispositivos. Esse tipo de rede sem fio é chamado de *Wifi* (SILVA, 2008).

Wifi é uma abreviação de “*Wireless Fidelity*” que significa fidelidade sem fio, em português. É um tipo de rede sem fio que se comunica por meio de ondas de rádio e que não necessita de licença para instalação e/ou operação. Sendo necessário apenas que o aparelho receptor seja homologado pela ANATEL e esteja ao alcance das ondas de rádios transmitidas pelo aparelho emissor do sinal.



Figura 7 – Logo WiFi.

2.2.1 Módulo ESP8266

O módulo WiFi ESP8266 é um "sistema em um chip", do inglês *System On Chip* ou SOC, com o protocolo TCP/IP integrado que consegue dar a qualquer microcontrolador acesso a sua rede *WiFi*. O ESP8266 é capaz tanto de hospedar uma aplicação quanto descarregar todas as funções de redes *WiFi* a partir de outro processador de aplicação.

2.3 Microcomputadores - Raspberry Pi

Os microcomputadores já existem há anos e são conhecidos por seu uso doméstico e em aplicações nas mais diversas áreas, tanto profissional, quanto para jogos e entretenimento. Ainda assim, um dos usos mais importantes de um microcomputador é na programação, pois, de um modo geral, é que mais gera benefício para desenvolvimento da humanidade.

Percebendo que o nível de conhecimento dos novos alunos, em programação, estava em constante declínio, um grupo de estudantes do laboratório de computação da Universidade de Cambridge, nos Estados Unidos, criou o projeto Raspberry pi, que tem como objetivo disponibilizar um microcomputador simples e de baixíssimo custo para que jovens e crianças de todos os países possam ter acesso às ferramentas básicas para o aprendizado de programação ([Raspberry Pi Foundation, 2016](#)).

Segundo [Machado \(2012\)](#), os primeiros protótipos nasceram ainda em 2006 e têm sido aprimorados desde então. Hoje em dia, a versão mais recente resume-se a apenas uma pequena placa contendo todos os elementos centrais de um PC, com o tamanho próximo ao de um cartão de crédito. No projeto foi usado o Raspberry Pi 1 - Modelo B (Fig. 10), que além de contar com o processador ARM1176JZF-S de 700 Mhz e 512MB total de RAM, possui as seguintes especificações:

- GPU VideoCore IV de 250 MHz;
- Saída de Vídeo HDMI e RCA;
- Saída de áudio via 3.5mm Jack;
- Interface de rede *ethernet*, 10/100mb (RJ45);
- 2 portas USB 2.0;
- Conector Micro USB para alimentação (5 volts, 700mA).



Figura 10 – Raspberry Pi 1 - Modelo B
Fonte da imagem: [Raspberry Pi Foundation \(2016\)](#)

3 Arquitetura e Configuração do Sistema de Monitoramento de Consumo de Energia

Neste capítulo tem-se a descrição detalhada de todos os componentes do projeto, suas funções, modos de operação e os métodos usados para a configuração e programação dos mesmos.

3.1 Sistema Inteligente de Monitoramento

3.1.1 Descrição

O projeto constituiu-se na construção de um protótipo para a medição do consumo unitário de aparelhos domésticos, e foi dividido em duas partes principais: Medidores e uma Central de controle.

Os medidores são aparelhos similares a um benjamim comum, que é conectado entre a tomada e o aparelho que deseja-se medir o consumo. O usuário pode decidir quantos equipamentos ele deseja monitorar, sendo instalado um medidor por aparelho. Cada medidor envia a potência consumida em intervalos de 1 segundo, via *Wifi*, para a central.

A central é um aparelho único, responsável por receber, armazenar, enviar para a rede interna do usuário, e disponibilizar todas as informações referentes às medidas de cada medidor. O armazenamento é contínuo, mesmo em casos de queda de conexão com a internet, garantindo que todo o histórico do consumo seja contabilizado. Os diagramas de blocos das Figs. 11a e 11b ilustram os pontos principais do sistema, que serão mais detalhados adiante.

3.1.2 Central Inteligente

A central inteligente é composta por um NodeMCU ESP-12E (Node) e um Raspberry pi (RPI) (Fig. 11). O Node é responsável por receber as informações dos ESP's dos medidores, que enviam o valor da potência consumida do aparelho conectado. Esse valor é passado para o RPI através de uma comunicação serial, Tx/Rx, e é armazenado junto com o horário do envio e a identificação do medidor. Dessa maneira, todas as informações permanecem armazenadas em um banco de dados no RPI e transmitidas para uma página HTML da rede interna, onde o cliente tem acesso através de um *smartphone*, *tablet* ou computador. Nessa página de acesso do usuário, pesquisas sobre o consumo individual de cada aparelho são fornecidas com resultados mostrados em tabelas.

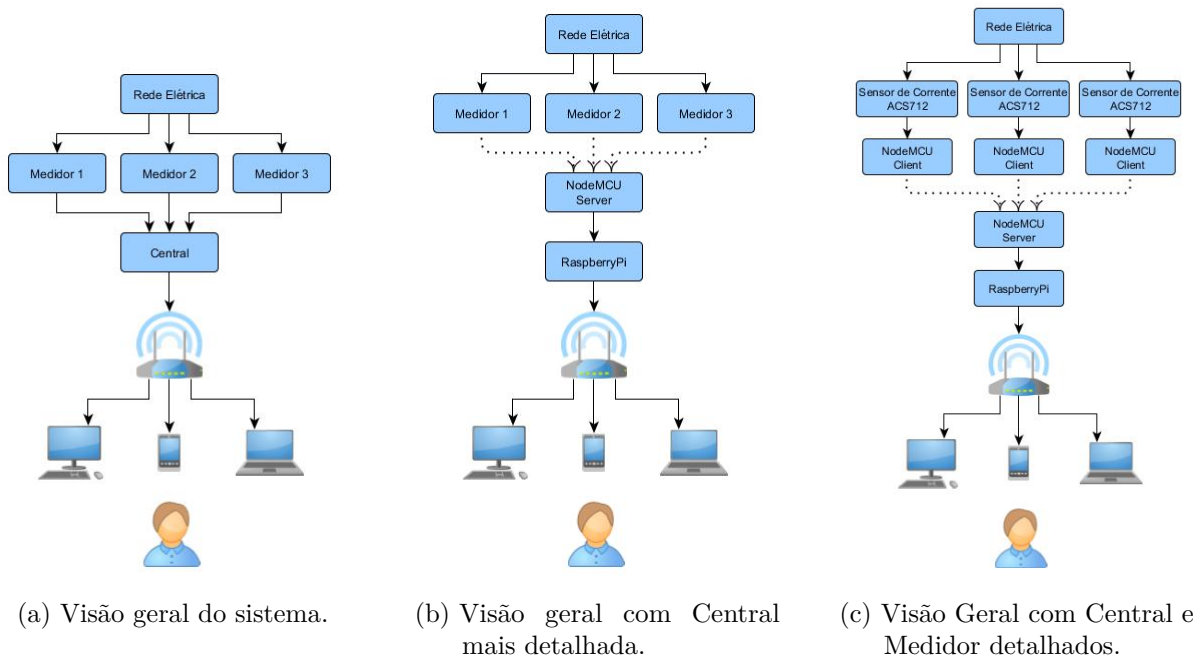


Figura 11 – Principais diagramas de blocos do sistema.

3.1.3 Medidores

Os medidores são responsáveis por conter todo o sistema de aferimento da potência dos aparelhos monitorados, são conectados diretamente na tomada, fazendo uma ponte até o aparelho que deseja-se monitorar. Esse dispositivo possui dois componentes principais: um NodeMcu e um sensor de corrente ACS712. O diagrama da Fig. 11c mostra a disposição desses blocos no sistema, bem como é um detalhamento final dos principais blocos que o compõem.

O medidor de potência é alimentado a partir da rede em que está conectado, possuindo em seu circuito eletrônico uma fonte reguladora de saída 5V conectada em paralelo com a rede, alimentando o sensor de corrente ACS712 e o NodeMCU. O conector, do tipo borne, do ACS712 é colocado em série com rede, da mesma maneira que mede-se corrente com um amperímetro comum, de forma que a corrente total consumida pelo aparelho conectado seja aferida. Na Fig. 12 pode-se verificar os componentes internos dos medidores, bem como suas conexões.

O ACS712 usa o Efeito *Hall* para a obtenção do valor da corrente. A corrente que passa pelos conectores gera um campo magnético no circuito *Hall*, que por sua vez, desloca a tensão de saída proporcionalmente a esse campo. A tensão de saída do sensor varia de acordo com sua tensão de alimentação, sendo 0 V para 30 A negativo, menor valor de corrente medido, e 5 V (V_{cc}) caso a corrente seja máxima, 30 A. Ou seja, caso não haja corrente fluindo até o equipamento, a tensão de saída do sensor será 2,5V ($V_{cc}/2$), obedecendo uma relação linear entre a corrente medida (I) e a tensão de resposta do

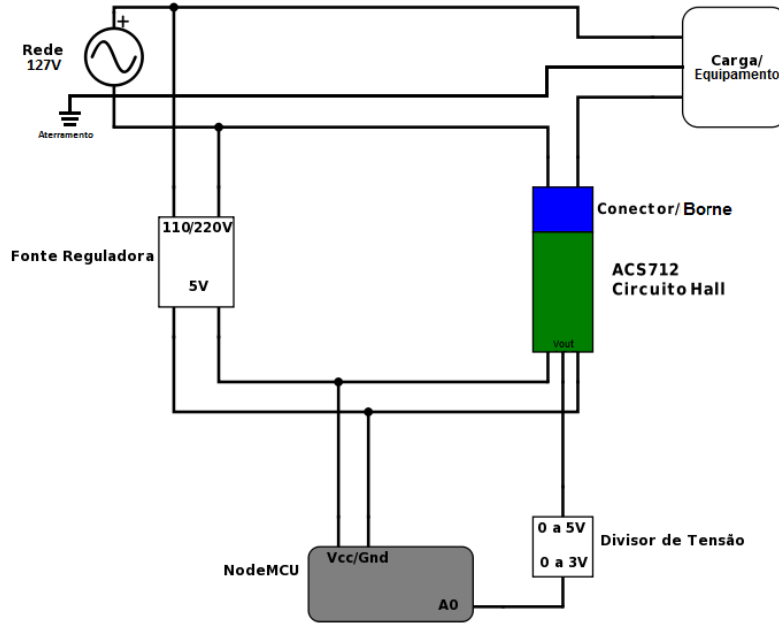


Figura 12 – Esquemático dos medidores.

sensor (V_{out}), que pode ser descrita pela por:

$$V_{out} = m * I + \frac{V_{cc}}{2} \quad (3.1)$$

em m é a sensibilidade do sensor, dada em V/A. Rearranjando (3.1) tem-se que:

$$I = \frac{V_{out} - \frac{V_{cc}}{2}}{m} \quad (3.2)$$

Esse cálculo da corrente I é realizado pelo NodeMCU, que recebe o valor da tensão de saída do sensor de corrente através de uma porta analógica (A0). Entretanto, a entrada do Node suporta no máximo 3,3 V, por isso, um circuito divisor de tensão foi adicionado, a fim de diminuir essa tensão de uma maneira proporcional, sendo assim, o V_{cc} usado como referência para os cálculos, será de 3,3 V.

O Node possui um conversor analógico digital (ADC) de 10 bits e 3,3V máximo, ou seja, um *range* de 3,3 V é dividido em 2^{10} valores digitais, representados de 0 a 1023. O valor de tensão gerado pelo sensor de corrente representa o valor de pico, assim o valor da tensão de saída eficaz é calculada por:

$$V_{ef} = \frac{V_{out}}{\sqrt{2}}. \quad (3.3)$$

O trecho de código a seguir demonstra a conversão do valor analógico de tensão de saída do sensor de corrente, em um valor aproximado de corrente.

```
1 float Sensibilidade = 0.0378745478;
```

```
2 float Vcc=3.3; //Tensao de alimentacao do sensor usada para os
   calculos
3 float voltageSensor = analogRead(A0)*(3.3 / 1023); //leitura do
   sensor, Vout, V de pico
4 voltageSensor =voltageSensor * 0.707; //valor eficaz
5 float I=(voltageSensor - (Vcc/2))/Sensibilidade; //Equacao para
   obter a corrente
6 Serial.println("Corrente: ");
7 Serial.println(I,3);
```

Sensibilidade

A sensibilidade do sensor é dada pelo fabricante, porém, foi feita uma calibragem a fim de melhorar esse valor para as condições que estavam sendo utilizadas. Primeiramente, foi obtido o valor de tensão gerado pelo programa quando se tinha um valor de corrente nulo, ou seja, nada estava conectado ao sensor. Em uma outra situação, foi aplicado um valor de corrente conhecido, através de uma fonte CC do laboratório, e anotado o valor de tensão gerado novamente. Como tem-se uma relação linear entre a corrente e a tensão, foi calculado o coeficiente angular da reta formada pelos dois pontos obtidos, obtendo assim um valor ajustado para a sensibilidade do sensor:

$$m = \frac{V_2 - V_1}{I_2 - I_1}. \quad (3.4)$$

Os limites da calibração foram os limites dos equipamentos disponíveis em laboratório, sendo testado com uma corrente máxima de 10A, valor que não garante totalmente a linearidade até a corrente máxima de 30A, mas que fornece uma boa aproximação.

4 Testes Realizados

Para a programação do NodeMCU, foi utilizado a IDE (*Integrated Development Environment*) do Arduino, versão 1.6.12, bem como sua linguagem padrão, C. Porém, para a gravação do chip ESP8266, foi necessário instalar um pacote adicional de placas que o próprio fabricante fornece. Para essa inclusão, foram seguidos os passos do [Thomsen, Adilson - Blog Filipe Flop \(2016\)](#), descritos abaixo.

1. *Entre na IDE do Arduino e clique em Arquivo -> Preferências:*

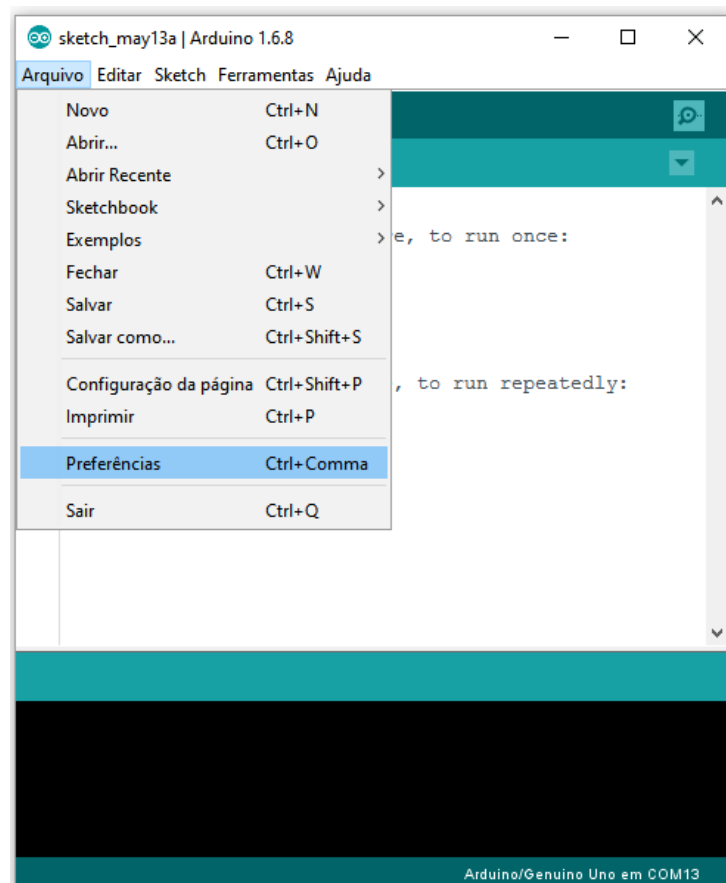


Figura 13 – Instalação do pacote do NodeMCU na IDE Arduino - Passo 1

2. *Na tela seguinte, digite o link abaixo no campo URLs adicionais de Gerenciadores de Placas:*
`http://arduino.esp8266.com/stable/package_esp8266com_index.json`
3. *A sua tela ficará como mostrado na Fig. 14:*
4. *Clique em OK para retornar à tela principal da IDE.*

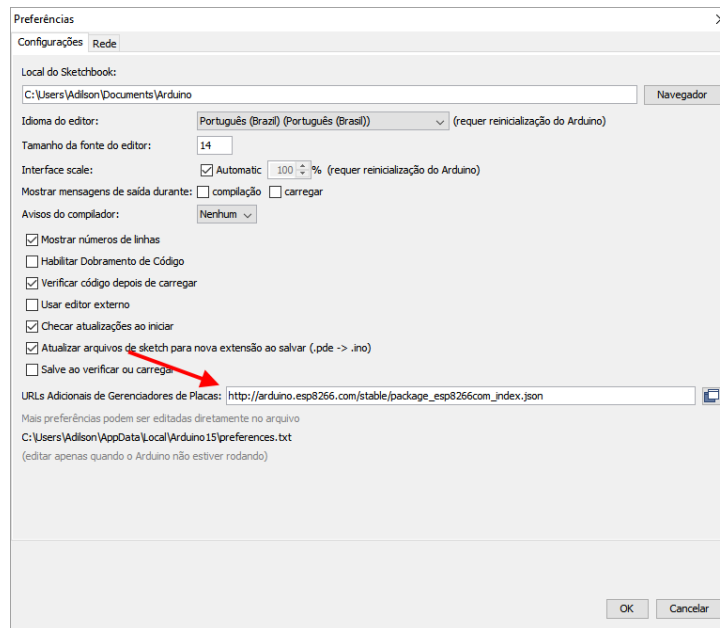


Figura 14 – Instalação do pacote do NodeMCU na IDE Arduino - Passo 2

5. Reinicie a IDE do Arduino.

6. Agora clique em Ferramentas -> Placa -> Gerenciador de Placas - Como mostra a Fig.15.

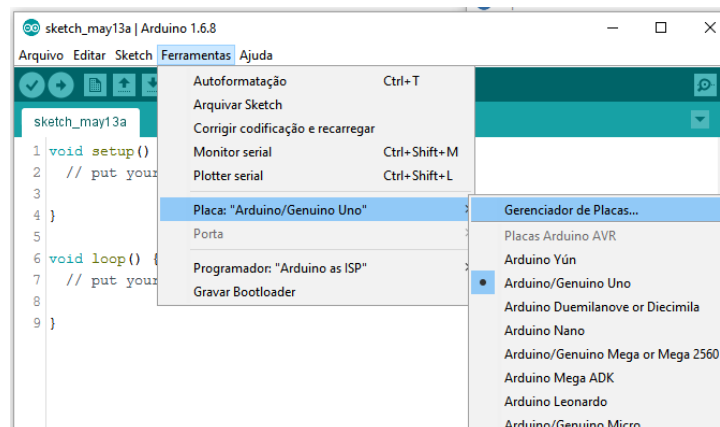


Figura 15 – Instalação do pacote do NodeMCU na IDE Arduino - Passo 3

7. Utilize a barra de rolagem para encontrar o ESP8266 by ESP8266 Community e clique em INSTALAR.

Após alguns minutos as placas da linha ESP8266 já estão disponíveis na lista de placas da IDE do Arduino.

• Teste do *Wifi* no NodeMCU - Rede interna

Após a instalação do pacote de placas do ESP8266, a IDE do Arduino estava pronta para receber os códigos do NodeMCU. O primeiro teste realizado foi para verificar

e estudar o funcionamento da comunicação *Wifi* do Node. Usando os principais comandos do pacote fornecido, foi desenvolvido um programa base de comunicação via *Wifi* para uso em todo o projeto. Tem-se abaixo um trecho desse código.

```
1 #include <ESP8266WiFi.h>
2
3 const char* ssid = "wireless-lac"; //nome da rede
4 const char* password = "12345678"; //password
5 ;
6 WiFiServer server(80);
7
8 void setup() {
9     Serial.begin(115200);
10    delay(10);
11
12    // Connect to WiFi network
13    Serial.println();
14    Serial.println();
15    Serial.print("Connecting to ");
16    Serial.println(ssid);
17
18    WiFi.begin(ssid, password);
19
20    while (WiFi.status() != WL_CONNECTED) {
21        delay(500);
22        Serial.print(".");
23    }
24
25    Serial.println("");
26    Serial.println("WiFi connected");
27
28    // Start the server
29    server.begin();
30    Serial.println("Server started");
31
32    // Print the IP addresscc
33    Serial.print("Use this URL to connect: ");
34    Serial.print("http://");
35    Serial.println(WiFi.localIP());
36    Serial.println(WiFi.macAddress());
37    Serial.println("/");
38    delay(1);
```

```
39 }
40
41 void loop() {
42     // Check if a client has connected
43     WiFiClient client = server.available();
44     if (!client) {
45         return;
46     }
47
48     // Wait until the client sends some data
49     Serial.println("new client");
50     while(!client.available()){
51         delay(1);
52     }
53
54     // Read the first line of the request
55     String request = client.readStringUntil('\r');
56     Serial.println(request);
57     client.flush();
58
59     //use "request" para tratar os dados recebidos
60 }
```

No Apêndice B está o código completo usado para comunicação *Wifi* com a rede interna e criação de uma interface que disponibiliza a opção de acender e apagar *leds* conectados ao Node. Essa interface é acessada através de um IP interno da rede, fornecido logo após a comunicação, e verificado através do *Serial Monitor* do Arduino IDE.

• Teste de Comunicação entre ESPs

A comunicação entre os ESPs *client* e o ESP *server* foi realizada logo em seguida, deixando assim o sistema de comunicação entre os medidores e a central pronto. Foram feitos testes no tempo de envio e recebimento dos dados, bem como a distância que essa comunicação permanecia estável. Dentro de um apartamento de tamanho médio, foi possível a comunicação sem grandes perdas de sinal, a uma distância de 10 a 12 metros com barreiras.

O sistema funciona de maneira simples: no Node *server*, uma rede é criada e definida como receptora, desse modo, o sistema fica verificando se há sinais *Wifi* que estejam procurando por essa rede. O nome da rede é muito importante, pois é essa referência que os Nodes *clients* irão usar na identificação da Central de Controle correta. O código completo está no Apêndice C.

A gravação do Node *client* é feita de maneira análoga ao *server*. Nele é criada uma rede com um nome de identificação, e essa rede é configurada para procurar a rede do *server* a partir do nome dado, que nesse caso foi colocado diretamente no código, porém, pode-se deixar em aberto para que o usuário configure seus Medidores com a Central da própria residência. O código completo está no Apêndice D.

• Teste Final

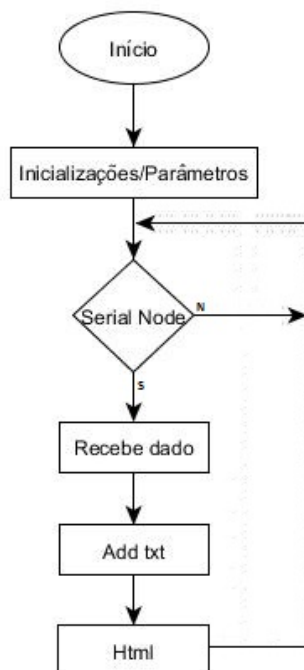
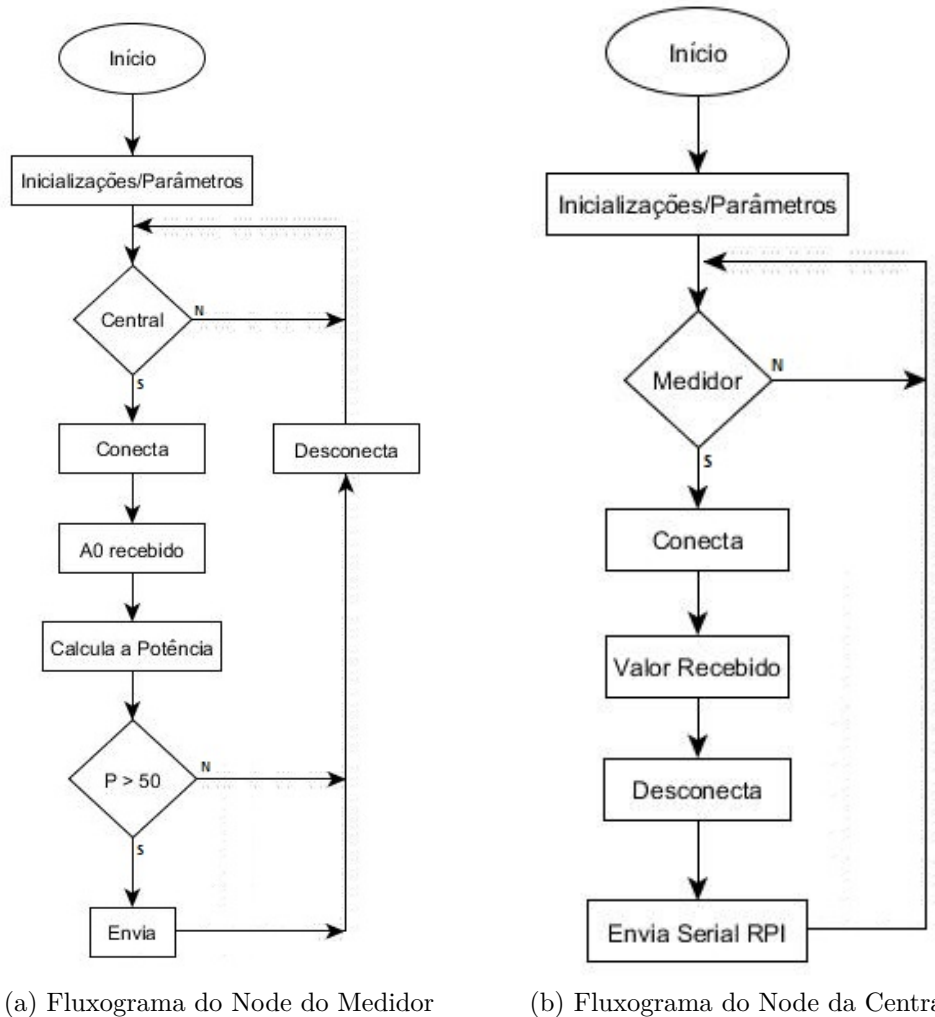
O sensor de corrente ACS712 fornece como resposta, um sinal de tensão proporcional à corrente, que é analisado pelo próprio NodeMCU. O valor de tensão máxima de saída de 5V do sensor passa por um divisor de tensão, tornando-se assim, compatível com a porta de entrada do Node, que é de 3,3V. No código dos Nodes *clients*, foi programado essa análise (Apêndice D), na qual foi utilizada o valor da sensibilidade do sensor, feita individualmente para os medidores seguindo (3.4). O valor analógico medido é convertido para o valor referente de corrente, assim como foi explicado na seção dos Medidores (Cap. 3.1.3). Ao final do cálculo de corrente, é obtido o valor da potência, dado por:

$$P = V * I \quad (4.1)$$

Sendo o valor V, a tensão da rede, que foi fixado para o projeto em um único valor, 127V.

Em uma última análise antes de enviar o valor para a Central, é verificado se o aparelho está em funcionamento ou *standby*. Para isso, o valor da potência deve ser maior que 50 *watts*, o que significa um valor de corrente de pelo menos 0.4A. Caso o valor seja menor, o Medidor não envia o dado, fazendo com que o usuário possua apenas as informações mais relevantes para o monitoramento de seu consumo, evitando também uma sobrecarga de informação. Junto com o valor da potência, é enviado um código de identificação (ID) do medidor, para que a Central possa diferenciar os dados recebidos.

Os fluxogramas a seguir mostram a lógica de programação do Node dos medidores, do Node da Central e do RPI.



(c) Fluxograma do Raspberry Pi

Figura 16 – Fluxogramas de toda programação.

5 Resultados

A análise final dos dados é feita pela Central: o valor recebido do Medidor é separado em tabelas de acordo com o ID recebido, junto do horário e data de recebimento. O fluxograma da Fig.17 ilustra a lógica final utilizada nessa análise.



Figura 17 – Fluxograma da análise final

O protótipo final foi montado em duas *protoboards* independentes, sendo uma para a Central e outra para um Medidor. As Figs. 18 e 19 mostram o circuito principal do Medidor, composto pelo NodeMcu, o sensor de corrente ACS712, e um divisor de tensão.

O complemento do Medidor pode ser visto na Fig. 20, tendo um benjamim comum modificado para que o sensor entre em série com o equipamento que será medido. Essa modificação pode ser vista na Fig. 21.

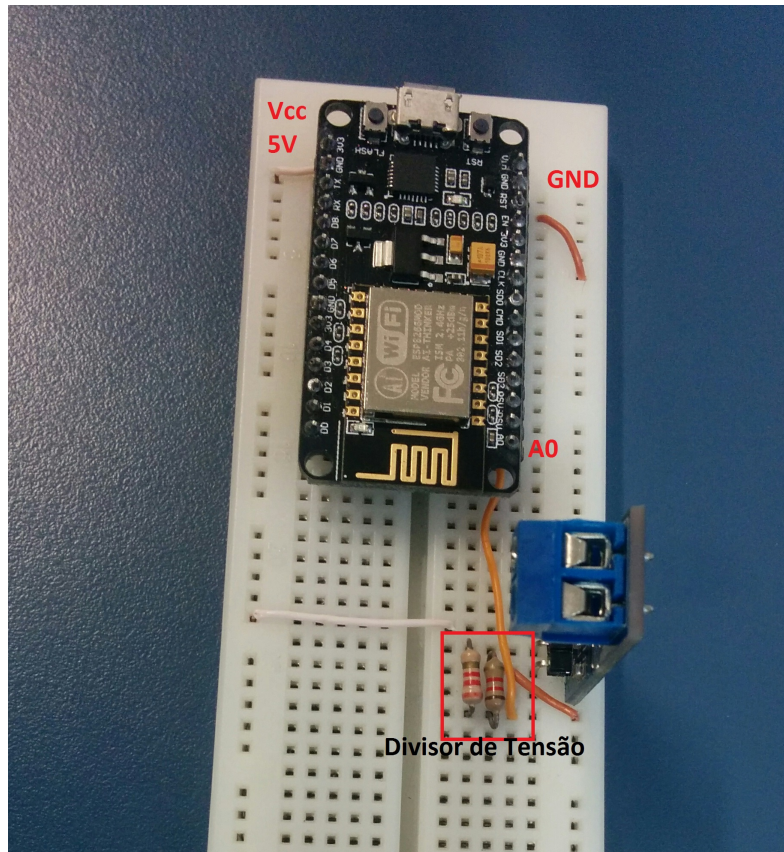
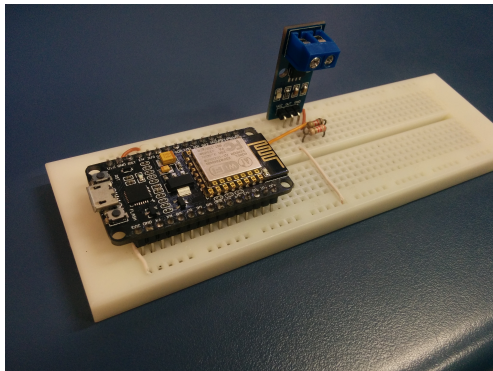
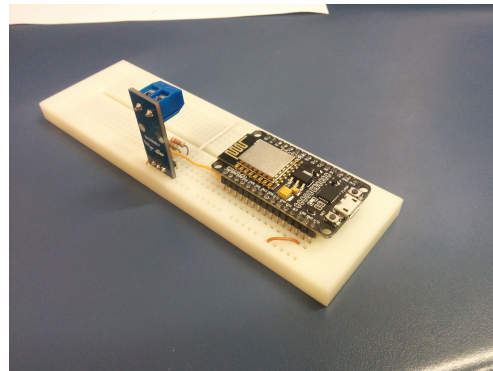


Figura 18 – BLocos que compõem o circuito do Medidor



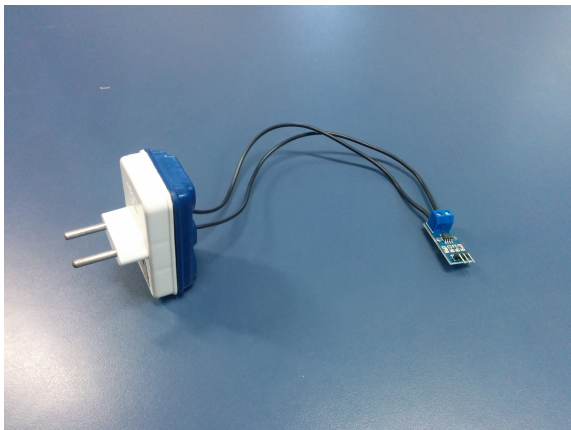
(a) Visão lateral 1 - circuito do Medidor



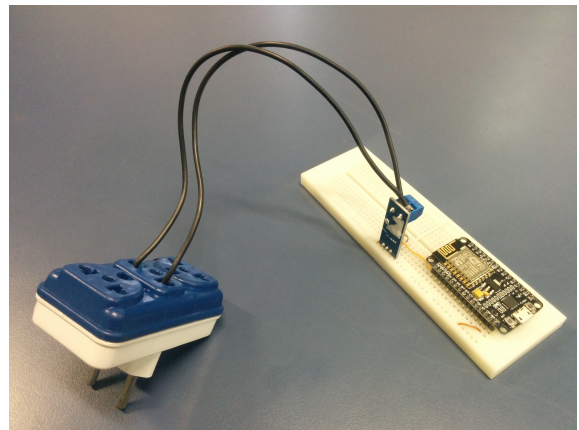
(b) Visão lateral 2 - circuito do Medidor

Figura 19 – Circuito do Medidor

O benjamim utilizado é composto por uma entrada, que se conecta na rede, e três saídas, nas quais, normalmente, três aparelhos podem ser paralelamente conectados. Baseado nesse circuito, foi cortada uma fase (fase A) de uma das saídas (saída 1), separando-a das demais. O sensor de corrente foi conectado entre a fase A da saída 2 e da saída 1, desse modo, quando o aparelho é conectado na saída 1, sua ligação até a rede é feita através da saída 2, passando antes pelo sensor, fazendo com que toda a corrente passe entre os bornes do ACS712.



(a) Benjamim adaptado com o sensor de corrente



(b) Conexão do benjamim no circuito do Medidor

Figura 20 – Circuito do Medidor com benjamim adaptado

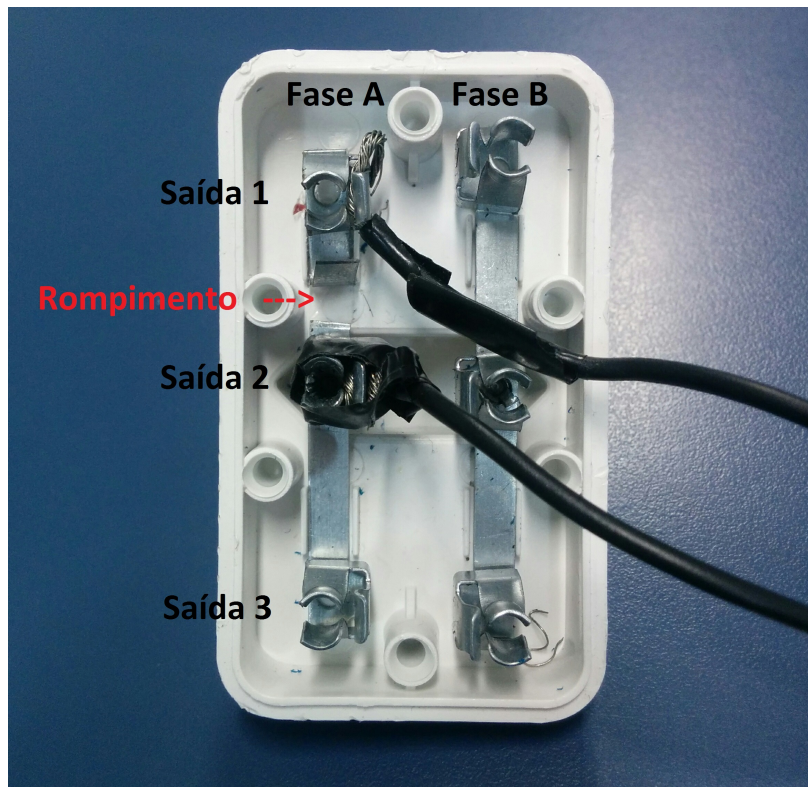


Figura 21 – Visão interna do benjamim

A fase 3 do benjamim permaneceu intacta, sendo utilizada para conectar a fonte de 5V que alimenta o circuito do Medidor. A Fig. 22 mostra como ficou o benjamim adaptado, bem como a Fig. 23, o sistema do Medidor por completo.

A montagem do circuito da Central (Fig. 24) foi ainda mais simples, tendo apenas o RPI e o Node interligados pelos pinos Rx e Tx. Além disso, o próprio RPI alimenta o Node, deixando ainda mais compacto o sistema. Como pode ser visto também, o RPI é alimentado por uma fonte de 5V e tem um cabo *ETHERNET* responsável pela comunicação com a rede do cliente.

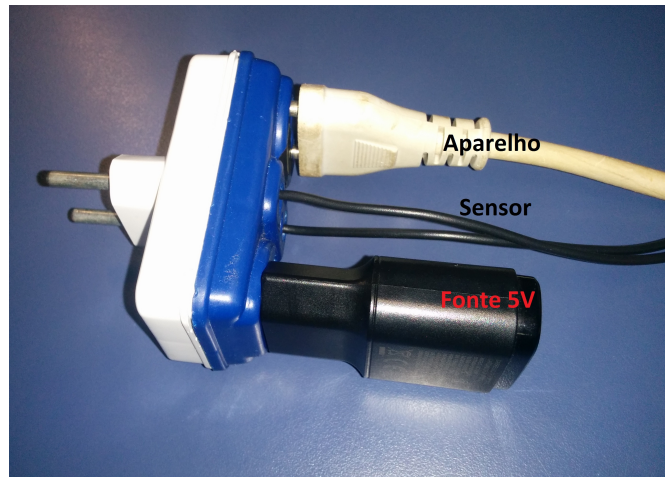


Figura 22 – Conectores do equipamento, sensor e fonte de alimentação

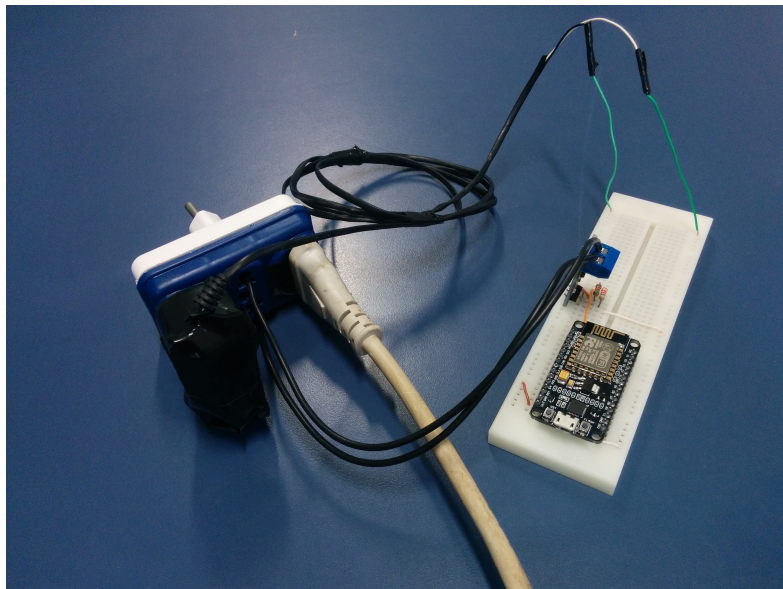


Figura 23 – Protótipo final do Medidor

Para a validação desse projeto, três aparelhos foram utilizados no teste final, um ferro de passar roupa, uma geladeira e um forno elétrico pequeno. Cada um foi conectado ao Medidor e sua potência consumida durante o uso, enviada à Central, armazenada e disponibilizada via *browser* para o usuário, junto de uma estimativa do consumo em reais. A página que o usuário tem acesso, pode ser visualizada na Fig. 25 e as medidas realizadas se encontram na Fig. 26. Esses aparelhos foram escolhidos por possuírem correntes e gastos relativamente altos, fazendo com que uma economia no uso dos mesmos cause um impacto considerável na conta de energia.

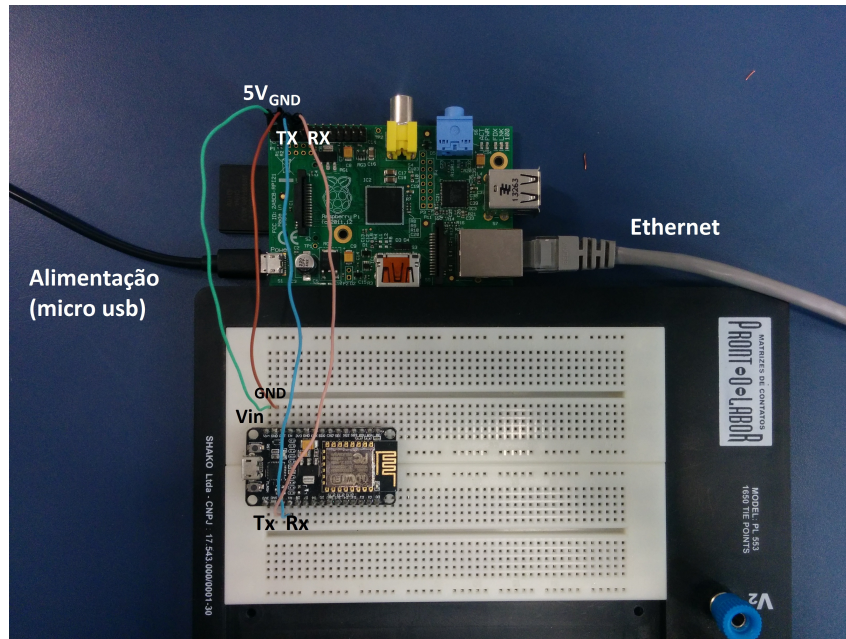


Figura 24 – Protótipo final da Central

O Medidor 1, 2 e 3 representa o ferro de passar roupa, a geladeira e o forno elétrico, respectivamente, sendo que como resultado final foi escolhido apresentar apenas o valor simples da potência consumida, pois é o valor de maior importância para o projeto, já que a partir desse dado, toda informação referente ao consumo do aparelho pode ser obtida.

Central de Controle

10.235.0.229/cgi-bin/control_unit.py

CENTRAL DE CONTROLE

Ambiente de Interface com o Usuário

Dados Armazenados:

Data	Medidor	Potência (Watts)	Data	Medidor	Potência (Watts)	Data	Medidor	Potência (Watts)
08/11/2016 08:49:36	1	1053.04	08/11/2016 08:57:02	2	530.72	08/11/2016 09:14:57	3	1530.72
08/11/2016 08:50:00	1	1053.41	08/11/2016 09:07:44	2	530.28	08/11/2016 09:15:00	3	1531.81
08/11/2016 08:51:04	1	1053.14	08/11/2016 09:07:45	2	530.22	08/11/2016 09:15:04	3	1531.15
08/11/2016 08:51:05	1	1053.43	08/11/2016 09:07:46	2	530.19	08/11/2016 09:15:07	3	1531.15
08/11/2016 08:51:06	1	1052.98	08/11/2016 09:07:47	2	530.22	08/11/2016 09:15:09	3	1531.24
08/11/2016 08:51:07	1	1052.96	08/11/2016 09:07:48	2	530.09	08/11/2016 09:15:11	3	1531.00
08/11/2016 08:51:09	1	1053.41	08/11/2016 09:07:49	2	529.95	08/11/2016 09:15:25	3	1531.27
08/11/2016 08:51:10	1	1053.14	08/11/2016 09:07:50	2	529.94	08/11/2016 09:15:26	3	1531.48
08/11/2016 08:51:11	1	1053.52	08/11/2016 09:07:51	2	530.39	08/11/2016 09:15:27	3	1529.79
08/11/2016 08:51:12	1	1052.96	08/11/2016 09:07:52	2	530.73	08/11/2016 09:15:28	3	1529.97
08/11/2016 08:51:13	1	1053.52	08/11/2016 09:08:04	2	530.81	08/11/2016 09:15:29	3	1530.18
08/11/2016 08:57:04	1	1053.23	08/11/2016 09:08:05	2	530.78	08/11/2016 09:15:30	3	1530.33
08/11/2016 08:59:24	1	1052.96	08/11/2016 09:08:06	2	530.25	08/11/2016 09:15:31	3	1530.85
08/11/2016 08:59:25	1	1053.35	08/11/2016 09:08:07	2	530.32	08/11/2016 09:15:32	3	1531.27
08/11/2016 08:59:26	1	1053.37	08/11/2016 09:08:08	2	530.30	08/11/2016 09:15:41	3	1531.06
08/11/2016 08:59:27	1	1053.23	08/11/2016 09:08:09	2	530.33	08/11/2016 09:15:42	3	1531.24
08/11/2016 08:59:28	1	1052.02	08/11/2016 09:08:10	2	530.14	08/11/2016 09:15:43	3	1531.30
08/11/2016 08:59:29	1	1051.83	08/11/2016 09:08:11	2	530.41	08/11/2016 09:15:44	3	1531.33
08/11/2016 08:59:30	1	1052.06	08/11/2016 09:08:12	2	530.79	08/11/2016 09:15:45	3	1531.48
08/11/2016 08:59:31	1	1052.96	08/11/2016 09:08:20	2	530.76	08/11/2016 09:15:46	3	1531.15
08/11/2016 08:59:32	1	1053.12	08/11/2016 09:08:21	2	530.76	08/11/2016 09:15:47	3	1531.63
08/11/2016 09:00:17	1	1052.91	08/11/2016 09:08:22	2	530.84	08/11/2016 09:15:48	3	1531.63
08/11/2016 09:00:18	1	1052.89	08/11/2016 09:08:23	2	530.99	08/11/2016 09:15:49	3	1531.39
08/11/2016 09:00:19	1	1053.10	08/11/2016 09:08:24	2	530.85	08/11/2016 09:15:50	3	1531.45
08/11/2016 09:00:20	1	1052.77	08/11/2016 09:08:25	2	530.98	08/11/2016 09:15:51	3	1531.30
08/11/2016 09:00:21	1	1053.04	08/11/2016 09:08:26	2	530.80	08/11/2016 09:15:57	3	1531.18

Item	Consumo em reais
Medidor 1	R\$ 1.57
Medidor 2	R\$ 0.73
Medidor 3	R\$ 1.09

Figura 25 – Interface com o Usuário - Página Final

Data	Medidor	Potência (Watts)	Data	Medidor	Potência (Watts)	Data	Medidor	Potência (Watts)	Item	Consumo em reais
08/11/2016 08:49:36	1	1053.04	08/11/2016 08:57:02	2	530.72	08/11/2016 09:14:57	3	1530.72	Medidor 1	R\$ 1.57
08/11/2016 08:50:00	1	1053.41	08/11/2016 09:07:44	2	530.28	08/11/2016 09:15:00	3	1531.81	Medidor 2	R\$ 0.73
08/11/2016 08:51:04	1	1053.14	08/11/2016 09:07:45	2	530.22	08/11/2016 09:15:04	3	1531.15	Medidor 3	R\$ 1.09
08/11/2016 08:51:05	1	1053.43	08/11/2016 09:07:46	2	530.19	08/11/2016 09:15:07	3	1531.15		
08/11/2016 08:51:06	1	1052.98	08/11/2016 09:07:47	2	530.22	08/11/2016 09:15:09	3	1531.24		
08/11/2016 08:51:07	1	1052.96	08/11/2016 09:07:48	2	530.09	08/11/2016 09:15:11	3	1531.00		
08/11/2016 08:51:09	1	1053.41	08/11/2016 09:07:49	2	529.95	08/11/2016 09:15:25	3	1531.27		
08/11/2016 08:51:10	1	1053.14	08/11/2016 09:07:50	2	529.94	08/11/2016 09:15:26	3	1531.48		
08/11/2016 08:51:11	1	1053.52	08/11/2016 09:07:51	2	530.39	08/11/2016 09:15:27	3	1529.79		
08/11/2016 08:51:12	1	1052.96	08/11/2016 09:07:52	2	530.73	08/11/2016 09:15:28	3	1529.97		
08/11/2016 08:51:13	1	1053.52	08/11/2016 09:08:04	2	530.81	08/11/2016 09:15:29	3	1530.18		
08/11/2016 08:57:04	1	1053.23	08/11/2016 09:08:05	2	530.78	08/11/2016 09:15:30	3	1530.33		
08/11/2016 08:59:24	1	1052.96	08/11/2016 09:08:06	2	530.25	08/11/2016 09:15:31	3	1530.85		
08/11/2016 08:59:25	1	1053.35	08/11/2016 09:08:07	2	530.32	08/11/2016 09:15:32	3	1531.27		
08/11/2016 08:59:26	1	1053.37	08/11/2016 09:08:08	2	530.30	08/11/2016 09:15:41	3	1531.06		
08/11/2016 08:59:27	1	1053.23	08/11/2016 09:08:09	2	530.33	08/11/2016 09:15:42	3	1531.24		
08/11/2016 08:59:28	1	1052.02	08/11/2016 09:08:10	2	530.14	08/11/2016 09:15:43	3	1531.30		
08/11/2016 08:59:29	1	1051.83	08/11/2016 09:08:11	2	530.41	08/11/2016 09:15:44	3	1531.33		
08/11/2016 08:59:30	1	1052.06	08/11/2016 09:08:12	2	530.79	08/11/2016 09:15:45	3	1531.48		
08/11/2016 08:59:31	1	1052.96	08/11/2016 09:08:20	2	530.76	08/11/2016 09:15:46	3	1531.15		
08/11/2016 08:59:32	1	1053.12	08/11/2016 09:08:21	2	530.76	08/11/2016 09:15:47	3	1531.63		
08/11/2016 09:00:17	1	1052.91	08/11/2016 09:08:22	2	530.84	08/11/2016 09:15:48	3	1531.63		
08/11/2016 09:00:18	1	1052.89	08/11/2016 09:08:23	2	530.99	08/11/2016 09:15:49	3	1531.39		
08/11/2016 09:00:19	1	1053.10	08/11/2016 09:08:24	2	530.85	08/11/2016 09:15:50	3	1531.45		
08/11/2016 09:00:20	1	1052.77	08/11/2016 09:08:25	2	530.98	08/11/2016 09:15:51	3	1531.30		
08/11/2016 09:00:21	1	1053.04	08/11/2016 09:08:26	2	530.80	08/11/2016 09:15:57	3	1531.18		
08/11/2016 09:00:22	1	1053.14	08/11/2016 09:08:27	2	530.87	08/11/2016 09:15:58	3	1531.48		
08/11/2016 09:00:30	1	1052.62	08/11/2016 09:08:28	2	530.86	08/11/2016 09:15:59	3	1531.24		
08/11/2016 09:00:31	1	1053.00	08/11/2016 09:08:29	2	530.94	08/11/2016 09:16:00	3	1530.54		
08/11/2016 09:00:32	1	1052.87	08/11/2016 09:08:30	2	530.80	08/11/2016 09:16:01	3	1529.79		
08/11/2016 09:00:33	1	1052.98	08/11/2016 09:08:31	2	530.88	08/11/2016 09:16:02	3	1529.92		

Figura 26 – Medidas obtidas nos testes com os equipamentos

5.1 Estimativa dos Custos do Projeto

Abaixo tem-se uma estimativa do preço total do sistema considerando a construção de 1 Central e 3 Medidores. Os valores foram encontrados em lojas online e podem variar de acordo com o número de itens comprados, sendo uma compra em larga escala, muito mais vantajosa.

Material	Preço por Unidade	Unidades	Total
RaspberryPi	R\$150,00	1	R\$150,00
ACS712	R\$15,00	3	R\$45,00
NodeMCU	R\$40,00	1	R\$40,00
ESP12e	R\$15,00	2	R\$30,00
Outros	R\$30,00	-	R\$30,00
		Total:	R\$295,00
		Custo Central*	R\$165,00
		Custo por Medidor*	R\$30,00

Observação: * O NodeMCU pode ser substituído pelo ESP12e.

6 Considerações Finais

O objetivo deste protótipo foi garantir uma medida confiável e uma comunicação entre os Medidores a Central de qualidade, de forma que nenhum dado de consumo seja perdido. Dessa maneira, algumas limitações foram impostas a fim de manter o foco onde era mais importante.

Com o uso do sensor de corrente ACS712 de 30A, medidas de correntes baixas, de até 2A, possuem uma precisão baixa, porém, aparelhos domésticos que usam apenas 2A ou menos, não têm um impacto relevante no consumo final total, ou seja, uma economia no uso de tais aparelhos não causaria grandes resultados tanto no âmbito financeiro quanto ambiental, o que fez com que esse problema fosse descartado.

Outro detalhe do projeto, foi a respeito da tensão de entrada do aparelho medido. Para o cálculo final da potência, foi utilizado um valor fixo de 127V para a tensão, de maneira que, caso o protótipo seja testado em aparelhos 220V, haverá uma divergência nos resultados obtidos. Entretanto, nenhum dano será causado, tanto ao aparelho, quanto ao sistema do Medidor. De qualquer forma, esse problema pode ser facilmente resolvido utilizando um sensor de tensão ou qualquer outra lógica que verifique esse valor de entrada, porém, a fim de apenas verificar a usabilidade do Sistema Inteligente, esse detalhe também foi desconsiderado.

Devido à estas limitações, o protótipo desenvolvido serve para aparelhos de baixa e média potência, não tendo suporte para grandes equipamentos industriais. Entretanto, a adaptação do sistema para o uso industrial pode ocorrer de maneira simples, dando-se principalmente pela substituição do sensor de corrente.

Outro ponto que pode ser incorporado, é na maneira como o usuário tem acesso aos resultados. Neste projeto, não foi dado muito enfoque à essa interface, porém, a partir da criação de um banco de dados mais rebuscado, é possível a criação de um software ou aplicativo pelo qual o usuário possa ter acesso de qualquer lugar, necessitando apenas de comunicação com a internet.

O objetivo principal do projeto foi alcançado com êxito. As medidas enviadas dos Medidores à Central ocorreram da maneira esperada e todos os dados obtidos no ambiente final de interface com o usuário foram satisfatórios.

APÊNDICE A – Pesquisa Realizada

Qual sua idade? (116 respostas)

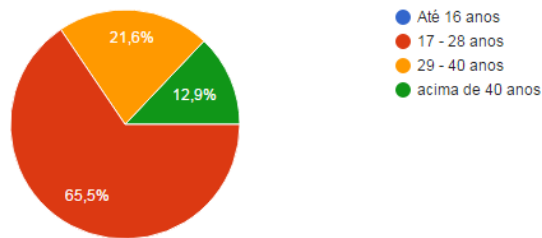


Figura 27 – Enquete: Pergunta 1

Você se considera um consumidor consciente? (116 respostas)

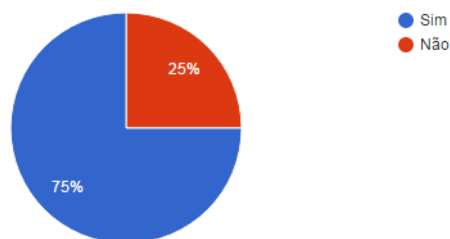


Figura 28 – Enquete: Pergunta 2

Você sabe qual o valor cobrado do kWh (quilowatt-hora) da sua conta de luz?
(116 respostas)

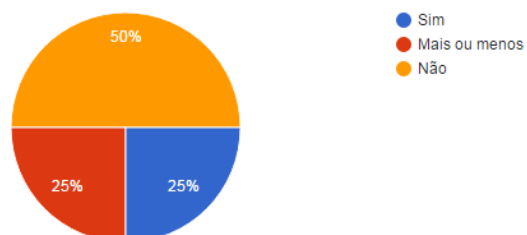


Figura 29 – Enquete: Pergunta 3

Você sabe quanto cada aparelho da sua casa consome de energia?
(116 respostas)

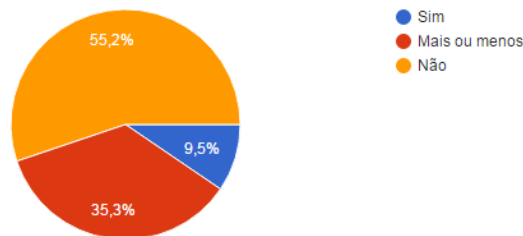


Figura 30 – Enquete: Pergunta 4

Você gostaria de saber quanto cada aparelho da sua casa gasta durante determinado uso?
(116 respostas)

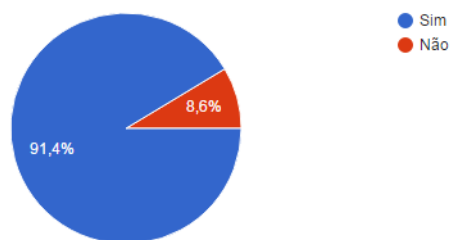


Figura 31 – Enquete: Pergunta 5

Se você soubesse, você acha que conseguiria economizar energia?
(116 respostas)

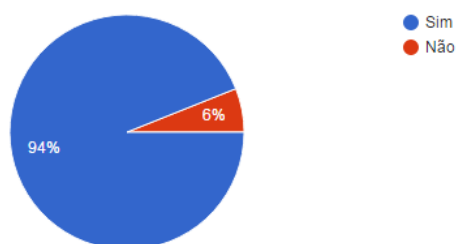


Figura 32 – Enquete: Pergunta 6

APÊNDICE B – Código *Wifi* Simples

```
1 #include <ESP8266WiFi.h>
2
3 const char* ssid = "wireless-lac"; //nome da rede
4 const char* password = "12345678";
5
6 ; //
7 WiFiServer server(80);
8
9 void setup() {
10 Serial.begin(115200);
11 delay(10);
12 pinMode(5, OUTPUT);
13 pinMode(4, OUTPUT);
14 pinMode(0, OUTPUT);
15 pinMode(13, OUTPUT);
16 digitalWrite(5, LOW);
17 digitalWrite(4, LOW);
18 digitalWrite(0, LOW);
19 digitalWrite(13, LOW);
20
21
22 // Connect to WiFi network
23 Serial.println();
24 Serial.println();
25 Serial.print("Connecting to ");
26 Serial.println(ssid);
27
28 WiFi.begin(ssid, password);
29
30 while (WiFi.status() != WL_CONNECTED) {
31 delay(500);
32 Serial.print(".");
33 }
34 Serial.println("");
35 Serial.println("WiFi connected");
36
37 // Start the server
38 server.begin();
```

```
39 Serial.println("Server started");
40
41 // Print the IP addresscc
42 Serial.print("Use this URL to connect: ");
43 Serial.print("http://");
44 Serial.println(WiFi.localIP());
45 Serial.println(WiFi.macAddress());
46 Serial.println("/");
47 delay(1);
48
49
50 }
51
52 void loop() {
53 // Check if a client has connected
54 WiFiClient client = server.available();
55 if (!client) {
56 return; }
57
58 // Wait until the client sends some data
59 Serial.println("new client");
60 while(!client.available()){
61 delay(1); }
62
63 // Read the first line of the request
64 String request = client.readStringUntil('\r');
65 Serial.println(request);
66 client.flush();
67
68 // Match the request
69
70 if (request.indexOf("/light1on") > 0) {
71 digitalWrite(5, HIGH);}
72 if (request.indexOf("/light1off") >0) {
73 digitalWrite(5, LOW);}
74 if (request.indexOf("/light2on") > 0) {
75 digitalWrite(4, HIGH);}
76 if (request.indexOf("/light2off") >0) {
77 digitalWrite(4, LOW);}
78 if (request.indexOf("/light3on") >0) {
79 digitalWrite(0, HIGH);}
80 if (request.indexOf("/light3off") > 0) {
```

```

81 digitalWrite(0, LOW);}
82 if (request.indexOf("/light4on") > 0) {
83 digitalWrite(13, HIGH);}
84 if (request.indexOf("/light4off") > 0) {
85 digitalWrite(13, LOW);}
86 // Set ledPin according to the request
87 //digitalWrite(ledPin, value);
88
89 // Return the response
90 client.println("HTTP/1.1 200 OK");
91 client.println("Content-Type: text/html");
92 client.println(""); // do not forget this one
93 client.println("<!DOCTYPE HTML>");
94 client.println("<html>");
95 client.println("<head>");
96 client.println("<meta name='apple-mobile-web-app-capable' content
    ='yes' />");
97 client.println("<meta name='apple-mobile-web-app-status-bar-style
    ' content='black-translucent' />");
98 client.println("</head>");
99 client.println("<body bgcolor = \"#f7e6ec\">");
100 client.println("<hr/><hr>");
101 client.println("<h4><center> Esp8266 Electrical Device Control </
    center></h4>");
102 client.println("<hr/><hr>");
103 client.println("<br><br>");
104 client.println("<br><br>");
105 client.println("<center>");
106 client.println("Device 1");
107 client.println("<a href=\"/light1on\"><button>Turn On </button
    ></a>");
108 client.println("<a href=\"/light1off\"><button>Turn Off </
    button></a><br />");
109 client.println("</center>");
110 client.println("<br><br>");
111 client.println("<center>");
112 client.println("Device 2");
113 client.println("<a href=\"/light2on\"><button>Turn On </button
    ></a>");
114 client.println("<a href=\"/light2off\"><button>Turn Off </
    button></a><br />");
115 client.println("</center>");

```

```
116 client.println("<br><br>");
117 client.println("<center>");
118 client.println("Device 3");
119 client.println("<a href=\""/light3on\"/\"><button>Turn On </button
    ></a>");
120 client.println("<a href=\""/light3off\"/\"><button>Turn Off </
    button></a><br />");
121 client.println("</center>");
122 client.println("<br><br>");
123 client.println("<center>");
124 client.println("Device 4");
125 client.println("<a href=\""/light4on\"/\"><button>Turn On </button
    ></a>");
126 client.println("<a href=\""/light4off\"/\"><button>Turn Off </
    button></a><br />");
127 client.println("</center>");
128 client.println("<br><br>");
129 client.println("<center>");
130 client.println("<table border=\"5\">");
131 client.println("<tr>");
132 if (digitalRead(5)) {
133 client.print("<td>Light 1 is ON</td>");}
134 else {
135 client.print("<td>Light 1 is OFF</td>");}
136 client.println("<br />");
137 if (digitalRead(4)) {
138 client.print("<td>Light 2 is ON</td>"); }
139 else {
140 client.print("<td>Light 2 is OFF</td>");}
141 client.println("</tr>");
142 client.println("<tr>");
143 if (digitalRead(0)){
144 client.print("<td>Light 3 is ON</td>");}
145 else {
146 client.print("<td>Light 3 is OFF</td>");}
147 if (digitalRead(13)) {
148 client.print("<td>Light 4 is ON</td>"); }
149 else {
150 client.print("<td>Light 4 is OFF</td>"); }
151 client.println("</tr>");
152 client.println("</table>");
153 client.println("</center>");
```

```
154 client.println("</html>");
155 delay(1);
156 Serial.println("Client disonnected");
157 Serial.println("");
158
159 }
```


APÊNDICE C – Código *Wifi Server*

```

1  #include <ESP8266WiFi.h>
2
3  // Define I/O Pins
4  #define      LED0      2          // WIFI Module LED
5  #define      LED1      D0        // Connectivity With Client
6
7
8  // Authentication Variables
9  char*        Central_Inteligente; // SERVER WIFI NAME
10 char*        CIPassword;          // SERVER PASSWORD
11
12 #define      MAXSC      6          // MAXIMUM NUMBER OF CLIENTS
13
14 WiFiServer   CI_Server(9001);      // THE SERVER AND THE PORT
15           NUMBER
16 WiFiClient   CI_Client[MAXSC];     // THE SERVER CLIENTS
17
18 void setup()
19 {
20     // Setting The Serial Port
21     Serial.begin(115200);           // Computer Communication
22
23     // Setting The Mode Of Pins
24     pinMode(LED0, OUTPUT);          // WIFI OnBoard LED Light
25     pinMode(LED1, OUTPUT);          // Indicator For Client #1
26           Connectivity
27
28     // Setting Up A Wifi Access Point
29     SetWifi("CENTRAL_INTELIGENTE", ""); //nome do wifi que o
30           cliente vai procurar
31
32 }
33
34 void loop()
35 {
36     IsClients();
37 }

```

```
36
37
38
39 void SetWifi(char* Name, char* Password)
40 {
41     // Stop Any Previous WIFI
42     WiFi.disconnect();
43
44     // Setting The Wifi Mode
45     WiFi.mode(WIFI_AP_STA);
46     Serial.println("WIFI Mode : AccessPoint Station");
47
48     // Setting The AccessPoint Name & Password
49     Central_Inteligente = Name;
50     CIPassword = Password;
51
52     // Starting The Access Point
53     WiFi.softAP(Central_Inteligente, CIPassword);
54     Serial.println("WIFI < " + String(Central_Inteligente) + " >
55         Started");
56
57     // Wait For Few Seconds
58     delay(1000);
59
60     // Getting Server IP
61     IPAddress IP = WiFi.softAPIP();
62
63     // Printing The Server IP Address
64     Serial.print("AccessPoint IP : ");
65     Serial.println(IP);
66
67     // Starting Server
68     CI_Server.begin();
69     Serial.println("Server Started");
70
71
72
73 void IsClients()
74 {
75     if(CI_Server.hasClient())
76     {
```

```
77     WiFiClient CI_Client = CI_Server.available();
78
79     if(digitalRead(LED0) == HIGH) digitalWrite(LED0, LOW);
80
81     digitalWrite(LED1, HIGH);
82
83     while(1)
84     {
85         //-----
86         // If Clients Are Connected
87         //-----
88
89         if(CI_Client.available())
90         {
91             // Here We Read The Message
92             String Message = CI_Client.readStringUntil('\r');
93             // Here We Print The Message On The Screen
94             Serial.println(Message);
95             // Here We Reply To The Client With A Message
96             // CI_Client.print("\nWe Got Your Message");
97             CI_Client.flush();
98         }
99
100         //-----
101         // If Clients Are Disconnected
102         //-----
103
104         if(!CI_Client || !CI_Client.connected())
105         {
106             // Here We Turn Off The LED To Indicated The Its
107             // Disconnected
108             digitalWrite(LED1, LOW);
109             // Here We Jump Out Of The While Loop
110             break;
111         }
112     }
113 }
114 else
115 {
116     // This LED Blinks If No Clients Where Available
117     digitalWrite(LED0, HIGH);
```

```
118         delay(250);
119         digitalWrite(LED0, LOW);
120         delay(250);
121     }
122 }
```

APÊNDICE D – Código *Wifi Client*

```

1  #include <Wire.h>
2  #include <ESP8266WiFi.h>
3
4  // Defining I/O Pins
5
6  #define          LED0          2          // WIFI Module LED
7  #define          LED1          D0          // Connectivity With Client
8
9  // Variables
10
11 float Sensibilidade = 0.0378745478; //sensibilidade em Volts/
    Ampere para sensor de 30A
12 float voltAcms = 3.33;
13 float tensao = 127; //valor de tensao para o calculo de potencia
14 float power;
15
16 const String  ID_client          = "3"; // nome/id de identificacao
    do Medidor
17
18 // LED Delay Variables
19
20 int          LEDState          = LOW;
21 unsigned long CurrMillis          = 0;
22 unsigned long PrevMillis          = 0;
23 unsigned long Interval          = 1000;
24
25 // Authentication Variables
26
27 char*          Central_Inteligente;
28 char*          CIpasword;
29 IPAddress      CI_Server(192,168,4,1);
30 WiFiClient      CI_Client;
31
32 void setup()
33 {
34 // Setting The Serial Port
35 Serial.begin(115200);          // Computer Communication
36

```

```
37 // Setting The Mode Of Pins
38 pinMode(LED0, OUTPUT);           // WIFI OnBoard LED Light
39 pinMode(LED1, OUTPUT);           // Indicator For Client #1
    Connectivity
40 digitalWrite(LED0, !LOW);        // Turn WiFi LED Off
41
42 // Starting To Connect
43 WiFi.mode(WIFI_STA);             // To Avoid Broadcasting An SSID
44 WiFi.begin("CENTRAL_INTELIGENTE"); // The SSID That We
    Want To Connect To
45
46 // Printing Message For User That Connetion Is On Process
47 Serial.println("!--- Connecting To " + WiFi.SSID() + " ---!");
48
49 // WiFi Connectivity
50 CheckConnectivity();             // Checking For Connection
51
52 // Stop Blinking To Indicate Connected
53 digitalWrite(LED0, !HIGH);
54 Serial.println("!-- Client Device Connected --!");
55
56 // Printing IP Address
57 Serial.println("Connected To      : " + String(WiFi.SSID()));
58 Serial.println("Signal Strenght   : " + String(WiFi.RSSI()) + "
    dBm");
59 Serial.print  ("Server IP Address : ");
60 Serial.println(CI_Server);
61 Serial.print  ("Device IP Address : ");
62 Serial.println(WiFi.localIP());
63
64 // Conecting The Device As A Client
65 CI_Request();
66 }
67
68
69 void loop()
70 {
71   ReadPower();
72 }
73
74
75 void ReadPower()
```

```
76 {
77   power = calcular_potencia();
78   if (power > 50)    // minimo de corrente 0.4A
79   {
80     CI_Client.println (ID_client + "|" + String(power) + ";");
81     delay(100); // tempo de leitura
82     CI_Client.flush();
83     delay(100);
84   }
85 }
86
87 float calcular_potencia()
88 {
89   float voltageSensor =get_voltage(1000) ;//obtemos a tensao do
        sensor usando 1000 amostras
90   float I=(voltageSensor-(voltAcms/2))/Sensibilidade; //Equacao
        para obter a corrente
91   delay(10);
92   float pot = I * tensao    ;
93   return pot;
94 }
95
96 float get_voltage(int n_amostras)
97 {
98   float voltage=0;
99   for(int i=0;i<n_amostras;i++)
100  {
101    voltage =voltage+analogRead(A0) * (3.33 / 1023.0);
102  }
103  voltage=voltage/n_amostras; // valor de pico da tensao
104  voltage=voltage*0.707; //valor eficaz
105  return(voltage);
106 }
107
108
109 void CheckConnectivity()
110 {
111   while(WiFi.status() != WL_CONNECTED)
112   {
113     for(int i=0; i < 10; i++)
114     {
115       digitalWrite(LED0, !HIGH);
```

```
116 delay(250);
117 digitalWrite(LED0, !LOW);
118 delay(250);
119 Serial.print(".");
120 }
121 Serial.println("");
122 }
123 }
124
125
126 void CI_Request()
127 {
128 // First Make Sure You Got Disconnected
129 CI_Client.stop();
130
131 // If Sucessfully Connected Send Connection Message
132 if(CI_Client.connect(CI_Server, 9001))
133 {
134 Serial.println   ("<" + ID_client + "-CONNECTED>");
135 CI_Client.println("<" + ID_client + "-CONNECTED>");
136 }
137 }
```

Referências

- CASTELLS, M. O. *A Sociedade em Rede*. São Paulo: Paz e Terra, 1999. v. 1. Citado na página 18.
- GE Look Ahead. *A Quiet Revolution no More*. 2014. <<http://gelookahead.economist.com/quiet-revolution-industrial-internet/>>. Acessado: 2016-10-17. Citado na página 18.
- LLC, A. M. *ACS712 - Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor*. Worcester: Allegro MicroSystems LLC, 2012. <<http://www.datasheetpdf.com/datasheet/ACS712.html>>. Rev. 15, Acessado: 2016-10-08. Citado na página 22.
- MACHADO, J. *Raspberry Pi: Computador de 50 Reais Pode Revolucionar a Informática*. 2012. <<http://www.tecmundo.com.br/hardware/23175-raspberry-pi-como-um-computador-de-50-reais-pode-revolucionar-a-informatica.htm>>. Acessado: 2016-10-08. Citado na página 25.
- Ministério do Meio Ambiente. *Quem é o Consumidor Consciente*. 2014. <<http://www.mma.gov.br/responsabilidade-socioambiental/producao-e-consumo-sustentavel/consumo-consciente-de-embalagem/quem-e-o-consumidor-consciente>>. Acessado: 2016-10-15. Citado na página 18.
- Raspberry Pi Foundation. *ABOUT US*. 2016. <<https://www.raspberrypi.org/about/>>. Acessado: 2016-07-11. Citado na página 25.
- SENSORS, A. C. *Engineer's Reference Handbook*. 2002. Citado 3 vezes nas páginas 13, 21 e 22.
- SILVA, R. L. da. *Manual Tecnologia Wireless - Modulo sobre tecnologia Wireless*. 2008. Citado na página 23.
- SYSTEMS, E. *ESP8266EX Datasheet*. Shanghai: Espressif Systems, 2016. <https://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf>. Version 4.9. Citado na página 24.
- Thomsen, Adilson - Blog Filipe Flop. *Como Programar o NodeMCU com IDE Arduino*. 2016. <<http://blog.filipeflop.com/wireless/programar-nodemcu-com-ide-arduino.html>>. Acessado: 2016-06-12. Citado na página 31.
- WIKIPÉDIA. *Efeito Hall*. 2016. <https://pt.wikipedia.org/wiki/Efeito_Hall>. Acessado: 2016-08-22. Citado na página 21.
- WITCHALLS, C.; CHAMBERS, J. The internet of things business index: A quiet revolution gathers pace. *The Economist Intelligence Unit.*, 2013. Citado na página 17.