

Fabiano Azevedo César Jardim

**Projeto e Desenvolvimento de uma Régua  
Inteligente para Monitoramento do Consumo  
de Energia de Eletrodomésticos**

Belo Horizonte

2018



Fabiano Azevedo César Jardim

**Projeto e Desenvolvimento de uma Régua Inteligente  
para Monitoramento do Consumo de Energia de  
Eletrodomésticos**

Monografia apresentada durante o Seminário dos Trabalhos de Conclusão do Curso de Graduação em Engenharia Elétrica da UFMG, como parte dos requisitos necessários à obtenção do título de Engenheiro Eletricista.

Universidade Federal de Minas Gerais – UFMG

Escola de Engenharia

Curso de Graduação em Engenharia Elétrica

Orientador: Alair Dias Júnior

Belo Horizonte

2018



# Listas de ilustrações

Figura 1 – Diferentes situações de desempenho de um sistema . . . . .	18
Figura 2 – Diagrama de módulos da Régua Inteligente . . . . .	28
Figura 3 – Exemplo de resposta esperada dos sensores de tensão e corrente . . . . .	28
Figura 4 – Diagrama de rede do projeto da Régua Inteligente . . . . .	30
Figura 5 – Sensor de tensão baseado no transformador ZMPT101B . . . . .	43
Figura 6 – Sensor de corrente ACS712 . . . . .	45
Figura 7 – Curva do sensor ACS712 . . . . .	46
Figura 8 – Conversor A/D ADS1015 . . . . .	46
Figura 9 – Conversor lógico digital 5V/3.3V . . . . .	47
Figura 10 – Microcontrolador ESP8266 D1 mini . . . . .	48
Figura 11 – Conversor DC/DC 5V/3.3V . . . . .	50
Figura 12 – Estrutura completa do <i>hardware</i> . . . . .	51
Figura 13 – Servidor HTTP Apache . . . . .	52
Figura 14 – Servidores mais utilizados em sites com alto tráfego . . . . .	53
Figura 15 – Desempenho dos servidores para diferentes quantidades de requisições por segundo . . . . .	53
Figura 16 – Desempenho dos servidores para diferentes concorrências de usuários (1000 requisições por segundo) . . . . .	54
Figura 17 – Banco de dados relacional MySQL . . . . .	55
Figura 18 – Linguagem Python . . . . .	56
Figura 19 – Interface (mod_wsgi) entre a aplicação em Python e o Apache . . . . .	57
Figura 20 – <i>Framework</i> Angular . . . . .	57
Figura 21 – Estrutura completa do <i>software</i> . . . . .	59



# **Lista de tabelas**

Tabela 1 – Resumo das tabelas do link Tabelas...	(2018b) . . . . .	34
Tabela 2 – Especificações do sensor de tensão . . . . .	44	
Tabela 3 – Especificações do sensor de corrente . . . . .	45	
Tabela 4 – Especificações do conversor A/D . . . . .	47	
Tabela 5 – Especificações do conversor lógico digital . . . . .	47	
Tabela 6 – Especificações do microcontrolador . . . . .	49	
Tabela 7 – Especificações do conversor DC/DC . . . . .	50	
Tabela 8 – Valores de potência ativa, corrente eficaz e fator de potência indicados e medidos . . . . .	61	
Tabela 9 – Erro relativo das medições encontradas . . . . .	61	



# Listas de abreviaturas e siglas

A/D	Analógico - Digital ( <i>Analog - Digital</i> )
DC/DC	Corrente Contínua - Corrente Contínua ( <i>Direct Current - Direct Current</i> )
IoT	Internet das Coisas ( <i>Internet of Things</i> )
HTTP	Protocolo de Transferência de Hipertexto ( <i>Hypertext Transfer Protocol</i> )
OSI	Interconexão de Sistemas Abertos ( <i>Open System Interconnection</i> )
FTP	Protocolo de Transferência de Arquivos ( <i>File Transfer Protocol</i> )
SMTP	Protocolo de transferência de correio simples ( <i>Simple Mail Transfer Protocol</i> )
IIS	Serviços de Informação para Internet ( <i>Internet Information Services</i> )
TCP	Protocolo de Controle de Transmissão ( <i>Transmission Control Protocol</i> )
IP	Protocolo da Internet ( <i>Internet Protocol</i> )
DNS	Sistema de Nomes de Domínio ( <i>Domain Name System</i> )
RDBMS	Sistema Gerenciador de Banco de Dados Relacional ( <i>Relational Database Management System</i> )
SQL	Linguagem de Consulta Estruturada ( <i>Structured Query Language</i> )
HTML	Linguagem de Marcação de Hipertexto ( <i>Hypertext Markup Language</i> )
CSS	Folha de Estilo em Cascatas ( <i>Cascading Style Sheets</i> )
GPRS	Serviço Geral de Pacotes de Radio ( <i>General Packet Radio Service</i> )
PLC	Linha de Comunicação de Potência ( <i>Power Line Communication</i> )
REG	Portal de Comunicação de Energia Renovável ( <i>Renewable Energy Gateway</i> )
GSM	Sistema Global para Comunicações Móveis ( <i>Global System for Mobile communications</i> )
SMS	Serviço de Mensagens Curtas ( <i>Short Message Service</i> )

SIM	Módulo de Identificação do Assinante ( <i>Subscriber Identity Module</i> )
BLE	Bluetooth Baixa Consumo de Energia ( <i>Bluetooth Low Energy</i> )
PCB	Placa de Circuito Impresso ( <i>Printed Circuit Board</i> )
API	Interface de Programação de Aplicativos ( <i>Application Programming Interface</i> )
IDE	Ambiente de Desenvolvimento Integrado ( <i>Integrated Development Environment</i> )

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	<b>Formulação do problema</b>	<b>13</b>
1.2	<b>Objetivos</b>	<b>13</b>
1.3	<b>Justificativa</b>	<b>14</b>
1.4	<b>Metodologia</b>	<b>15</b>
1.5	<b>Contribuições</b>	<b>15</b>
1.6	<b>Estrutura do texto</b>	<b>16</b>
<b>2</b>	<b>REVISÃO DA LITERATURA</b>	<b>17</b>
2.1	<b>Conceitos preliminares</b>	<b>17</b>
2.1.1	Fatores importantes de uma medição	18
2.1.2	Exatidão e precisão	18
2.1.3	Erros de medição, sistemático e aleatório	19
2.1.4	Repetitividade	20
2.1.5	Faixa dinâmica de valores da medição	20
2.1.6	Resposta em frequência de um sistema	21
2.1.7	Taxa de amostragem e resolução	22
2.2	<b>Trabalhos relacionados</b>	<b>22</b>
<b>3</b>	<b>ESPECIFICAÇÕES DO SISTEMA</b>	<b>27</b>
3.1	<b>Componentes do <i>hardware</i></b>	<b>27</b>
3.1.1	Fonte de tensão e conversor de tensão DC/DC	27
3.1.2	Sensores de tensão e corrente	28
3.1.3	Conversor A/D	29
3.1.4	Conversor de nível lógico	29
3.1.5	Módulo Wi-Fi	29
3.2	<b>Softwares da aplicação e do servidor</b>	<b>29</b>
3.2.1	<i>Web server</i>	30
3.2.2	<i>Database server</i>	31
3.2.3	<i>Comunicação</i>	32
3.2.4	Interface do navegador	32
3.3	<b>Especificações do <i>hardware</i></b>	<b>33</b>
3.3.1	Faixa de medição de tensão	33
3.3.2	Faixa de medição de corrente	33
3.3.3	Níveis de tensão e corrente máximos suportados pelos sensores	35
3.3.4	Resposta em frequência dos sensores	35

3.3.5	Tensões de comunicação entre os sensores e o conversor A/D . . . . .	35
3.3.6	Tensões e tipo de comunicação entre o conversor A/D e microcontrolador . . . . .	35
3.3.7	Resolução, taxa de amostragem e quantidade de canais do conversor A/D . . . . .	36
3.3.8	Microcontrolador com suporte Wi-Fi . . . . .	36
3.3.9	Consumo de energia dos módulos . . . . .	37
3.3.10	Resumo das especificações do <i>hardware</i> . . . . .	37
<b>3.4</b>	<b>Especificações do <i>software</i></b> . . . . .	<b>38</b>
3.4.1	Quantidade de requisições por segundo . . . . .	38
3.4.2	API de tratamento de requisições e interação com o banco de dados . . . . .	39
3.4.3	Interface de visualização dos dados pelo usuário . . . . .	39
3.4.4	<i>Software</i> do microcontrolador . . . . .	40
3.4.5	Resumo das especificações do <i>software</i> . . . . .	40
<b>4</b>	<b>VALIDAÇÃO DO SISTEMA</b> . . . . .	<b>41</b>
<b>4.1</b>	<b>Processo de validação do <i>hardware</i></b> . . . . .	<b>41</b>
<b>4.2</b>	<b>Processo de validação do <i>software</i></b> . . . . .	<b>41</b>
<b>4.3</b>	<b>Processo de validação do conjunto <i>hardware</i> e <i>software</i></b> . . . . .	<b>42</b>
<b>5</b>	<b>MATERIAIS E MÉTODOS</b> . . . . .	<b>43</b>
<b>5.1</b>	<b>Hardware</b> . . . . .	<b>43</b>
5.1.1	Sensor de tensão . . . . .	43
5.1.2	Sensor de corrente . . . . .	44
5.1.3	Conversor A/D . . . . .	45
5.1.4	Conversor lógico digital . . . . .	47
5.1.5	Microcontrolador . . . . .	48
5.1.6	Conversor DC/DC . . . . .	49
5.1.7	Estrutura de <i>hardware</i> resultante . . . . .	51
<b>5.2</b>	<b>Software</b> . . . . .	<b>51</b>
5.2.1	Servidor HTTP . . . . .	52
5.2.2	Banco de dados . . . . .	55
5.2.3	API em Python . . . . .	55
5.2.4	Interface do navegador . . . . .	57
5.2.5	<i>Software</i> do microcontrolador . . . . .	58
5.2.6	Estrutura de <i>software</i> resultante . . . . .	59
<b>6</b>	<b>RESULTADOS</b> . . . . .	<b>61</b>
<b>6.1</b>	<b>Validações</b> . . . . .	<b>61</b>
<b>7</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b> . . . . .	<b>63</b>





# 1 Introdução

Nos últimos anos tem-se discutido bastante sobre as diferentes formas de atuação do homem para que as nossas relações com a natureza aconteçam de uma forma mais sustentável. O conceito de sustentabilidade e a sua aplicação fazem parte de diversas áreas do nosso conhecimento, e a preocupação com o tema pode ser exemplificada de muitas formas. Uma dessas exemplificações é a preocupação das pessoas em poder melhorar o monitoramento do consumo de energia elétrica de suas casas.

A forma predominante de monitoramento do consumo de eletricidade das casas é por meio de um registro geral. Este equipamento indica o consumo total de energia elétrica da casa ao longo de um período de tempo. Porém, para um controle mais eficaz do consumo individual dos equipamentos, seria interessante a existência de um dispositivo que conseguisse monitorar o consumo de cada equipamento da casa e apresentasse as informações por meio de uma interface simples.

Dentro desse contexto, já estão sendo estudados e fabricados os *smartplugs*, que são dispositivos que tentam fazer esses papéis de monitoramento e apresentação do consumo de energia. O tema deste trabalho é justamente a proposta de um dispositivo que possa monitorar o consumo de energia elétrica de equipamentos elétricos comuns de uma casa e apresentar as informações obtidas por ele através de uma interface simples.

## 1.1 Formulação do problema

O problema consiste em monitorar localmente ou remotamente o consumo de energia elétrica de um equipamento de uma casa e apresentar, de forma simples, informações úteis obtidas pelo monitoramento.

## 1.2 Objetivos

O objetivo geral deste trabalho é o de construir uma Régua Inteligente que possibilitará o monitoramento local ou remoto de energia elétrica de três equipamentos e apresentará as informações obtidas através de uma interface construída para navegadores Web. O objetivo geral se abre nos seguintes objetivos específicos:

- Especificação de sensores de tensão e corrente, conversor A/D, conversor de nível lógico, conversor de tensão DC/DC, módulo Wi-Fi e uma fonte de alimentação;

- Implementação de um programa em um microcontrolador que conterá a lógica de manipulação dos dados recebidos dos sensores e a lógica que enviará via Wi-Fi as informações manipuladas para um servidor Web;
- Implementação de um programa em um servidor para processar, armazenar e posteriormente enviar os dados recebidos da Régua Inteligente;
- Implementação de uma interface visual em um navegador Web que exiba as informações recebidas do servidor para o usuário.

O primeiro objetivo específico apresentado representa a implementação do *hardware* da Régua Inteligente. Os outros três representam a implementação do *software* do projeto.

### 1.3 Justificativa

O projeto da Régua Inteligente proposto neste trabalho abrangerá diferentes áreas da Engenharia Elétrica. Para a produção do *hardware* serão necessárias especificações teóricas e práticas que dizem respeito a aspectos de circuitos elétricos, eletrônica e instrumentação. Na elaboração do *software* serão trabalhados conceitos de microcontroladores, programação e comunicação de dados. Essa abrangência exigirá um olhar tanto específico, quanto a análise, especificação e implementação de partes específicas do projeto acontecer, quanto um olhar holístico, que busca entender o projeto como a união de partes menores. Dessa forma, esses olhares contribuirão para um entendimento melhor das áreas necessárias à realização do projeto, sob os pontos de vista teórico e prático.

Além dos aspectos físicos de implementação citados no parágrafo anterior, também será trabalhada a visão do desenvolvimento de um produto tendo em mente um usuário final. Esse fato fará que questões de experiência de usuário sejam pensadas, pois além de um produto funcionar ele deve agradar quem o utiliza.

Ainda é interessante notar que o desenvolvimento da Régua Inteligente se enquadra em um tópico que tem sido cada vez mais discutido nos últimos anos, que é o *Internet of Things* (IoT). Existe uma grande busca pela ubiquidade das coisas, o mundo está cada vez mais conectado e, inevitavelmente, estará ainda mais nos próximos anos.

O conceito de *Big data* também está presente neste trabalho. Caso a Régua Inteligente seja utilizada por muitos usuários, um banco de dados com informações de tensão, corrente, potência, dentre outros valores de interesse será formado, e desse banco informações valiosas poderão ser retiradas e utilizadas por quem tem interesse.

Todos esses pontos citados são do meu interesse, e a implementação da Régua Inteligente possibilitará o trabalho deles.

## 1.4 Metodologia

Na primeira parte da execução do projeto acontecerá a revisão de trabalhos que são semelhantes a este. Serão abordados diferentes tipos de implementação de *smartplugs* com o intuito de estabelecer um breve histórico sobre eles, isso servirá como um direcionamento inicial para implementação deste projeto.

Em seguida será feito o levantamento de requisitos tendo em vista os possíveis usuários que utilizarão a Régua e o contexto em que esta utilização acontecerá. O próximo passo será pesquisar e especificar cada um dos componentes do projeto de acordo com os requisitos encontrados.

Com os componentes definidos, o desenvolvimento do *hardware* iniciará. No primeiro momento, a validação e verificação do funcionamento de cada um deles vão acontecer. Em seguida eles serão interligados um por um verificando se o funcionamento do conjunto resultante é o esperado, garantindo assim um *hardware* funcional.

Depois a implementação do *software* acontecerá, respeitando à ordem dos três últimos objetivos específicos da seção Objetivos deste capítulo. Para cada implementação finalizada, testes serão feitos com o intuito de verificar o código implementado, em busca de erros.

Quando as partes anteriores forem concluídas, testes de monitoramento de diferentes cargas serão feitos para verificar o funcionamento do projeto completo. Também serão realizados testes comparativos entre a Régua Inteligente construída e outros *smartplugs*, para verificar o desempenho da mesma.

## 1.5 Contribuições

As contribuições serão as seguintes:

- a) um levantamento de trabalhos e produtos comerciais disponíveis para medição de consumo de energia de eletrodomésticos;
- b) um levantamento de requisitos que permite identificar as principais necessidades dos usuários do sistema;
- c) o projeto e desenvolvimento do *hardware* e *firmware* de uma Régua Inteligente para a medição de consumo de energia elétrica;
- d) o desenvolvimento de um software servidor e de aplicação para coletar e monitorar os dados recolhidos pela régua.

## 1.6 Estrutura do texto

Este trabalho será dividido em sete capítulos, introdução, revisão da literatura, especificações do sistema, validação do sistema, materiais e métodos, resultados e conclusão e trabalhos futuros.

No capítulo de revisão da literatura serão discutidos conceitos preliminares que são necessários para que o entendimento deste trabalho seja possível. Serão apresentados fatores importantes para que uma medição seja realizada e também trabalhos relacionados que servirão para indicar diferentes tipos de implementações de *smart plug*.

O terceiro capítulo, especificações do sistema, primeiramente apresentará as estruturas do *hardware* e *software* que deverão compor a Régua Inteligente. Com base nessas estruturas esperadas, as especificações de cada uma das partes do *hardware* e *software* do projeto serão feitas.

No capítulo de validação do sistema serão abordadas as diferentes formas de validação do *hardware*, *software* e do funcionamento do conjunto dessas duas partes.

O quinto capítulo apresentará os materiais e ferramentas que foram escolhidos para a construção da Régua Intelige, com base nas especificações feitas no capítulo três. O resultado final da construção dessas duas partes também será indicado.

Os resultados das validações serão apresentados no capítulo seguinte. Serão mostrados os dados de medição de consumo de energia encontrados para diferentes tipos de carga e a qualidade dessas medições de acordo com as especificações de cada carga e de um *smart plug* referência.

No final será feita uma conclusão que resgatará o trabalho feito e seus resultados, assim como uma apresentação de tarefas que ainda devem ser efetuadas e possíveis melhorias.

## 2 Revisão da Literatura

Para que medir algo? O site do *Cambridge Dictionary* define medir como "*to discover the exact size, amount, etc., of something*". Como a definição anterior sugere, medimos algo para conhecer o tamanho ou a quantidade de alguma coisa, conhecer tais propriedades de algo pode nos ajudar em muitas coisas, por exemplo nas comparações, no processo de tomada de decisões e até no auxílio do entendimento de fenômenos naturais.

Quando Tales de Mileto percebeu a eletricidade estática há aproximadamente 600 a.C através do atrito de âmbar (em grego *elektron*) com peles de animais, ele provavelmente teve uma experiência de medição qualitativa da interação de força elétrica entre objetos, percebendo que após o atrito, ao aproximar o âmbar de objetos leves e secos os mesmos eram atraídos.

Um exemplo de medição quantitativa, talvez um dos mais célebres na história do Eletromagnetismo, foi feita por Charles Coulomb no final do século XVIII. Com sua balança de torção e com esferas carregadas eletricamente, ele deduziu sua famosa Lei de Coulomb, que afirma que "*a intensidade da força elétrica de interação entre cargas puntiformes é diretamente proporcional ao produto dos módulos de cada carga e inversamente proporcional ao quadrado da distância que as separa*".

As medições, qualitativas ou quantitativas, sempre foram feitas ao longo da história do homem. Atualmente, um dos fundamentos mais importantes na Teoria de Controle é o da realimentação, para ela existir é necessário um instrumento que faça a medição de uma variável de controle para que em seguida uma ação de controle seja tomada, garantindo um comportamento desejável para um sistema qualquer.

Dentro da perspectiva deste trabalho, a medição que a Régua Inteligente fará, possibilitará o acompanhamento do consumo de energia elétrica de um ou mais dispositivos de uma casa, e com as informações obtidas da medição o usuário da régua poderá tomar decisões que favoreçam a redução do consumo de eletricidade de sua casa.

### 2.1 Conceitos preliminares

Neste capítulo serão abordados conceitos teóricos importantes para o entendimento e elaboração do projeto da Régua Inteligente. Primeiramente serão apresentados alguns fatores importantes que devem ser conhecidos para que a realização de uma medição seja correta. Depois serão apresentados alguns trabalhos que estão relacionados a este projeto.

### 2.1.1 Fatores importantes de uma medição

Os fatores relevantes de medição que devem ser considerados para a construção da Régua Inteligente serão comentados no texto que se segue, uma lista dos mesmos está indicada abaixo:

- Exatidão e precisão;
- Erros de medição, sistemático e aleatório;
- Repetitividade;
- Faixa dinâmica de valores da medição;
- Resposta em frequência de um sistema;
- Frequência de amostragem e resolução.

### 2.1.2 Exatidão e precisão

A exatidão de uma medição expressa a capacidade de um sistema funcionar sem erros, tendo sempre um bom desempenho. Isso significa que o instrumento consegue medir valores que são muito próximos do valor verdadeiro do mensurando. Um sistema que sempre acerta é visto como um sistema de ótima exatidão.

No que diz respeito a precisão, ela indica o nível de dispersão entre diferentes medições. Quanto mais próximos são os valores de medições repetidas, sob condições especificadas, mais preciso é o sistema.

Os dois parâmetros são qualitativos e estão associados ao desempenho de um sistema. Quanto mais preciso é um sistema, maior é a capacidade dele de repetição, o que indica uma pequena dispersão. Se um sistema possui muita exatidão, isso significa que ele apresenta poucos erros. A Figura 1 exemplifica diferentes situações de desempenho de um sistema.

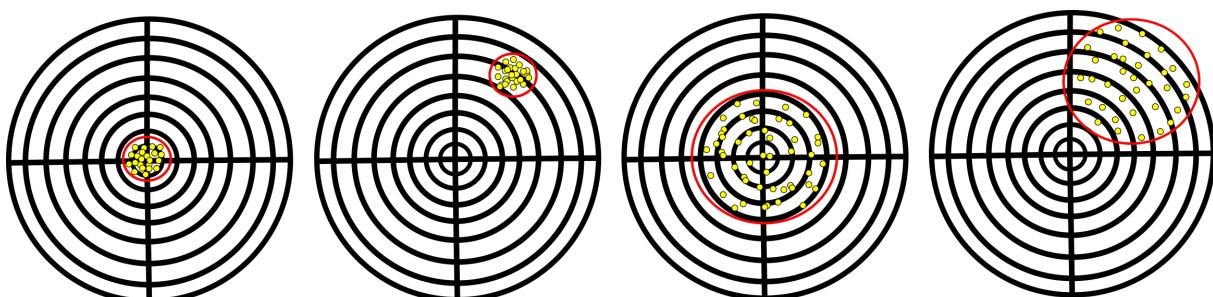


Figura 1 – Diferentes situações de desempenho de um sistema

O primeiro alvo da figura indica um sistema que tem alta exatidão e precisão. O segundo alvo mostra um sistema que tem pouca exatidão, uma vez que os valores estão longe do centro do alvo, apesar disso ele é preciso, pois a dispersão dos valores é pequena. O terceiro alvo exemplifica o oposto da situação do segundo alvo. Já o último indica um sistema com pouca exatidão e precisão.

### 2.1.3 Erros de medição, sistemático e aleatório

Falando agora sobre os diferentes tipos de erros, o primeiro é o erro de medição. Ele expressa a diferença entre o valor medido de uma grandeza e um valor determinado como referência (valor verdadeiro convencional). Faz sentido pensar em um valor referência porque se soubéssemos desde do princípio o valor que se quer medir, o processo de medição não faria sentido.

Analizando a definição do erro de medição, pode-se imaginar que se conhecido tal valor poderíamos encontrar o valor verdadeiro do mensurando. Isso não acontece pois na prática esse erro depende de vários fatores aleatórios, como condições climáticas, quem é o operador da medição, deterioração do instrumento, dentre outros. O erro de medição só poderia ser encontrado se o valor real do mensurando fosse conhecido, mas como discutimos no parágrafo anterior esse não é o caso.

A abordagem que se dá diante dessa situação é o desmembramento do erro de medição em duas componentes quantitativas, os erros sistemático e aleatório. O erro sistemático é aquele que após muitas medições repetidas permanece constante ou varia de maneira previsível. Ele está relacionado a uma polarização do resultado de medição em relação ao valor verdadeiro.

É possível estimar o erro sistemático, para isso devem ser feitas muitas medições, idealmente infinitas, de um mensurando cujo valor verdadeiro é bem conhecido. A Equação 2.1 indica tal estimativa:

$$E_s = \bar{I}_\infty - V_v \quad (2.1)$$

Na equação acima  $E_s$  é o erro sistemático,  $\bar{I}_\infty$  é a média de um número infinito de indicações e  $V_v$  é o valor verdadeiro do mensurando. Como na prática um número infinito de indicações é impossível, e o valor verdadeiro não é conhecido, o que se define é uma tendência do erro sistemático, definida pela Equação 2.2 abaixo:

$$T_d = \bar{I} - V_{vc} \quad (2.2)$$

Na equação acima  $T_d$  é a tendência do erro sistemático,  $\bar{I}$  é a média de um número finito de indicações e  $V_{vc}$  é o valor verdadeiro convencional ou de referência do mensurando.

Para corrigir o erro sistemático na medição basta somar  $-T_d$  a  $\bar{I}$ .

Após a compensação do erro sistemático na medição, em média as indicações são corretas. Mas elas ainda se diferem e isso acontece devido aos aspectos aleatórios da medição. O erro aleatório pode ser definido pela seguinte equação:

$$Ea_i = I_i - \bar{I} \quad (2.3)$$

Na Equação 2.3,  $Ea_i$  é o erro aleatório da i-ésima indicação,  $I_i$  a i-ésima indicação e  $\bar{I}$  a média das indicações.

Na Figura 1 está indicado um círculo vermelho que engloba os valores de medição. Quanto maior é o raio deste círculo, maior será o erro aleatório. A distância desse círculo do centro do alvo indica o tamanho do erro sistemático, quanto maior for a distância maior será o erro sistemático.

#### 2.1.4 Repetitividade

Outro fator importante no processo de medição é a repetitividade. Ela é definida como a faixa de valores dentro da qual se espera encontrar o erro aleatório de um sistema de medição. Quando a repetitividade é boa, isso indica que um instrumento tem a capacidade de fornecer indicações muito próximas, sob as mesmas condições de medição.

Ainda analisando a Figura 1, quanto maior é o raio do círculo vermelho, pior será a repetitividade.

#### 2.1.5 Faixa dinâmica de valores da medição

Um fator importante que ajudará no dimensionamento dos circuitos da Régua Inteligente é a faixa dinâmica de valores da medição. A intenção deste projeto é de medir o consumo de energia de eletrodomésticos, dentro deste contexto as variáveis relevantes para o cálculo do consumo são corrente e tensão. No Brasil, os valores de tensão fornecidos pela rede para casas podem ser de  $127V_{ef}$  e  $220V_{ef}$ , com flutuações de  $\pm 2\%$ . Os valores de corrente vão depender da potência de cada dispositivo, mas sabemos que no Brasil as tomadas disponíveis suportam valores máximos de corrente de  $10A$  ou  $20A$ . A partir dessas informações é razoável inferir que a Régua Inteligente deverá suportar níveis de tensão entre  $\pm 220V_{ef}$  e níveis de corrente entre  $\pm 20A$ . Com esses intervalos conhecidos será possível dimensionar os circuitos da Régua para que os níveis de potência, corrente e tensão estejam dentro dos limites de funcionamento de cada um deles.

### 2.1.6 Resposta em frequência de um sistema

Será discutido agora a resposta em frequência de um sistema sob o ponto de vista de um sensor, um fator que é importante para nos indicar se o sinal medido representa de fato o sinal real. Sabe-se que as tensões disponíveis nas casas tem o formato bem próximo de uma senoide com frequência de  $60\text{Hz}$ , descontadas as flutuações de tensão e frequência que podem acontecer na rede. Porém, as correntes podem ter os formatos mais distintos e aleatórios, dependendo do tipo e estado de funcionamento do eletrodoméstico. Uma preocupação que surge deste fato é de como é o espectro de frequências do sinal de corrente que se deseja medir. Caso o sinal de corrente possua componentes de alta frequência, com amplitudes também altas, serão necessários sensores que possuam uma banda de passagem tão alta quanto as maiores componentes de frequência. Isso é necessário para que o sinal medido pelo sensor não possua distorção na sua forma em relação a forma real do mensurando.

Esse é um aspecto importante e que deve ser levado em consideração, mas para definir uma ação diante desta questão é necessário a reposta de uma pergunta, qual a finalidade da medição? No caso deste trabalho, o que se deseja medir é o consumo médio de energia elétrica de um eletrodoméstico. Neste caso é possível utilizar a definição de potência média para sinais discretos, para um intervalo de  $N$  amostras ela é definida abaixo:

$$P = \frac{1}{N} \sum_{n=0}^{N-1} v[n]i[n] \quad (2.4)$$

Uma vez calculada a potência média, o cálculo da energia do sinal para um intervalo de  $N$  amostras será:

$$E = PN \quad (2.5)$$

O valor eficaz de um sinal discreto  $u[n]$  pode ser encontrado a partir da seguinte equação:

$$U_{rms} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} u[n]^2} \quad (2.6)$$

Seja  $V_{rms}$  e  $I_{rms}$  os valores eficazes de tensão e corrente, a potência aparente será:

$$S = V_{rms}I_{rms} \quad (2.7)$$

A potência reativa pode ser expressa por:

$$Q = \sqrt{S^2 - P^2} \quad (2.8)$$

O fator de potência será:

$$F.P. = \frac{P}{S} \quad (2.9)$$

Essas equações serão implementadas no microcontrolador que recebe os sinais de tensão e corrente discretizados do conversor A/D.

### 2.1.7 Taxa de amostragem e resolução

Outros dois conceitos importantes são os de taxa de amostragem e resolução, fatores relacionados ao conversor A/D.

O teorema de Nyquist-Shannon afirma que a frequência de amostragem de um sinal deve ser maior que o dobro da maior frequência contida no sinal a ser amostrado, para que possa ser reproduzido integralmente sem erro de *aliasing*. Isso é importante para este projeto pois queremos que os sinais de tensão e corrente medidos na frequência de 60Hz sejam representados fielmente no tempo discreto, para que os cálculos que serão feitos no programa do microcontrolador sejam corretos.

A resolução de um conversor A/D irá determinar qual o menor valor que pode ser convertido dentro da faixa dinâmica de valores esperado do sinal amostrado. Ela é especificada pelo número de bits do conversor. Por exemplo, quando um conversor possui uma resolução de 12 bits significa que o conversor consegue diferenciar sinais com amplitude  $\frac{1}{2^{12}}$  do valor total da faixa dinâmica. Para uma faixa dinâmica de 5V, a menor amplitude que pode ser resolvida é  $\frac{5V}{12} = 1,22mV$ .

## 2.2 Trabalhos relacionados

Nesta seção serão apresentados outros trabalhos mencionados em artigos que possuem a finalidade parecida com o que se deseja projetar neste trabalho. É interessante indicar que foi fácil encontrar artigos recentes, nos últimos cinco anos, que tratam do tema da medição de energia elétrica em casas, o que é uma evidência da busca pela IoT.

No artigo de Han et al. (2014) é apresentada uma arquitetura completa de monitoramento de consumo e geração de energia de uma casa. Sob o ponto de vista do monitoramento de consumo de energia, é proposta a utilização de dispositivos de medição que possibilitam a comunicação via uma rede ZigBee, uma tecnologia de baixo consumo de energia mas baixa área de alcance de sinal, com um servidor local da casa. Esse servidor local possui conexão com a internet e permite que informações de consumo possam ser

acessadas local ou remotamente. Sob o ponto de vista de geração de energia, é proposta uma arquitetura de painéis solares, geradores eólicos, inversores que comunicam com o servidor local da casa através de modems *Power Line Communication* (PLC) e *Renewable Energy Gateway* (REG). Com os dados recebidos, o servidor local consegue gerar diversas informações que podem ser utilizadas para entender o perfil de consumo e geração de energia da casa. Os autores ainda propõem uma servidor externo de gestão de energia que pode comunicar com vários servidores locais de casas através da internet e gerar um perfil de consumo e geração de energia de regiões maiores. A arquitetura apresentada é abrange muitos aspectos do monitoramento e geração de energia de forma inteligente, como a IoT sugere. Mas a solução requer a utilização de diversos dispositivos e isso pode fazer com que o preço do projeto seja muito elevado.

No trabalho de [Didhe, Kulkarni e Bakal \(2015\)](#) é proposta uma solução de monitoramento e controle (liga/desliga) de eletrodomésticos para uma grande área de alcance de sinal, como uma casa ou restaurante. Ela é baseada em uma estrutura com uma aplicação Android em um *smartphone* como a interface visual de monitoramento e controle, um modem *General Packet Radio Service* (GPRS) como a interface de comunicação entre o *smartphone* e o dispositivo que faz a medição, um microcontrolador que tem duas funções, uma é receber as informações dos sensores, que fazem as medições, e processa os dados e encaminha para o modem GPRS, a outra é acionar *relays* a partir de comandos recebidos do *smartphone* para fazer o comando de liga/desliga. Uma vantagem desta configuração é que o modem GPRS possui uma área de alcance de sinal muito grande, por outro lado o consumo de energia da configuração pode ser elevado se comparado a outras soluções, além do preço do modem GPRS ser elevado comparado a outros módulos de comunicação sem fio. Ainda existe o problema do acesso remoto do dispositivo, que não é possível.

No artigo apresentado por [Oladimeji et al. \(2017\)](#) é discutida uma solução de automação de uma casa utilizando as tecnologias de *Global System for Mobile communications* (GSM) e Bluetooth. A ideia é possibilitar que o monitoramento e controle de um *smartplug* possa ser feita dentro de dois cenários, o primeiro é quando o usuário está próximo do dispositivo, neste caso a comunicação entre o celular do usuário, que possuí uma interface desenvolvida em Android, será feita através da tecnologia Bluetooth. No segundo caso o usuário estará longe do dispositivo e poderá controlar o mesmo através de comandos especificados em mensagens *Short Message Service* (SMS). Para que essas comunicações sejam possíveis dois módulos de *hardware* são utilizados, um Bluetooth e outro GSM que possuí um *Subscriber Identity Module* (SIM) para que as mensagens SMS possam ser trocadas entre o *smartplug* e celular. O aspecto híbrido dessa implementação é muito interessante, pois possibilita tanto a comunicação local quanto remota, além de nos fazer refletir sobre a possibilidade de *hardwares* híbridos que atendem a diferentes cenários de utilização de dispositivos. Por outro lado a implementação do *hardware* e *software* serão mais complexos, uma vez que eles devem funcionar de acordo com duas tecnologias

diferentes. Outro aspecto que deve ser levado em consideração é sobre a forma como a interação remota acontece, mensagens SMS oferecem uma interface para o controle e monitoramento do dispositivo bem limitada, pois a interação entre o usuário e o dispositivo apenas acontecerá através de textos. Com as tecnologias atuais para a construção de interfaces de *front-end* para Web e Mobile é possível atingir resultados bem melhores do que uma interação apenas textual.

No trabalho de [Musleh, Debouza e Farook \(2017\)](#) é utilizada uma estrutura de comunicação mestre e escravo para atingir o objetivo de monitoramento de energia elétrica. As estruturas escravo serão os *smartplugs* que estarão conectados com os dispositivos elétricos da casa, cada um deles possuí em seu *hardware* sensores de corrente e tensão, um microcontrolador e um módulo que possibilita a conectividade através da tecnologia ZigBee. Eles estão configurados para funcionar em uma rede *mesh* juntamente com a unidade mestre, que é composta por um Raspberry Pi e um módulo ZigBee para a comunicação com os *smartplugs*. O Raspberry Pi possui conectividade Wi-Fi e por ela permite a comunicação remota entre o módulo mestre e uma aplicação desenvolvida na plataforma *Windows Phone*. A solução proposta é interessante pois a tecnologia ZigBee proporciona um consumo baixo de energia dos *smartplugs* e a configuração em rede *mesh* desses dispositivos melhora o alcance do sinal de cada dispositivo. Além disso existe a possibilidade de comunicação remota uma vez que é utilizado um Raspberry Pi que possui conectividade Wi-Fi. Porém a necessidade do Raspberry Pi para possibilitar a comunicação Wi-Fi pode ser um problema se a intenção é desenvolver um produto para grande produção e venda, uma vez que ele encarece o preço do produto.

O texto de [S e Mahadevaswamy \(2018\)](#) apresenta uma solução para monitoramento e controle de eletrodomésticos com a utilização de um conjunto de *smartplugs* conectados via *Bluetooth Low Energy* (BLE) a uma aplicação Android executada em um celular. Foi desenvolvido um *hardware* capaz de medir tensão, corrente, potência, fator de potência e frequência. Ele consiste na utilização de uma fonte de tensão, um microcontrolador, sensores de corrente e tensão e um módulo BLE que foram conectados através de uma *Printed Circuit Board* (PCB). A implementação do *software* foi feita através do *MIT app inventor*, através dele a programação é feita pela ligação de blocos lógicos, o que facilita a tarefa de programação da interface gráfica e de comunicação com o *hardware*. Para testar o desempenho do *smartplug*, ele foi conectado a diferentes tipos de cargas para verificar a qualidade do monitoramento e controle, que aconteceram com sucesso. Tal sucesso é apresentado através de comparações entre os valores de consumo definidos nos manuais das cargas e os valores medidos pelo dispositivo. A utilização do BLE demonstra uma grande vantagem em relação a outras soluções quando a intenção é o consumo reduzido de energia para a operação do *smartplug*, mas uma limitação da utilização de tal tecnologia decorre quando se deseja monitorar ou controlar de forma remota o *smartplug*, porque ela necessita que o celular esteja próximo do *smartplug* para que a comunicação de dados seja

possível.

Analizando de uma forma holística as propostas de monitoramento e controle de energia elétrica discutidas nos parágrafos acima percebemos que existem vantagens e desvantagens na utilização de cada uma das soluções. Em alguns projetos é possível criar *smartplugs* que possuem um baixo consumo de energia porém não possibilitam comunicação remota. Em outros é possível um alcance maior do sinal de comunicação mas o preço da solução e o consumo de energia podem ser mais elevados. Ainda existem soluções que possibilitam a comunicação remota mas a interface de monitoramento e controle não é tão intuitiva. O interessante da análise dos artigos é o estudo das diferentes formas de solução para o problema de monitoramento e controle de energia elétrica com o intuito de direcionar este trabalho a atingir o seu objetivo da melhor maneira possível, o que foi muito útil.

Este trabalho apresentará uma solução que permitirá o monitoramento remoto do consumo de energia elétrica através da utilização de módulos eletrônicos populares no mercado, e da utilização de ferramentas de software atuais e poderosas que possibilitarão a construção de um servidor, interfaces de comunicação e interface visual robustos. Além disso, será feita uma tentativa de medir outras variáveis elétricas importantes como, potência aparente, potência reativa, fator de potência, tensão eficaz, corrente eficaz e valores máximos e mínimos de tensão e corrente. Esses fatos fazem que este trabalho possa contribuir em diferentes aspectos dos trabalhos aqui discutidos.



# 3 Especificações do sistema

Neste capítulo serão abordados os tipos de componentes e ferramentas que deverão compor o *hardware* e *software* do projeto, sob um ponto de vista mais teórico. Isso será feito para criar uma ideia inicial do projeto. Em seguida serão feitas as especificações do *hardware* e do *software*.

As especificações podem ser feitas sob o ponto de vista de dois contextos. O primeiro diz respeito a aspectos físicos do projeto, como especificações de potência, corrente, tensão, capacidade de processamento de dados, dentre outros. O segundo trata dos aspectos de mercado, como necessidades do cliente, custo de produção, custo de manutenção, receita, dentre outros.

Apesar de existir a intenção de lançar o produto, os aspectos que envolvem a análise do mercado não serão feitos neste trabalho. O objetivo maior deste TCC é de trabalhar os conceitos de *hardware* e *software* que estão envolvidos no desenvolvimento da Régua Inteligente, assim como a sua implementação física. A análise de mercado poderá ser feita em um segundo momento.

Sendo assim eu parto da hipótese de mercado, que não será validada, de que as pessoas estão cada vez mais conscientes sobre as questões de consumo energético e que a Régua Inteligente fornecerá uma possibilidade de monitoramento e controle maior do consumo de energia de suas casas, logo o produto será aceito no mercado.

## 3.1 Componentes do *hardware*

O digrama de módulos indicado na Figura 2 mostra os módulos que serão necessários para a construção do *hardware* da Régua Inteligente. Nesta seção serão discutidos cada um dos módulos, tratando a função de cada um deles.

### 3.1.1 Fonte de tensão e conversor de tensão DC/DC

O primeiro conjunto de módulos da Figura 2 mostra os módulos de fonte de tensão e conversor de tensão DC/DC. Eles terão o papel de alimentação dos módulos da régua.

A fonte de tensão será uma fonte chaveada que fornecerá níveis de tensão e corrente adequados a alimentação dos módulos da régua. Será necessário também um conversor de tensão DC/DC uma vez que os módulos vão exigir diferentes níveis de tensão de alimentação.

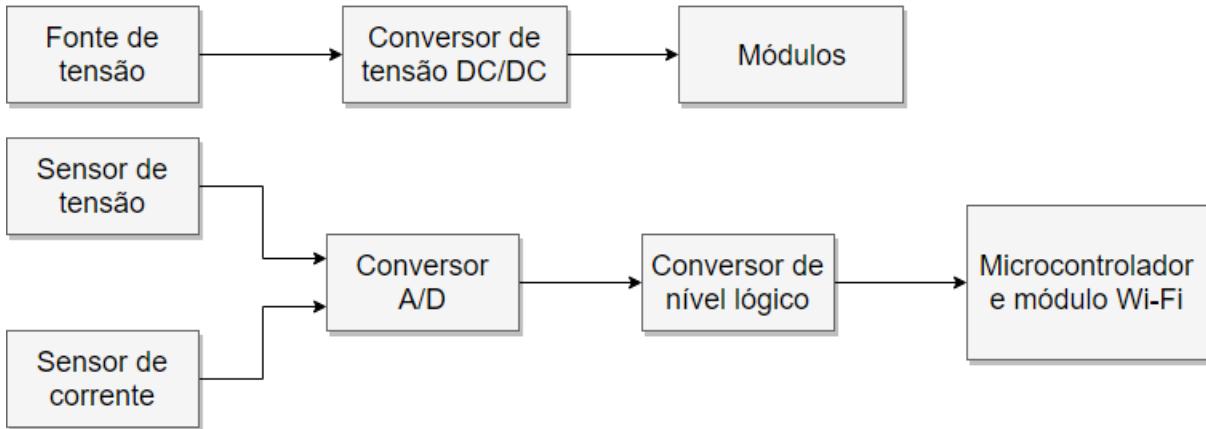


Figura 2 – Diagrama de módulos da Régua Inteligente

### 3.1.2 Sensores de tensão e corrente

Os sensores serão responsáveis em medir os sinais de tensão e corrente que serão entradas para o conversor A/D. A Figura 3 indica qual é o comportamento esperado dos sensores.

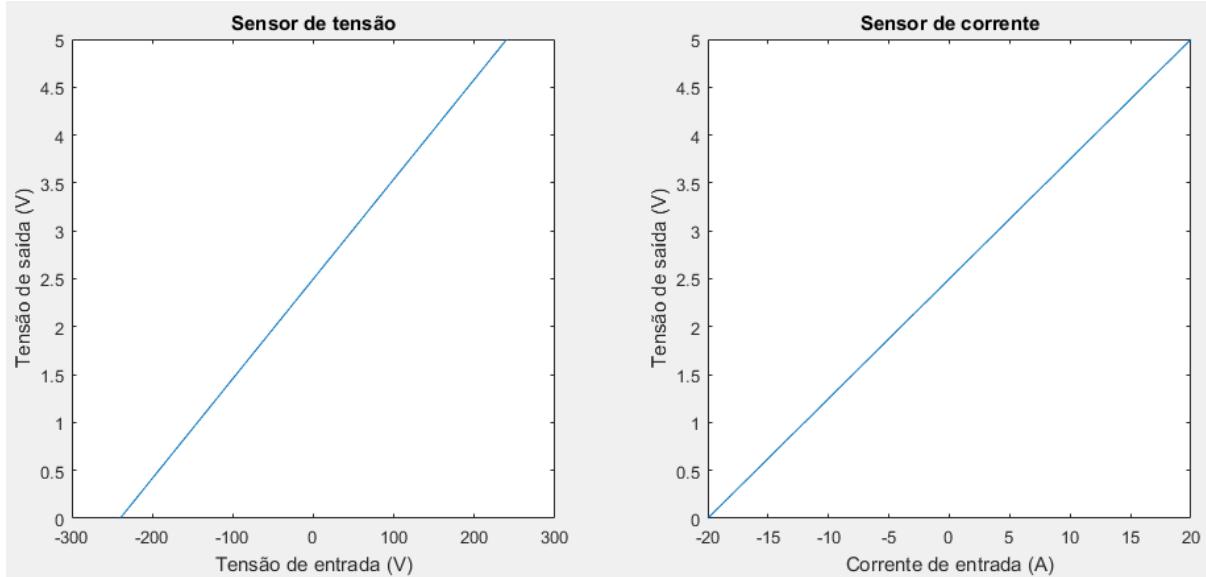


Figura 3 – Exemplo de resposta esperada dos sensores de tensão e corrente

Na figura acima percebemos que a tensão de entrada varia entre  $\pm 240V$  e a corrente de entrada varia entre  $\pm 20A$ . A resposta para os dois casos é linear e o intervalo de valores da saída estão entre 0 e 5V. Esse é o comportamento esperado pois os níveis de tensão de operação das entradas dos conversores A/D normalmente estão entre 0 e 5V.

Outro fator importante desses sensores é a resposta em frequência. De acordo com o que foi discutido na seção de resposta em frequência de um sistema, espera-se que os

sensores tenham um perfil de filtro do tipo passa-baixas ou passa-faixa, uma vez que as componentes de  $60Hz$  da tensão e corrente são as relevantes para o cálculo da potência média.

### 3.1.3 Conversor A/D

Sobre o conversor A/D é importante falar sobre sua taxa de amostragem e resolução.

Como a componente de  $60Hz$  dos sinais de tensão e corrente é a principal para o cálculo da potência média, o conversor A/D deve ter uma frequência de amostragem de no mínimo duas vezes  $60Hz$ . Porém é uma prática comum em projetos práticos aumentar a margem da frequência de amostragem para no mínimo dez vezes o valor da maior frequência que se deseja medir, o que daria algo em torno de  $600Hz$ .

Pensando agora sobre a resolução do conversor, caso ele tenha uma resolução de 12 bits e a faixa dinâmica esperada dos valores que ele irá converter seja de 0 a  $5V$ , encontramos na seção de taxa de amostragem e resolução que a menor amplitude que poderá ser resolvida é de  $1,22mV$ .

Outra informação importante sobre o conversor A/D é que ele deve ser capaz de converter quatro entradas analógicas e enviar as informações discretizadas para o microcontrolador através de uma comunicação serial.

### 3.1.4 Conversor de nível lógico

O conversor de nível lógico irá receber o sinal do conversor A/D e converter o nível de tensão para um valor que o microcontrolador opera para sua comunicação serial.

### 3.1.5 Módulo Wi-Fi

O módulo Wi-Fi deverá possuir um microcontrolador que terá uma interface de comunicação serial para a comunicação com o conversor A/D e irá disponibilizar uma quantidade de memória mínima para a gravação de um programa que possibilitará o cálculo de potência e energia média e o envio de informações via Wi-Fi. O módulo deverá possuir um *firmware* que facilitará a comunicação Wi-Fi, dando suporte às interações TCP/IP.

## 3.2 Softwares da aplicação e do servidor

A Figura 4 indica os dispositivos que farão parte do sistema de comunicação do projeto que possibilitará a troca de informações entre eles. Nesta seção serão discutidos os papéis de cada um desses dispositivos e das interfaces de comunicação entre eles.

Os *softwares* da aplicação serão implementados na Régua Inteligente e na Interface de apresentação ao usuário através de um navegador, já os *softwares* do servidor serão implementados no *Web Server* e *Database Server*.

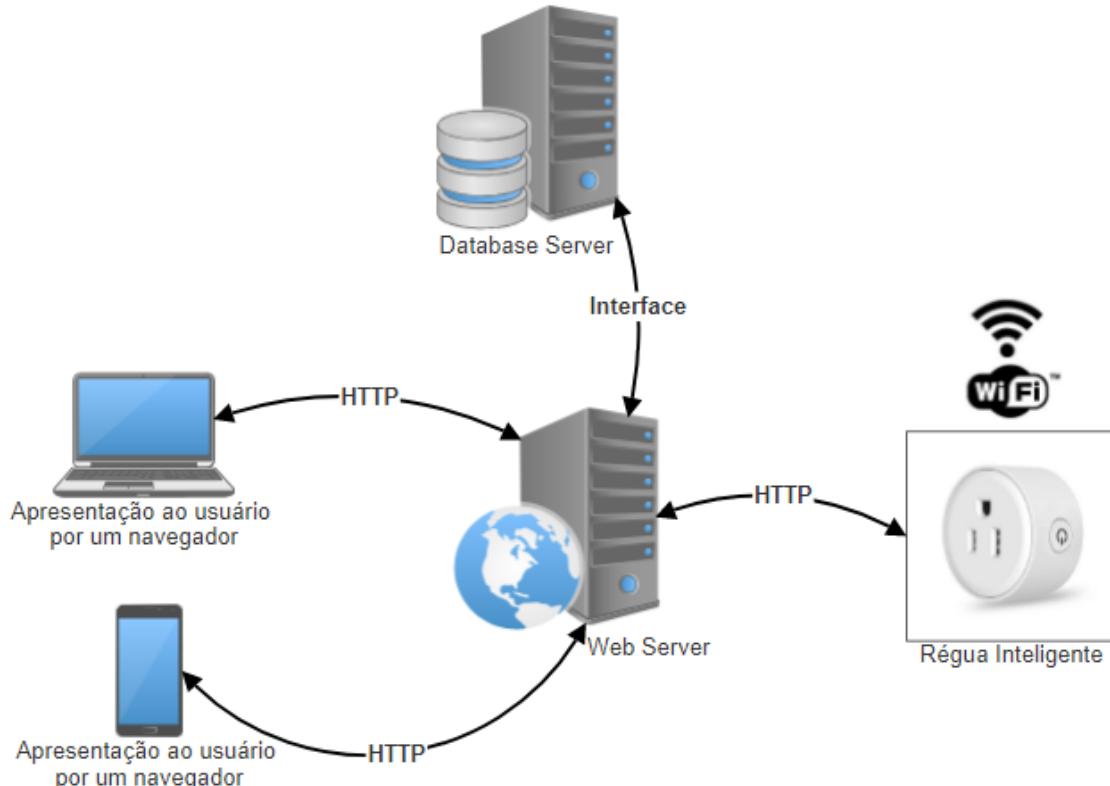


Figura 4 – Diagrama de rede do projeto da Régua Inteligente

### 3.2.1 Web server

Antes de falar sobre a implementação do *Web Server* vale a pena notar com a análise da Figura 4 que as interfaces de comunicação entre a Régua Inteligente e a Interface do navegador com o *Web Server* serão feitas através do protocolo de comunicação *Hypertext Transfer Protocol* (HTTP). Esse é um protocolo de comunicação que pertence a camada de aplicação do modelo *Open System Interconnection* (OSI) e é um dos mais utilizados na internet para o tráfego de dados. Existem diferentes tipos de servidores e aplicações que implementam os padrões do protocolo HTTP e possibilitam que requisições sejam enviadas e recebidas.

O protocolo HTTP define oito métodos que possuem funções diferentes dependendo do tipo de requisição a ser feita. Para este projeto os mais relevantes são os métodos GET, POST, PUT e DELETE. O método GET é utilizado para requisições que requerem algum tipo de dado de um servidor, como dados armazenados em um *Web Server*. O método POST indica uma requisição que irá enviar dados a serem processados, como por

exemplo uma requisição que requer o armazenamento de um novo dado em um *Database Server*. O método PUT é similar ao POST, mas indica uma atualização de um dado já existente. A função do DELETE é a exclusão de algum dado. Existem outros protocolos de comunicação na camada de aplicação como *File Transfer Protocol* (FTP) e *Simple Mail Transfer Protocol* (SMTP). Para mais informações sobre os protocolos aqui citados, o modelo OSI e outros conceitos de rede de computadores que serão citados neste trabalho, o livro do [Tanenbaum \(2011\)](#) é recomendado.

Falando agora sobre a implementação do *Web Server*, como foi dito anteriormente existem diferentes implementações de servidores que dão suporte à comunicação HTTP. Exemplos comuns são Apache, Nginx e *Internet Information Services* (IIS). Cada um desses servidores diferem na sua forma de configuração, no desempenho de execução e no suporte de diferentes linguagens e interfaces ao desenvolvimento de aplicações. Cabe ao projetista analisar as vantagens e desvantagens da implementação de cada servidor HTTP e fazer a escolha que trará mais benefícios à implementação do projeto desejado.

Os servidores Web e de banco de dados estarão hospedados em um mesmo servidor. Esse servidor terá um endereço *Internet Protocol* (IP) público que será traduzido por um servidor *Domain Name System* (DNS). As requisições HTTP serão redirecionadas pelo servidor para um endereço IP local onde o *Web Server* estará hospedado.

Vale ainda ressaltar que além de hospedar o site que irá ser entregue ao usuários finais do produto, o *Web Server* possuirá uma aplicação que será responsável pelo tratamento das requisições HTTP e também pela interface de comunicação com o banco de dados. Essa aplicação pode ser implementada em diferentes linguagens, como Python, PHP e C#.

### 3.2.2 Database server

É muito comum que exista um servidor de banco de dados para que informações importantes do projeto possam ser armazenadas dentro de uma estrutura robusta. Uma categorização mais ampla classifica os bancos de dados em relacionais e não relacionais.

Um banco de dado relacional organiza a sua estrutura de dados em um conjunto de tabelas, com uma chave única identificando cada um dos dados dessas tabelas. Elas podem ser relacionadas através de chaves, por isso o nome banco de dados relacional. Para a manipulação dessas tabelas existem os *Relational Database Management System* (RDBMS), que utilizam variações da linguagem *Structured Query Language* (SQL). Exemplos de RDBMS são Oracle, MySQL e Microsoft SQL Server.

Os bancos de dados não relacionais estruturam seus dados de formas diferentes, como *Key-values*, *Column*, *Document* e *Graph databases*. Assim como a estruturação dos dados difere, a forma como os dados são manipulados também será diferente. Devido as

diferenças de estruturação e manipulação de dados, o desempenho dos banco de dados não relacionais podem superar o desempenho dos relacionais, por terem maior flexibilidade de estruturação de dados e da facilitação de um crescimento em maior velocidade dos mesmos. Exemplos de banco de dados não relacionais são Redis, HBase, MongoDB e Neo4J.

O servidor de banco de dados estará hospedado no mesmo servidor que o *Web Server* está presente, com um endereço IP local diferente.

### 3.2.3 Comunicação

A lógica de comunicação HTTP da Régua Inteligente com o *Web Server* será feita através de bibliotecas voltadas à estrutura de programação do microcontrolador que facilitam a comunicação Wi-Fi. A Régua Inteligente de um usuário fará requisições POST para o registro da Régua e o envio das informações obtidas por ela para o *Web Server* para que posteriormente possam ser armazenadas no banco de dados.

Em vez da comunicação direta via Wi-Fi da Régua Inteligente com o *Web Server*, outros protocolos intermediários poderiam ser utilizados, como *Bluetooth Low Energy*, *Bluetooth* e *ZigBee*. Cabe ao projetista identificar as vantagens e desvantagens da utilização de cada um para que os objetivos desejados sejam alcançados. Na seção de trabalhos relacionados uma discussão maior de cada um desses protocolos será feita.

### 3.2.4 Interface do navegador

A implementação da interface do navegador para a exibição dos dados medidos pela Régua Inteligente pode ser feita somente com as linguagens de programação Web *Hypertext Markup Language* (HTML), *Cascading Style Sheets* (CSS) e JavaScript, ou com auxílio de *frameworks* ou bibliotecas desenvolvidas em JavaScript que ajudam no desenvolvimento de páginas mais dinâmicas. Exemplos de *frameworks* são Angular e Vue.js, um exemplo de biblioteca é o React. Cada uma dessas ferramentas oferece diferentes recursos para o desenvolvimento Web de *front-end*, o estudo de cada uma das ferramentas irá indicar a melhor escolha para o tipo de projeto desejado.

Independentemente da escolha do *framework* ou biblioteca, fato é que a interface final será armazenada no *Web Server* e poderá ser obtida por qualquer usuário que faça uma requisição para esse servidor. A aplicação de *front-end* fará requisições de tempo em tempo para o *Web Server* para a coleta dos dados medidos pela Régua Inteligente e armazenados no banco de dados, indicando tais valores na tela do navegador.

### 3.3 Especificações do hardware

As principais especificações do *hardware* necessárias à Régua Inteligente são as faixas de valores de medição de tensão e corrente dos sensores.

Os outros são níveis de tensão e corrente máximos suportados pelos sensores, resposta em frequência dos sensores, tensões de comunicação entre os sensores e o conversor A/D, tensões e tipo de comunicação entre o conversor A/D e microcontrolador, resolução, taxa de amostragem e quantidade de canais do conversor A/D, microcontrolador com suporte Wi-Fi e consumo de energia dos módulos.

#### 3.3.1 Faixa de medição de tensão

De acordo com a referência [ANEEL \(2018\)](#) as tensões mais comuns nas residências brasileiras são senoides de  $127V_{ef}$  e  $220V_{ef}$  na frequência de  $60Hz$ . Logo, a especificação para o sensor de tensão da Régua Inteligente é que ele consiga medir tensões que estão na faixa de  $\pm 220V_{ef}$  ou mais. Dessa forma a Régua Inteligente poderá ser utilizada pela maior quantidade de residências brasileiras.

#### 3.3.2 Faixa de medição de corrente

A potência elétrica aparente de um dispositivo pode ser definida pela seguinte equação:

$$S = VI \quad (3.1)$$

Na equação 3.1,  $S$  é a potência aparente do dispositivo,  $V$  a tensão eficaz aplicada ao dispositivo e  $I$  a corrente eficaz que circula pelo dispositivo. Essa equação indica que a corrente dependerá dos níveis de tensão presentes nas casas,  $127V_{ef}$  e  $220V_{ef}$ , e também da potência consumida pelo dispositivo.

As tensões esperadas já estão especificadas, resta descobrir quais são as potências dos eletrodomésticos mais comuns para que uma estimativa da faixa de corrente que o sensor deverá medir.

Para descobrir esses níveis de potência uma pesquisa de dados foi feita, os resultados estão exibidos nas tabelas do link [Tabelas... \(2018b\)](#) do repositório do projeto que criei no GitHub. As informações dessas tabelas foram retiradas de órgãos ligados à distribuição ou qualidade de energia, caso uma análise mais detalhada das informações seja desejada as fontes apresentadas podem ser consultadas. As tabelas indicam os níveis de potência real ou aparante dos eletrodomésticos mais comuns.

A Tabela 1 foi montada a partir das informações contidas nas tabelas encontradas no link [Tabelas... \(2018b\)](#) e mostra dados que são relevantes para a definição da faixa de valores de corrente que o sensor deverá medir. Nela,  $P_{av}$  indica a média das potências dos eletrodomésticos de umas das oito tabelas.  $I_{av}$  segue a mesma lógica.

Tabela 1 – Resumo das tabelas do link [Tabelas... \(2018b\)](#)

Tabela	$P_{av}$ (W)	$P_{av}$ dos os aparelhos com potência $\leq 4400W$ (W)	Potência Máxima (W)	Potência Mínima (W)	$I_{av}$ para tensão de 127 Vrms (A)	$I_{av}$ para tensão de 220 Vrms (A)
1	2531	718	52200	10	6	3
2	1079	1079	4400	40	9	5
3	1418	798	12100	10	6	4
4	1606	1023	7500	50	8	5
5	2243	737	38000	5	6	3
6	592	487	5500	3	4	2
7	741	602	6600	9	5	3
8	522	447	4500	6	4	2

A segunda coluna da Tabela 1 foi montada levando em consideração todos os eletrodomésticos de cada uma das tabelas. A terceira coluna levou em consideração apenas aparelhos com uma potência menor do que 4400W. Essa diferenciação foi feita pois alguns itens da tabela possuem dispositivos elétricos que são voltados para pequenas indústrias e que por essa razão possuem potência muito elevada, nas tabelas a potência desses itens estão indicados em amarelo para diferenciar dos eletrodomésticos mais usais encontrados em residências. As outras duas colunas que seguem representam as maiores e menores potências encontradas em cada uma das tabelas. Note que algumas potências máximas foram muito elevadas pois indicam um dispositivo elétrico utilizado em pequenas indústrias. As últimas duas colunas da tabela indicam as correntes médias levando em consideração a potência média de aparelhos com potência menor do que 4400W, para tensões de 127V<sub>ef</sub> e 220V<sub>ef</sub>.

As colunas das correntes médias sugerem que o sensor de corrente seja capaz de medir correntes na faixa de  $\pm 10A$ . Porém devemos lembrar que algumas potências podem ser maiores do que as especificadas na terceira coluna, como é o caso de um chuveiro elétrico de 4400W que irá consumir 20A de corrente em uma tensão de 220V<sub>ef</sub>.

Tendo essas questões em mente, a especificação do sensor de corrente é que ele seja capaz de medir valores de corrente na faixa de  $\pm 15A$  ou mais, que é uma margem razoável quando levamos em consideração os valores da Tabela 1. Ainda vale lembrar que o valor máximo de corrente mais usual nas tomadas das residências no brasil é de 20A, um sensor que seja capaz de medir um valor de corrente maior do que esse seria desnecessário para o objetivo deste trabalho.

### 3.3.3 Níveis de tensão e corrente máximos suportados pelos sensores

Foi especificado na parte anterior que as faixas de tensão e corrente que os sensores terão de medir devem estar entre  $\pm 220V_{ef}$  e  $\pm 15A$ , respectivamente. Analisando esses valores é possível afirmar que os máximos níveis de tensão e corrente suportados pelos sensores podem ser definidos como  $\pm 240V_{ef}$  e  $\pm 20A$ , essa é uma margem razoável que permitirá que os sensores operem com folga para medir as faixas requeridas. Respeitados os níveis máximos de medição de tensão e corrente dos sensores as potências máximas suportadas por eles também serão respeitadas.

### 3.3.4 Resposta em frequência dos sensores

Como foi discutido na seção de resposta em frequência de um sistema no capítulo de revisão da literatura, o objetivo da medição de corrente e tensão deste projeto é para que o cálculo da potência média e energia média sejam feitos, possibilitando ao usuário da Régua Inteligente obter uma estimativa do consumo de energia de um eletrodoméstico de sua casa. Tendo isso em mente, a resposta em frequência dos sensores não precisa ter uma banda de passagem elevada, pois apenas a componente média dos sinais é relevante para o cálculo da potência média e energia média. Caso o cálculo da potência aparante, da potência reativa e o fator de potência do dispositivo fossem desejados, aí sim seriam necessários sensores com banda de passagem maiores, pois o cálculo de tais grandezas exigem tal fato.

De qualquer maneira, os sensores com as maiores bandas de passagem que respeitam as outras especificações do projeto serão escolhidos. Isso permitirá que no futuro um possível cálculo das grandezas mencionadas no parágrafo anterior seja feito, verificando se os resultados encontrados são satisfatórios quando comparados com sensores mais precisos.

### 3.3.5 Tensões de comunicação entre os sensores e o conversor A/D

Os valores medidos pelos sensores deverão ser lidos por um conversor A/D. É extremamente comum encontrar conversores A/D no mercado que operam com níveis de tensão entre 0 a 5V ou 0 a 3,3V. Dessa forma os sensores de tensão e corrente deverão ter saídas proporcionais as entradas que estejam nessas faixas.

### 3.3.6 Tensões e tipo de comunicação entre o conversor A/D e microcontrolador

Da mesma forma que existem muitos conversores A/D que operam com níveis de tensão entre 0 a 5V ou 0 a 3,3V, o mesmo acontece para microcontroladores. Logo será escolhido um microcontrolador que opere em uma dessas duas faixas. Caso o conversor A/D e o microcontrolador operem em níveis de tensão diferentes, o que é possível dadas

as possibilidades, um conversor de nível lógico será utilizado para que as tensões fiquem compatíveis.

O tipo de comunicação entre o conversor A/D e microcontrolador deverá ser serial, pois dessa forma o conversor A/D poderá transmitir os dados de três correntes e uma tensão através de uma saída que estará conectada a entrada de comunicação serial do microcontrolador. O protocolo serial não será especificado, ele será consequência da seleção do conversor A/D e microcontrolador que melhor atendem as outras especificações.

### 3.3.7 Resolução, taxa de amostragem e quantidade de canais do conversor A/D

A taxa de amostragem do conversor A/D não precisa ser elevada, tendo em mente que apenas grandezas médias serão calculadas, como já foi discutido anteriormente. Porém é comum encontrar conversores A/D com taxas de amostragem por volta de 1000 amostras por segundo. Sabendo disso será escolhido um conversor A/D que atenda suas outras especificações e que tenha a maior taxa de amostragem possível.

Para analisar a resolução do conversor A/D será levado em consideração sensores que tenham como saída uma tensão na faixa de 0 a 5V, tensão máxima a ser medida de 220V e corrente máxima a ser medida de 20A. Para tal caso um conversor com 12bits de resolução, a menor amplitude de tensão de saída que poderá ser resolvida é de  $\frac{5V}{2^{12}} = 1,22mV$ . Para o sensor de tensão isso significa que a menor amplitude de tensão de entrada que será resolvida é de aproximadamente  $\frac{220V}{5V}1,22mV = 53,7mV$ . Para o sensor de corrente isso significa que a menor amplitude de corrente de entrada que será resolvida é de aproximadamente  $\frac{20A}{5V}1,22mV = 4,88mA$ . Esses valores podem ser considerados pequenos dadas as faixas de valores de medição especificadas para o projeto, logo a especificação de resolução do conversor A/D será de 12bits.

Os erros na medição do consumo de energia causados pela resolução serão calculados empiricamente através da comparação dos valores da Régua Inteligente com um dispositivo de medição mais preciso.

A quantidade de canais deverá ser igual a 4, pois a Régua Inteligente irá medir três correntes e uma tensão.

### 3.3.8 Microcontrolador com suporte Wi-Fi

O microcontrolador deve atender às outras especificações e deve possuir um *hardware* e *firmware* que facilite a comunicação Wi-Fi. Caso contrário o nível de complexidade do projeto iria aumentar bastante.

### 3.3.9 Consumo de energia dos módulos

A especificação do consumo de cada um dos módulos deverá ser a menor possível atendendo às outras especificações de cada um deles. Para verificarmos o desempenho do consumo da régua iremos utilizar como parâmetro de comparação a potência de operação do *smartplug* da referência [Etekcity \(2018\)](#), que está entre 0,7W e 1,2W. Esse *smartplug* é um dos mais vendidos no site da Amazon dos EUA.

### 3.3.10 Resumo das especificações do *hardware*

Os itens abaixo resumem as especificações do *hardware* da Régua Inteligente:

- a) Faixa de medição do sensor de tensão deve estar entre  $\pm 220V_{ef}$  ou mais;
- b) Faixa de medição do sensor de corrente deve estar entre  $\pm 15A$  ou mais;
- c) Nível máximo de tensão suportado pelo sensor de tensão deve ser no mínimo  $\pm 240V_{ef}$ ;
- d) Nível máximo de corrente suportado pelo sensor de corrente deve ser no mínimo  $\pm 20A$ ;
- e) A banda de passagem dos sensores deverá ser a maior possível;
- f) As tensões de operação dos sensores, conversor A/D e microcontrolador devem ser de 0 a 5V ou 0 a 3,3V;
- g) A taxa de amostragem do conversor A/D deve ser próxima de 1000 amostras por segundo;
- h) A resolução do conversor A/D deve ser de 12 bits;
- i) O tipo de comunicação entre o conversor A/D e microcontrolador deve ser serial;
- j) A quantidade de canais do conversor A/D deve ser 4;
- k) O microcontrolador deve possuir um *hardware* e *firmware* que facilite a comunicação Wi-Fi;
- l) O consumo de energia dos módulos deve ser o menor possível.

Ao analisar a Figura 2, percebe-se que a fonte de tensão e o conversor de tensão DC/DC não foram especificados. Isso acontece pois será utilizada uma fonte chaveada pronta que forneça um nível de tensão de 5V e que possa fornecer uma quantidade de corrente suficiente para a alimentação de todos os módulos da Régua Inteligente. Caso algum módulo precise ser alimentado com 3,3V um conversor de tensão DC/DC será utilizado.

## 3.4 Epecificações do *software*

Sob o ponto de vista da estruturação do servidor que cuidará das requisições e armazenará informações em um banco de dados, as especificações serão definidas através da estimativa da quantidade de requisições por segundo que o servidor deverá tratar para atender uma determinada quantidade de clientes de acordo com um tempo máximo de resposta por requisição. Essa informação ajudará na escolha do servidor HTTP e do banco de dados na seção de componentes e ferramentas. Ainda no servidor, deverá existir uma *Application Programming Interface* (API) desenvolvida em alguma linguagem que trate as requisições e interaja com o banco de dados.

Será também especificado a interface de visualização no navegador que o usuário irá utilizar para visualizar os dados armazenados no servidor.

Por último as especificações do *software* do microcontrolador serão apresentadas.

### 3.4.1 Quantidade de requisições por segundo

A quantidade de requisições por segundo pode ser estimada de acordo com a equação abaixo:

$$R = QN \quad (3.2)$$

Na equação 3.2,  $R$  representa a quantidade de requisições por segundo,  $Q$  a quantidade de requisições geradas por segundo por cliente e  $N$  a quantidade total de clientes que possuem uma Régua Inteligente.

Imagine um cenário em que a quantidade de clientes que possuem o produto é igual a mil e que o número de requisições geradas por segundo por um cliente seja de duas a cada cinco segundos. Entende-se aqui como cliente, tanto a Régua Inteligente que enviará informações para o servidor quanto usuários que requererão informações do servidor. Essas duas entidades geram requisições ao servidor.

A justificativa para a escolha da quantidade de clientes se dá imaginando que o produto seja aceito no mercado com intuito de se ter uma ideia das demandas de processamento para que um planejamento preventivo seja feito no primeiro momento, evitando problemas de performance que poderiam acontecer no futuro.

Para explicar a escolha da quantidade de requisições geradas por segundo por cliente é imaginado um cenário em que cada Régua Inteligente enviará uma requisição a cada cinco segundos ao servidor, ou seja, enviará a informação de potência média calculada no intervalo de cinco segundos ao servidor, e que um usuário enviará uma requisição a cada cinco segundos para o servidor requerendo a informação dessa potência média calculada. Assume-se aqui que o conhecimento da potência média a cada cinco segundos gerará um

conjunto de dados que possibilitará o cálculo do valor de energia média do eletrodoméstico com qualidade satisfatória, tal afirmação poderá ser comprovada posteriormente quando as validações da Régua forem feitas, onde a comparação dela com outros dispositivos de medição será feita.

Feitas essas considerações, a expectativa da quantidade de requisições por segundo ao servidor será de  $Q = \frac{2}{5} \frac{\text{requisições}}{\text{segundo}} \cdot 1000\text{cliente} = 400 \frac{\text{requisições}}{\text{segundo}}$ . Então, a primeira especificação é que o servidor seja capaz de tratar 400 requisições HTTP por segundo. Esse número pode ser superestimado tendo em mente que algumas réguas podem estar desligadas e que nem todos os clientes estão acompanhando o consumo de energia de seus dispositivos, mas serve como referência para uma especificação inicial.

Outro dado que pode ser estimado é a quantidade de armazenamento necessária em um ano devido ao armazenamento das requisições geradas pelas réguas e armazenadas no banco de dados. Supondo que cada requisição enviada pela régua tenha um tamanho de 1kB a quantidade de armazenamento necessária em um ano será:

$$\text{Armazenamento} = 400 \times 60 \times 60 \times 24 \times 365 \frac{kB}{ano} \approx 13 \frac{TB}{ano} \quad (3.3)$$

Esse número pode estar superestimado pois o tamanho de uma requisição da régua pode ser menor que 1kB uma vez que apenas um número inteiro será enviado ao servidor, representando a potência média calculada pela régua em um intervalo de cinco segundos. A estimativa do armazenamento necessário foi calculada apenas como curiosidade e não será uma especificação do projeto.

### 3.4.2 API de tratamento de requisições e interação com o banco de dados

No servidor deverá existir uma aplicação que trate as requisições HTTP que são recebidas por ele, sendo capaz de armazenar os dados gerados pelas réguas no banco de dados e tratando as requisições de usuários por esses dados.

Cada servidor web fornece diferentes tipos de suporte para a implementação dessa API, como foi discutido no capítulo de revisão da literatura. Porém será buscado um servidor que de suporte a implementação dessa API através de uma linguagem que possuí bibliotecas ou *frameworks* que facilitem o desenvolvimento das especificações deste item.

### 3.4.3 Interface de visualização dos dados pelo usuário

A interface de visualização deverá ser desenvolvida para o ambiente de navegadores e deverá possuir gráficos que mostram a potência e energia média consumidas por eletrodoméstico por dia, semana e mês. Além disso ela deverá mostrar a energia total consumida por dia, semana e mês e mostrar uma estimativa do valor a ser pago para cada

uma dessas situações, esse é o ponto mais vital do projeto pois ele dará a intuição ao usuário do consumo de cada eletrodoméstico.

Para implementar essa interface, deverá ser utilizada um biblioteca ou *framework* que facilite o desenvolvimento do *front-end* do projeto, tendo em mente as tecnologias mais utilizadas atualmente no mercado.

#### 3.4.4 Software do microcontrolador

O *software* implementado deverá realizar duas tarefas distintas.

A primeira delas é a de se comunicar com o conversor A/D através de uma comunicação serial com o intuito de processar os dados convertidos por ele para que em seguida a informação de potência média calculada possa ser enviada para o servidor.

A segunda tarefa é a de enviar informações através do protocolo HTTP para o servidor web, para que as informações possam ser armazenadas.

Bibliotecas deverão ser utilizadas para que a realização dessas tarefas sejam facilitadas.

#### 3.4.5 Resumo das especificações do software

Os itens abaixo resumem as especificações do *software* da Régua Inteligente.

- a) O servidor web deverá ser capaz de tratar 400 requisições por segundo;
- b) Deverá existir um banco de dados para que os dados da aplicação possam ser armazenados e recuperados de uma forma estruturada;
- c) Deverá existir uma API que trate as requisições entregues ao servidor. Bibliotecas ou *frameworks* devem ser utilizados para facilitar a realização desta tarefa;
- d) A interface de visualização deverá ser desenvolvida para o ambiente de navegadores e deverá possuir informações (gráficos e consumo total) relevantes ao usuário;
- e) A interface de visualização deverá ser desenvolvida com a utilização de bibliotecas ou *frameworks* mais utilizados atualmente;
- f) O *software* do microcontrolador deverá comunicar com o conversor A/D para processar os dados convertidos por ele e enviar a informação gerada ao servidor web.

# 4 Validação do sistema

Neste capítulo serão apresentados como os processos de validação do *hardware*, *software* e do conjunto serão feitos.

## 4.1 Processo de validação do *hardware*

O primeiro passo da validação do *hardware* da régua será o levantamento das relações entre entrada e saída de tensões dos sensores de tensão e corrente, e também da resposta em frequência dos mesmos. Para a realização dessa validação será utilizado um gerador de sinais, fonte de tensão, osciloscópio, multímetro e cargas experimentais. Com essas relações será possível comparar os resultados obtidos com as informações presentes nos *datasheets* dos sensores. A execução dessa tarefa auxiliará na calibragem dos sensores no *software* do microcontrolador.

Em seguida será feita a conexão do conversor A/D com os sensores e o microcontrolador. Um programa simples será implementado para verificar se os valores recebidos pelo microcontrolador fazem sentido e possíveis calibrações serão feitas. A verificação da coerência de tais valores será percebida através da utilização de outros instrumentos de medição, como o osciloscópio e multímetro.

Com o recebimento correto dos dados pelo microcontrolador e a calibração finalizada, o desenvolvimento do *software* do projeto terá início.

## 4.2 Processo de validação do *software*

Primeiramente será instalado e configurado o servidor web de tal forma que ele consiga dar suporte à uma aplicação que tratará as requisições recebidas por ele.

Em seguida será instalado e configurado o banco de dados que irá armazenar as informações de potência média enviadas pelas réguas.

Com o servidor e banco de dados configurados, a API será desenvolvida. Testes serão feitos para verificar se a interação com o banco de dados e o tratamento de requisições estão sendo feitos corretamente. Cenários de testes simples serão feitos para validar esta parte do desenvolvimento.

Ao finalizar a API, a configuração da estrutura do servidor estará completa. Daí será implementado o *software* do microcontrolador que irá processar os dados convertidos pelo conversor A/D e enviará para o servidor via Wi-Fi os dados de potência média

calculados. Testes serão realizados para verificar se os dados enviados pela régua estão sendo tratados pela API e armazenados no banco de dados corretamente.

A última implementação de *software* será a da interface do navegador. Primeiramente será desenvolvida a interface do *front-end* de acordo com as especificações ditas anteriormente. Em seguida será implementada as requisições ao servidor para resgatar as informações de potência armazenadas no banco de dados. Para validar essa etapa a interface será analisada para verificar se os dados que estão sendo recebidos estejam sendo exibidos de forma correta.

### 4.3 Processo de validação do conjunto *hardware* e *software*

A última validação que será feita irá verificar a performance da Régua Inteligente quando comparada ao desempenho de outros *smartplugs* do mercado. Para isso serão escolhidos diferentes tipos de carga e o monitoramento da energia consumida por eles será feita simultaneamente pela Régua Inteligente e outro *smartplug* com precisão conhecida. Caso os resultados encontrados sejam próximos isso dará um indicativo da performance da Régua Inteligente no monitoramento de energia daquele determinado tipo de carga.

# 5 Materiais e Métodos

Depois das especificações do projeto, foi feita uma pesquisa no mercado em busca de componentes eletrônicos e ferramentas de *software* que poderiam ser utilizados na implementação da régua. Nesta seção serão apresentados as escolhas do *hardware* e *software* resultantes da pesquisa feita.

## 5.1 *Hardware*

Nesta seção serão apresentados os componentes eletrônicos que foram escolhidos para a implementação da régua. A sequência de apresentação será a seguinte, sensor de tensão, sensor de corrente, conversor A/D, conversor lógico digital, microcontrolador e conversor DC/DC.

### 5.1.1 Sensor de tensão

O sensor de tensão escolhido pode ser visto na figura 5, ele é baseado no transformador ZMPT101B.

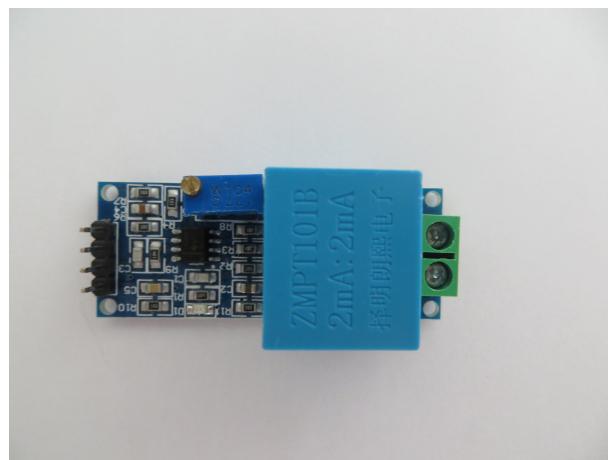


Figura 5 – Sensor de tensão baseado no transformador ZMPT101B

A tabela 2 indica as especificações do sensor, elas foram retiradas da referência Interplus... (2018).

Analizando a tabela acima percebemos que a faixa de tensão que o sensor mede,  $\pm 250V_{ef}$ , atende as especificações a) e c) do *hardware*. A tensão de saída está entre 0 e 5V, o que atende a especificação f). A informação de sensibilidade do sensor não foi encontrada, mas a referência Abubakar et al. (2017) apresenta um trabalho do levantamento da curva

Tabela 2 – Especificações do sensor de tensão

Parâmetro	Valor
Tensão de alimentação do módulo	5 a $30V_{dc}$
Corrente de consumo	$2mA$
Potência consumida	$10mW$ (à $5V_{dc}$ de alimentação)
Tensão de entrada	$\pm 250V_{ef}$
Tensão de saída	0 a $5V$
Frequência de operação	50 a $60Hz$
Precisão de leitura	$\pm 0,5\%$
Temperatura de operação	-40°C a 70°C
Dimensões	49mmX19mmX22mm
Peso	20g

de resposta do sensor e será utilizada para comparar com a curva encontrada no processo de validação.

Ainda resta saber qual a banda de passagem do sensor para verificar a especificação e) do *hardware*, mas essa informação não foi encontrada na pesquisa. Porém o sensor opera em  $60Hz$  que é a frequência mais comum nas casas brasileiras, além disso será verificada a banda de passagem do sensor na parte de validação.

A potência consumida de  $10mW$  pode ser considerada pequena quando comparada à potência da referência [Etekcity \(2018\)](#) que é aproximadamente  $1W$ . A temperatura de operação está dentro de uma faixa aceitável de operação, tendo em mente a faixa dos dispositivos eletrônicos usuais encontrados no mercado.

As outras informações da tabela foram apresentadas como informações adicionais, uma vez que não são especificações do projeto.

### 5.1.2 Sensor de corrente

O sensor de corrente escolhido pode ser visto na figura 6, seu nome é ACS712.

A tabela 3 indica as especificações do sensor, elas foram retiradas da referência [Allegro... \(2017\)](#).

Analizando a corrente de entrada,  $\pm 20A$ , verificamos que as especificações b) e d) do *hardware* são satisfeitas. Como a tensão de saída do sensor está entre 0,5 a 4,5V a especificação f) é satisfeita. A banda de passagem é  $80kHz$  que é um valor bem alto e, por isso, atende a especificação e). Esse fato talvez possibilite o cálculo das potências aparente, reativa e do fator de potência.

A curva de resposta do sensor está indicada na figura 7, ela foi retirada do *datasheet* [Allegro... \(2017\)](#). Percebe-se dela a justificativa do valor da sensibilidade de  $100mV/A$ . Ainda vale ressaltar que ela está parecida com o que foi discutido sobre a figura 3.

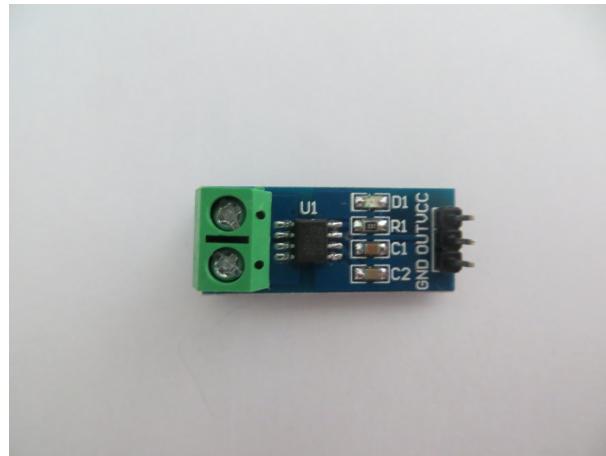


Figura 6 – Sensor de corrente ACS712

Tabela 3 – Especificações do sensor de corrente

Parâmetro	Valor
Tensão de alimentação do módulo	$5V_{dc}$
Corrente de consumo	$10mA$
Potência consumida	$50mW$
Corrente de entrada	$\pm 20A$
Tensão de saída	0,5 a 4,5V
Sensibilidade	$100mV/A$
Banda de passagem	$80kHz$
Precisão de leitura	$\pm 1.5\%$ em $25^\circ C$
Temperatura de operação	$-40^\circ C$ a $85^\circ C$
Dimensões	$31mm \times 13mm \times 11mm$
Peso	< 1g

A potência consumida de  $50mW$  pode ser considerada pequena quando comparada à  $1W$ . A temperatura de operação está dentro de uma faixa aceitável de operação, tendo em mente a faixa dos dispositivos eletrônicos usuais encontrados no mercado.

As outras informações da tabela foram apresentadas como informações adicionais, uma vez que não são especificações do projeto.

### 5.1.3 Conversor A/D

O conversor A/D escolhido pode ser visto na figura 8, seu nome é ADS1015.

A tabela 4 indica as especificações do sensor, elas foram retiradas da referência Adafruit... (2017).

Verificando a tabela, percebemos primeiramente que a tensão de operação do conversor pode ser de  $5V$ , o que atende a especificação f) do hardware. A taxa de amostragem pode ir até  $3,3ksps$  e a resolução é de  $12bits$ , satisfazendo as especificações g) e h). O

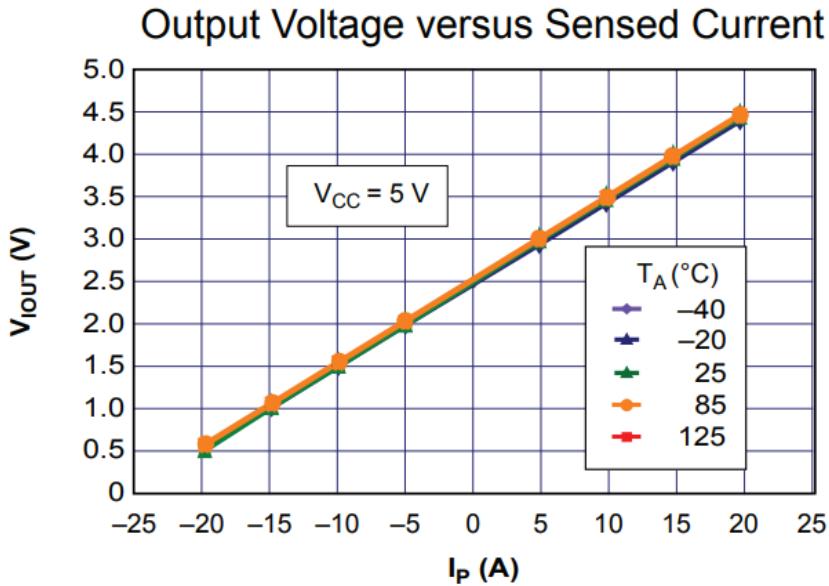


Figura 7 – Curva do sensor ACS712

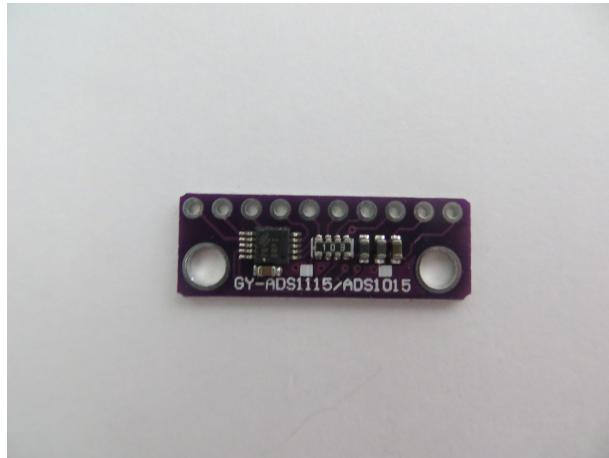


Figura 8 – Conversor A/D ADS1015

tipo de comunicação é serial, do tipo IC2. Posteriormente, quando o microcontrolador for apresentado, será visto que ele também possui um tipo de comunicação do tipo IC2. Essas duas informações fazem que a especificação i) seja atendida. Por fim, a quantidade de canais é 4, atendendo dessa forma a especificação j)).

A potência consumida de  $0,75\text{mW}$  pode ser considerada muito pequena quando comparada à  $1\text{W}$ . A temperatura de operação está dentro de uma faixa aceitável de operação, tendo em mente a faixa dos dispositivos eletrônicos usuais encontrados no mercado.

As outras informações da tabela foram apresentadas como informações adicionais, uma vez que não são especificações do projeto.

Tabela 4 – Especificações do conversor A/D

Parâmetro	Valor
Tensão de alimentação do módulo	2 a $5V_{dc}$
Corrente de consumo	$150\mu A$ em $25^\circ C$
Potência consumida	$0,75mW$ (à $5V_{dc}$ de alimentação)
Quantidade de canais	4
Resolução	12bits
Taxa de amostragem programável	128sps a 3,3ksp
Comunicação serial	I2C
Temperatura de operação	$-40^\circ C$ a $125^\circ C$
Dimensões	$25mm \times 9mm \times 2mm$
Peso	< 1g

#### 5.1.4 Conversor lógico digital

A figura 9 mostra um conversor lógico digital de 5V/3,3V.

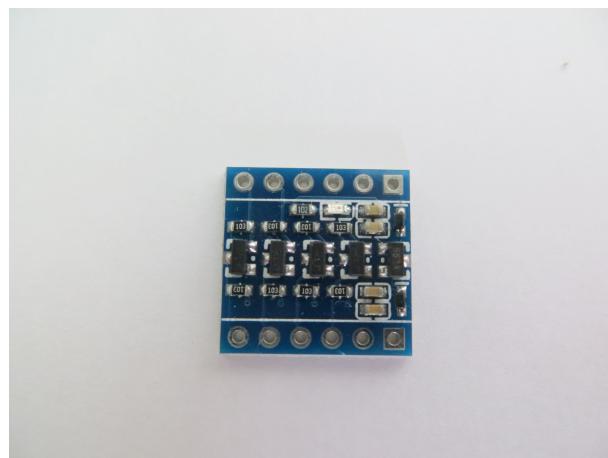


Figura 9 – Conversor lógico digital 5V/3,3V

A tabela 5 indica as especificações do conversor lógico digital, elas foram retiradas da referência [Your... \(2018\)](#).

Tabela 5 – Especificações do conversor lógico digital

Parâmetro	Valor
Tensões de alimentação	3,3V e 5,5V
Máxima corrente suportada	$150mA$
Máximo <i>baud rate</i> estável	$28800bps$
Dimensões	$16mm \times 16mm \times 2mm$
Peso	< 1g

A justificativa para a utilização do conversor lógico digital se dá quando verificamos as tensões de operação do conversor A/D e do microcontrolador. Como já vimos, o conversor opera com 5V. Na próxima seção veremos que o microcontrolador opera com tensões de

3,3V, por isso é necessário uma conversão de nível lógico na comunicação IC2 do conversor, fazendo que o sinal gerado pelo mesmo esteja em 3,3V, garantindo dessa forma a correta comunicação entre os dois dispositivos.

O máximo *baud rate* estável indica a maior taxa de transmissão de bits na comunicação serial que o conversor de nível lógico consegue atingir. Esse valor deve ser respeitado para que a comunicação aconteça sem erros de transmissão. Deve-se ficar atento também à máxima corrente suportada pelo dispositivo.

### 5.1.5 Microcontrolador

O microcontrolador escolhido pode ser visto na figura 10, seu nome é ESP8266. Existem diferentes tipos de módulos que utilizam o ESP8266, o nome do que está sendo utilizando neste projeto é o D1 mini.

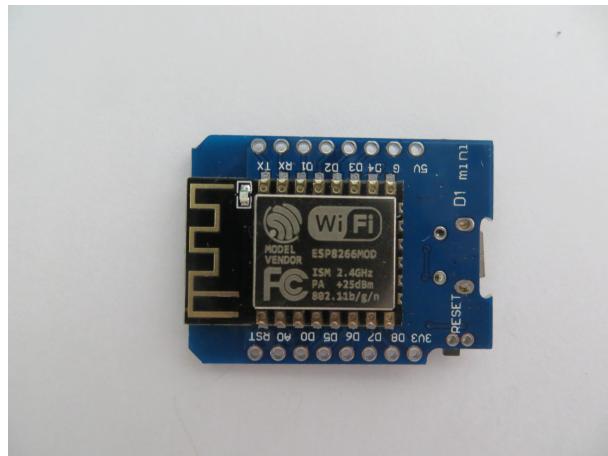


Figura 10 – Microcontrolador ESP8266 D1 mini

A tabela 6 indica as especificações do microcontrolador, elas foram retiradas da referência [Espressif... \(2018\)](#).

Analizando a tabela 6, nota-se que a tensão de alimentação do microcontrolador é de 3,3V, por isso é preciso um conversor de nível lógico, como já foi discutido anteriormente. A comunicação serial é a I2C, que está de acordo com a forma de comunicação do conversor A/D, atendendo a especificação i) do *hardware*. A memória de 4MB será suficiente para armazenar os programas que implementam a interface de comunicação com o conversor A/D, o processamento dos dados e a comunicação Wi-Fi.

Existem onze pinos digitais e um analógico que podem ser utilizados para realizar outras tarefas. LEDs indicativos poderiam ser acionados para indicar informações e relés poderiam ser chaveados para ligar ou desligar um eletrodoméstico remotamente. Essas funções não são especificações do projeto, mas vale a reflexão para uma possível implementação no futuro.

Tabela 6 – Especificações do microcontrolador

Parâmetro	Valor
Tensão de alimentação	$3,3V_{dc}$
Corrente de consumo	$80mA$
Potência consumida	$264mW$
Quantidade de pinos digitais I/O	11
Quantidade de pinos analógicos	1
Mémoria flash	$4MB$
Comunicação serial	I2C
Firmware	NodeMCU
Stack TCP/IP integrada	Sim
Suporte aos protocolos Wi-Fi	802.11 b/g/n
Tipo de antena	PCB Trace
Temperatura de operação	$-40^{\circ}C$ a $125^{\circ}C$
Dimensões	$34mm \times 25mm \times 8mm$
Peso	5g

Uma informação importante desta tabela é a presença do *firmware* NodeMCU. Ele fornece um suporte para que comunicações Wi-Fi, dentro dos protocolos 802.11 b/g/n, possam ser realizadas, implementando um código que trata a camada TCP/IP da comunicação. Existe ainda uma antena construída na placa de circuito impresso que melhora o sinal emitido pelo módulo. Esses fatos fazem com que a especificação k) seja atendida.

A potência consumida de  $264mW$  foi a maior dentre todos os componentes, e comparada à  $1W$  ela é expressiva. Isso era esperado pois o microcontrolador é o componente que possui a arquitetura mais complexa e por isso exige maior potência para que possa funcionar. A temperatura de operação está dentro de uma faixa aceitável de operação, tendo em mente a faixa dos dispositivos eletrônicos usuais encontrados no mercado.

As outras informações da tabela foram apresentadas como informações adicionais, uma vez que não são especificações do projeto.

### 5.1.6 Conversor DC/DC

A figura 11 mostra um conversor DC/DC de 5V/3,3V.

A tabela 7 indica as especificações do conversor DC/DC, elas foram retiradas da referência [Hyelesiontek \(2018\)](#).

Quando é analisada a alimentação de todos os dispositivos que serão utilizados para a construção da régua, percebemos que o ESP8266 é o único que requer  $3,3V$ . Por causa disso é preciso um conversor DC/DC *step-down* para que a alimentação seja possível.

Será utilizada uma fonte chaveada que fornece  $5V$  de tensão para todos os circuitos, exceto o microcontrolador que será alimentado pela tensão de  $3,3V$  quer será fornecida

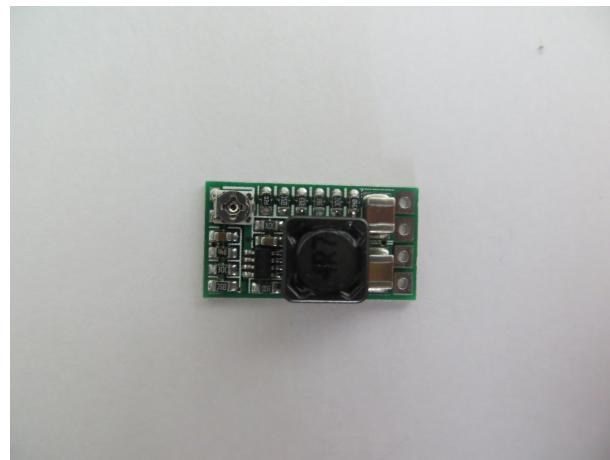


Figura 11 – Conversor DC/DC 5V/3.3V

Tabela 7 – Especificações do conversor DC/DC

Parâmetro	Valor
Tensão de alimentação	4,5 a 24V <sub>dc</sub>
Máxima corrente suportada	3A
Eficiência	97,5%
Precisão da conversão de tensão	±0,5%
Dimensões	20mmX10mmX5mm
Peso	< 1g

pelo conversor DC/DC.

### 5.1.7 Estrutura de *hardware* resultante

A figura 12 indica a estrutura completa do *hardware* construído no *protoboard*. Nela está indicada cada um dos módulos citados anteriormente. Perceba como essa figura segue a estrutura apresentada na figura 2.

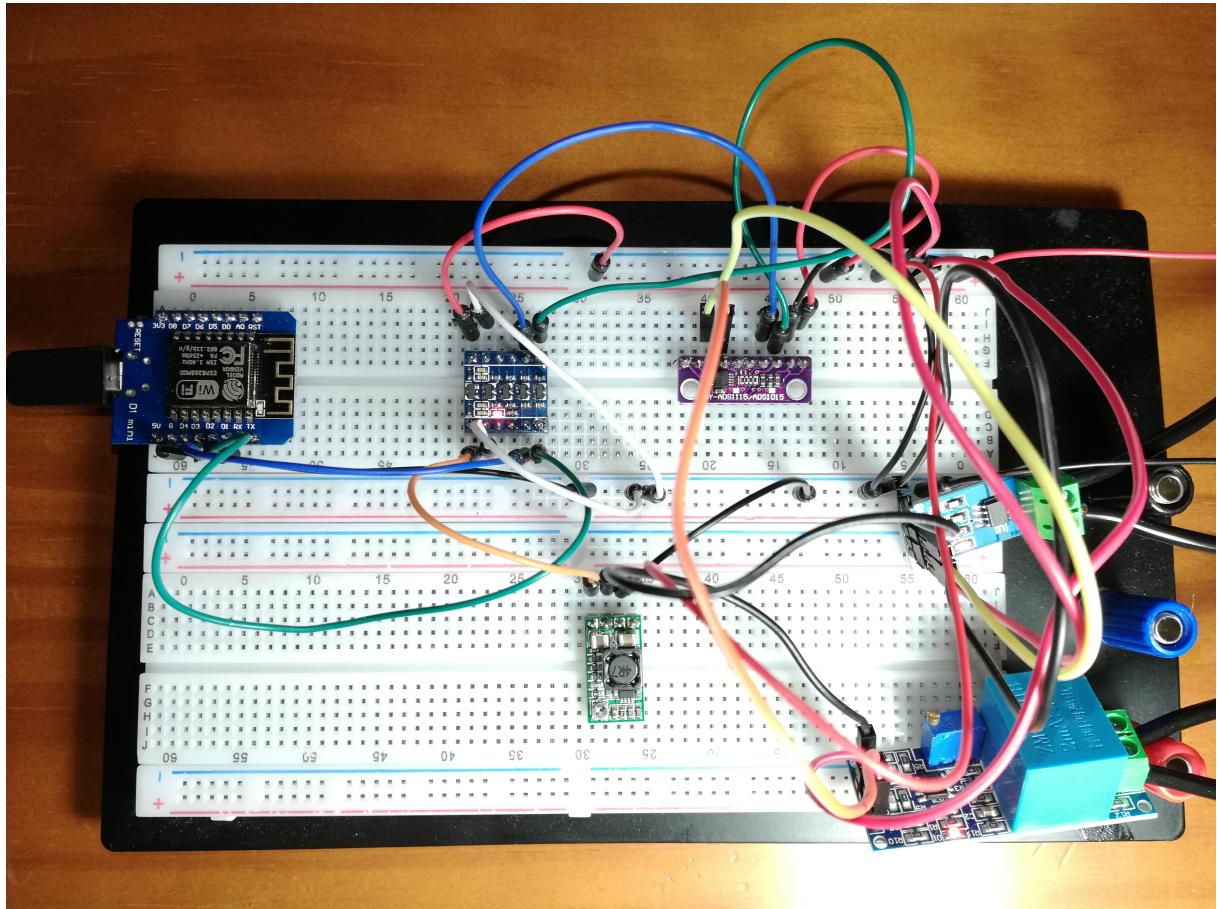


Figura 12 – Estrutura completa do *hardware*

## 5.2 Software

Nesta seção serão apresentados os *softwares* que serão utilizados ou implementados para que a comunicação entre os diferentes dispositivos do projeto seja possível. O sistema operacional utilizado para o desenvolvimento dos *softwares* será o Ubuntu 18.04.1 LTS. O Ubuntu e todas as ferramentas que aqui serão explicadas são *open source*. Essa escolha foi feita pois o desempenho dessas ferramentas será suficiente para atender as especificações deste projeto, como será apresentado, além de existir uma grande comunidade que pode oferecer suporte. Outra vantagem é a disponibilidade de diversas bibliotecas e *frameworks* desenvolvidos pela comunidade que podem facilitar a implementação deste projeto.

A sequência da apresentação dos itens deste seção será a seguinte, servidor HTTP Apache, banco de dados MySQL, API em Python, interface do navegador em Angular e *software* do microcontrolador.

### 5.2.1 Servidor HTTP

O servidor web escolhido foi o Apache, as razões para tal escolha serão justificadas nos próximos parágrafos. A figura 13 indica a logomarca do servidor.



Figura 13 – Servidor HTTP Apache

As especificações que dizem respeito ao servidor web são a capacidade do tratamento de requisições por segundo, que deve ser 400, e que o tempo de resposta por requisição deve ser menor do que dois segundos e meio. Antes de analisarmos esses aspectos, iremos verificar a utilização do Apache nos últimos anos. A figura 14 indica a porcentagem dos servidores mais utilizados para hospedar os sites com maior tráfego na internet, ela foi retirada da referência [Netcraft \(2018\)](#).

Percebemos analisando a figura acima que o servidor Apache é o mais utilizado nos últimos dez anos para hospedar os sites com maior tráfego na internet. Caso seja de interesse do leitor, a referência [Netcraft \(2018\)](#) ainda mostra outros cenários de utilização de servidores e o Apache é o mais utilizado na maioria das situações. O Apache se torna atrativo por ser um servidor que já está consolidado no mercado a muito tempo, é *open source*, possuí uma comunidade muito grande e ativa e tem uma performance muito boa para desempenhar diferentes funções. Uma grande quantidade de informações sobre o Apache pode ser encontrada nas referências [Apache \(2018\)](#) e [docs... \(2018\)](#) (documentação).

Ainda falando sobre a figura 14, nota-se que a utilização do servidor Nginx vem aumentando consideravelmente desde 2012. Esse servidor possuí uma arquitetura diferente de outros servidores, como o Apache, e tem um desempenho de operação muito bom, por isso ele tem sido utilizado cada vez mais pelo mercado. Mais informações sobre o servidor Nginx podem ser encontradas na referência [Nginx \(2018\)](#).

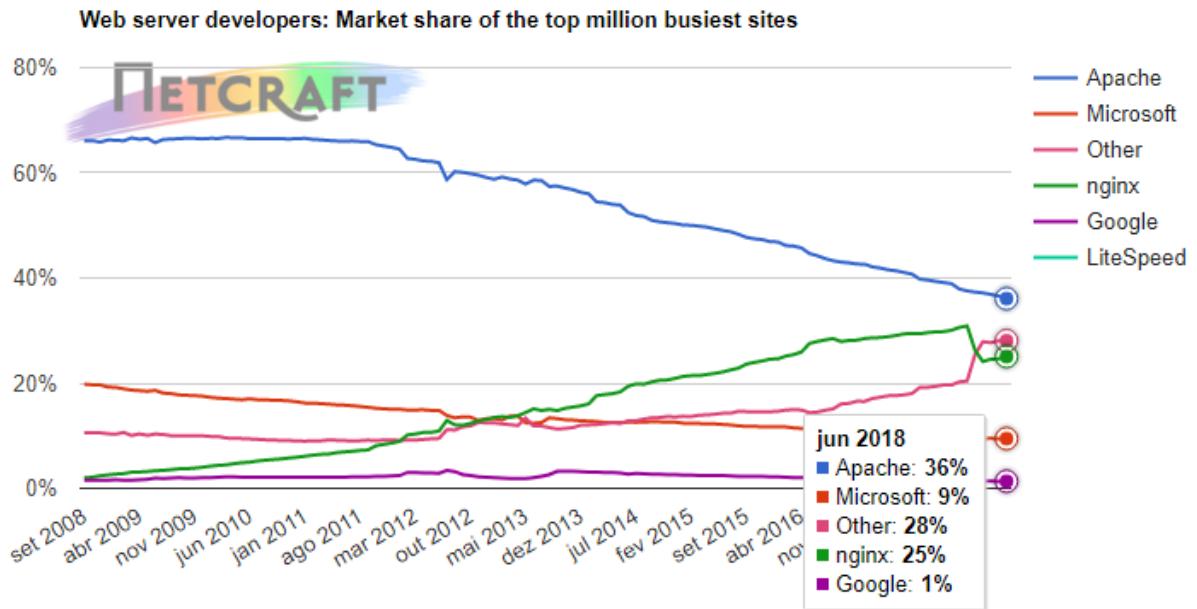


Figura 14 – Servidores mais utilizados em sites com alto tráfego

O servidor da Microsoft, IIS, manteve um porcentagem de domínio quase constante de 2008 até 2016, quando um aumento na sua utilização aconteceu até 2018. Por ser um servidor pago a sua utilização é menor, porém o seu desempenho é suficiente para desempenhar muitas funções. Mais informações sobre o IIS podem ser encontradas em [IIS \(2018\)](#).

Voltando a atenção para as especificações a) e b) do *software*, são apresentadas as figuras 15 e 16 para mostrar que o Apache atende a essas especificações, elas foram retiradas da referência [Liron \(2017\)](#).

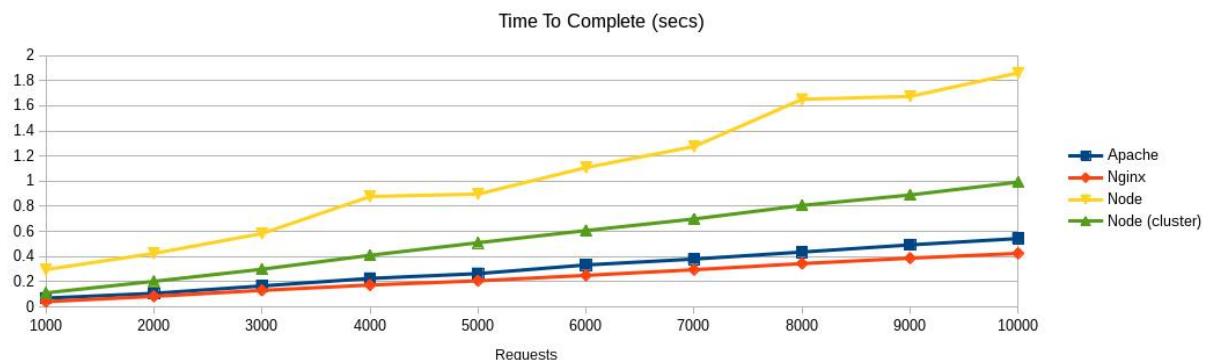


Figura 15 – Desempenho dos servidores para diferentes quantidades de requisições por segundo

A figura 15 mostra o tempo de tratamento em segundos de diferentes servidores para diferentes quantidades de requisições por segundo, cada requisição tinha um tamanho de

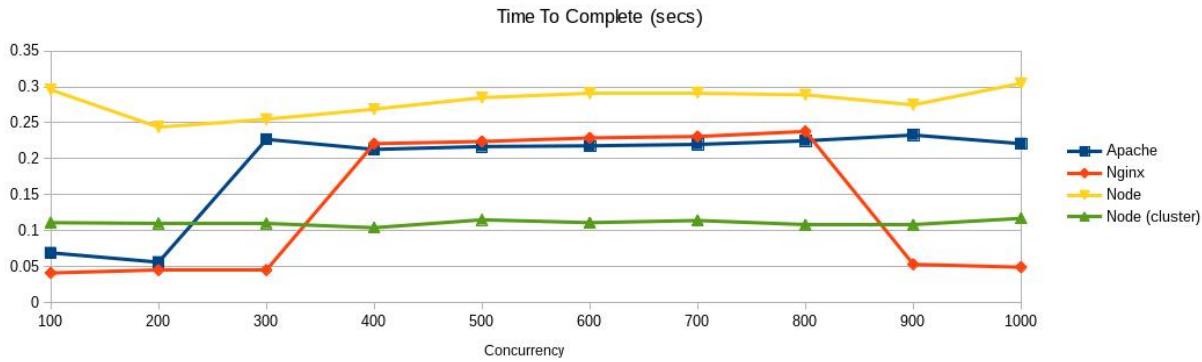


Figura 16 – Desempenho dos servidores para diferentes concorrências de usuários (1000 requisições por segundo)

aproximadamente 11kB. Analisando esse gráfico notamos que o servidor Apache conseguiu tratar 10000 requisições por segundo em um tempo de aproximadamente 0,4 segundos, o Nginx tratou a mesma quantidade de requisições em aproximadamente 0,3 segundos.

A figura 16 exibe um gráfico do tempo de tratamento em segundos de diferentes servidores para diferentes níveis de concorrência de usuários, isso quando cada um deles faz 1000 requisições por segundo com cada requisição tendo um tamanho de aproximadamente 11kB. Para uma concorrência de 1000 usuários o Apache demorou aproximadamente 0,3 segundos, o Nginx na mesma situação teve uma performance de 0,05 segundos.

Esses duas evidências apresentadas nos dois parágrafos anteriores mostram que o servidor Apache atende as especificações a) e b) do *software*, uma vez que o cenário imaginado para criar as especificações foi o de 1000 requisições por segundo, que é menos exigente do que o cenário de 10000 requisições por segundo. De fato, analisando a figura 15 o tempo para tratar 1000 requisições foi de aproximadamente 0,1 segundos.

Como foi dito anteriormente, o desempenho do servidor Nginx é muito bom para os diferentes cenários, sendo inclusive melhor do que o desempenho do Apache para a maioria dos casos apresentados nas figuras. Verificando a figura 16, percebemos que em níveis de concorrência intermediários, entre 400 e 800 usuários, o Nginx tem desempenho próximo ao Apache, porém para níveis de concorrência maiores do que 800 usuários e menores do que 400 usuários, o Nginx tem o desempenho melhor, chegando a cerca de 10 vezes maior do que o do Apache em níveis de concorrência maiores do que 800 usuários. Isso indica que a utilização do servidor Nginx pode ser interessante quando níveis de concorrência elevados são esperados.

Ainda analisando as duas figuras é possível perceber as curvas de outro servidor, o Node.js. Neste trabalho não será discutido esse servidor, mas vale ressaltar que ele tem sido utilizado atualmente por muitas empresas para soluções de desenvolvimento em *front-end* e *back-end*. Mais informações sobre esse servidor podem ser encontradas na referência

Node... (2018).

Outros resultados similares aos mostrados pelas figuras 15 e 16, que indicam que o servidor Apache atende as especificações a) e b) do *software*, podem ser encontrados na referência Jarrod (2016).

### 5.2.2 Banco de dados

O banco de dados escolhido foi o MySQL, sua logomarca está indicada na figura 17. MySQL é um banco de dados relacional.



Figura 17 – Banco de dados relacional MySQL

Os motivos para a escolha desse banco são a facilidade de encontrar informações sobre ele, uma vez que ele é um banco utilizado a muito tempo no mercado, possuí uma documentação muito boa, é *open source* e possuí desempenho suficiente para o cenário imaginado de uso da Régua Inteligente.

As referências Bassil (2012), Saikia et al. (2015) e Islam et al. (2016) mostram o desempenho do MySQL para diferentes tipos de *query* executadas. Na referência Bassil (2012) é possível perceber que as *queries* mais simples demoram em torno de 100ms para buscar um milhão de dados. O cenário da Régua Inteligente é menos exigente que tais testes e as *queries* programadas serão bem simples, o que garante que a especificação b) do *software* seja satisfeita.

A modelagem das tabelas, suas relações e a programação da *queries* serão tratadas no capítulo de resultados. Mais informações sobre o banco de dados MySQL podem ser econtradas na referência MySQL (2018).

### 5.2.3 API em Python

A especificação c) do *software* diz que deve existir uma aplicação que trate as requisições recebidas pelo servidor. Neste projeto, tratar requisições significa processar as requisições recebidas pelo servidor Apache, ou enviando uma resposta ao cliente ou armazenando informações no banco de dados MySQL.

Para a implementação da API que exerce tais funções será utilizada a linguagem de programação Python. A figura 18 mostra a logomarca da linguagem.



Figura 18 – Linguagem Python

As principais razões para a escolha dessa linguagem estão indicadas nos seguintes tópicos:

- Código de fácil leitura e manutenção (linguagem de alto nível);
- Suporte a diferentes tipos de paradigmas de programação;
- Compatível com a maioria das plataformas e sistemas;
- Biblioteca padrão robusta;
- Muitos *frameworks* e bibliotecas *open source* disponíveis;
- Simplificação do desenvolvimento de *softwares* complexos;
- Comunidade muito grande e prestativa ao suporte.

Dessas razões a que mais me interessa é o tamanho da comunidade em volta da linguagem. Esse fato faz com que a quantidade de materiais disponíveis na internet seja muito grande, isso facilita o desenvolvimento caso problemas sejam encontrados. Mais informações sobre a linguagem podem ser encontradas na referência [Python \(2018\)](#).

Para que a comunicação da API com o MySQL seja possível será utilizado o *driver* MySQL Connector/Python. Esse *driver* é implementado através de um módulo em Python, ele possui um conjunto de classes que possibilitam a interação da API com o banco de dados MySQL de uma maneira simples. Informações detalhadas do funcionamento do MySQL Connector/Python podem ser encontradas na referência [MySQL... \(2018\)](#).

O último detalhe que deve ser tratado é a questão da comunicação da API com o servidor. Para que isso seja possível é necessário instalar o módulo mod\_wsgi no Apache. Esse módulo cria uma interface que possibilita que a API tenha acesso às variáveis do

ambiente do Apache e que ela possa tratar as requisições recebidas pelo servidor. A instalação desse módulo é necessária pois o Apache não dá suporte a aplicações em Python nativamente. A figura 19 simboliza a interface descrita. Mais informações sobre o módulo mod\_wsgi podem ser encontradas na referência [Graham... \(2018\)](#).

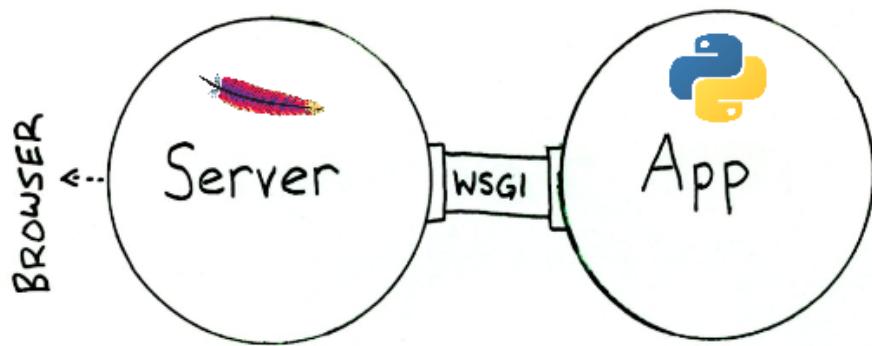


Figura 19 – Interface (mod\_wsgi) entre a aplicação em Python e o Apache

#### 5.2.4 Interface do navegador

Para a construção da interface do navegador será utilizado o *framework* Angular.

Esse *framework* é escrito em TypeScript (superset da linguagem JavaScript, com estrutura para a implementação orientada a objetos e abstrações) e serve para a construção de páginas web dinâmicas, juntamente com as linguagens HTML e CSS. A figura 20 indica o logo do *framework* Angular. Os próximos parágrafos discutirão alguns aspectos do *framework* e justificarão a escolha do mesmo para atender as especificações d) e e) do *software*.



Figura 20 – Framework Angular

Um projeto Angular tem como seu bloco de construção fundamental o componente. Em um projeto existirão mais de um componente; a quantidade de componentes utilizados depende da complexidade da página a ser construída, quanto mais complexa a página maior será a quantidade de componentes.

Um componente é composto de três arquivos, um HTML, outro CSS e um arquivo escrito em TypeScript que tem a extensão .ts. Cada componente representa uma determinada função ou característica da página Web a ser construída. Exemplos de componentes são a aba de navegação superior da página, a aba de navegação lateral, o conteúdo principal, as telas que compõem o site assim como as peculiaridades que elas possuem. No arquivo HTML são especificados os elementos tradicionais dessa linguagem, no arquivo CSS a estilização dos elementos e no arquivo TypeScript é especificada a lógica da dinâmica do componente. O poder do Angular reside nos recursos que ele proporciona para a integração dessas três linguagens e dos próprios componentes, com o intuito de atingir a criação de uma página dinâmica da maneira mais eficiente possível, com execução rápida, bem modularizada, segura, com tamanho reduzido e de fácil manutenção.

Com o Angular, é possível manipular o *template* escrito em HTML acessando as propriedades e eventos dos elementos HTML, assim como adicionar estilos dinamicamente que estão definidos no arquivo CSS.

Uma das principais ideias do *framework* é de criar o site como uma única página HTML. Quando um projeto Angular é finalizado e compilado, o resultado é um arquivo index.html que conterá um esqueleto HTML básico e junto a esse HTML um conjunto de arquivos em JavaScript que terão a função de manipular os elementos desse arquivo index.html para gerar o comportamento programado nos componentes. Todo o projeto gerado em TypeScript é compilado em JavaScript (linguagem interpretada pelos navegadores), e esses arquivos possuem a lógica programada do site sustentada pelo *framework* Angular, que também é compilado em JavaScript. Então basta o usuário baixar esses arquivos, juntamente com os arquivos de definição de estilos CSS, uma única vez e o site poderá ser visualizado sem mais requisições ao servidor, salvo os momentos que requisições e envios de dados são necessários. Com isso grande parte do esforço de execução do site é passado para o lado do cliente, aliviando o processamento do lado do servidor, onde o site é hospedado. Esse conceito nasce tendo em mente o constante crescimento do poder de processamento dos navegadores e dos computadores pessoais.

Um tratamento completo do *framework* pode ser encontrado na referência [Angular \(2018\)](#). No capítulo de resultados será apresentado o código resultante em Angular devido à implementação da interface para a Régua Inteligente, lá o entendimento dos conceitos discutidos aqui ficarão mais claros.

### 5.2.5 *Software* do microcontrolador

O *software* do microcontrolador deve ser capaz de ler os dados enviados do conversor A/D e de enviar os dados processados via Wi-Fi. Para que isso seja feito será instalado um pacote ao *Integrated Development Environment* (IDE) do Arduino que permite programar o ESP8266 nesse ambiente. Para resolver a comunicação do microcontrolador com o conversor

A/D serão utilizadas as bibliotecas Adafruit\_ADS1015 e Wire. Para resolver a questão da comunicação Wi-Fi serão utilizadas as bibliotecas ESP8266WiFi e ESP8266HTTPClient. Essas bibliotecas fornecem classes que facilitam o desenvolvimento das comunicações necessárias. Com isso a especificação f) do *software* é satisfeita. Mais informações sobre as bibliotecas podem ser encontradas nas referências [Adafruit \(2018\)](#) e [Arduino \(2018\)](#). No capítulo de resultados serão apresentados os códigos utilizados para programar o ESP8266 e ficará mais clara a utilização de cada uma dessas bibliotecas.

### 5.2.6 Estrutura de *software* resultante

A figura 21 indica a estrutura completa do *software*, mostrando as ferramentas citadas anteriormente e de que maneira elas se comunicam entre si. Perceba como essa figura segue a estrutura apresentada na figura 4.

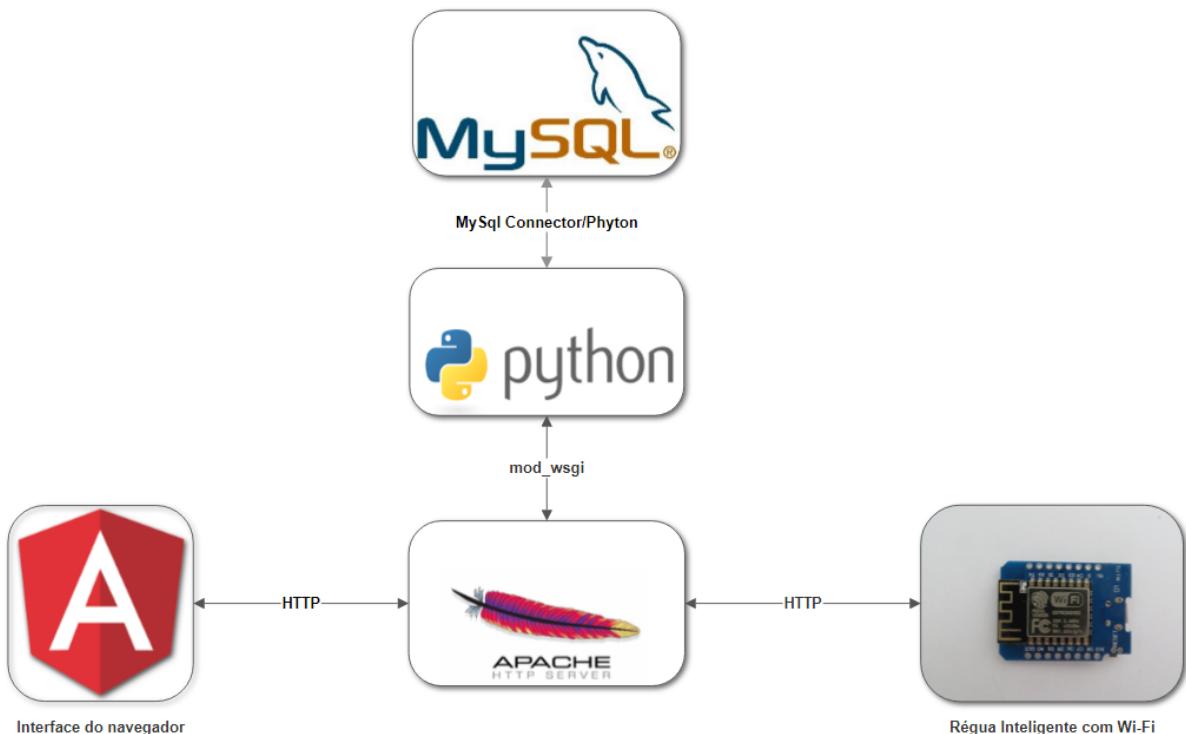


Figura 21 – Estrutura completa do *software*



# 6 Resultados

Neste capítulo serão indicados os resultados das validações obtidos da medição de potência média, corrente eficaz e fator de potência de diferentes tipos de carga.

## 6.1 Validações

Para avaliar o desempenho das medições feitas pela Régua Inteligente, diferentes tipos de cargas, resistivas e indutivas, foram selecionadas. Cada uma dessas cargas possuía um selo que indicava qual era o seu consumo de potência ativa, corrente eficaz e fator de potência. Com esses valores foi possível avaliar a qualidade das medições feitas pela Régua Inteligente.

Ainda para validar a qualidade das medições feitas pela régua, foi utilizado o *smart plug Voltson WiFi Outlet* da referência [Voltson... \(2018\)](#). De acordo com o manual deste produto, a sua precisão de medição pode variar entre  $\pm 1\%$  e  $\pm 6\%$ . Ele só mede a potência ativa das cargas.

Por dez minutos, em intervalos de cinco em cinco segundos, foram medidos os valores de potência ativa, corrente eficaz e fator de potência, tanto pela Régua Inteligente quanto pelo *Voltson WiFi Outlet*. Esses valores podem ser encontrados em tabelas que estão no arquivo do link da referência [Tabelas... \(2018a\)](#). Uma média dos valores medidos foi feita e os resultados encontrados foram resumidos nas tabelas 8 e 9.

Tabela 8 – Valores de potência ativa, corrente eficaz e fator de potência indicados e medidos

Cargas Testadas	P <sub>i</sub> (W)	I <sub>i</sub> (A)	F.P.i	P <sub>v</sub> (W)	P <sub>r</sub> (W)	I <sub>r</sub> (A)	F.P.r
Lâmpada LED Philips	8	0,075	0,92	9,000	7,733	0,07503	0,827
Lâmpada fluorescente FLC	15	0,216	0,55	15,821	12,955	0,21343	0,481
Lâmpada fluorescente Golden	15	0,191	0,50	14,750	11,909	0,18397	0,515
Lâmpada incandescente OSRAM	100	0,790	-	91,136	90,871	0,78904	0,919
Ventilador Mondial maxi power 30	50	0,390	-	47,563	45,283	0,40905	0,879
Ventilador Britânia C50 turbo 200W	200	1,500	-	114,000	101,567	1,02235	0,797
Secador de cabelo Taiff	1200	9,500	-	610,523	623,890	5,50658	0,944
TV Samsung	110	0,870	-	65,793	62,076	0,59632	0,834

Tabela 9 – Erro relativo das medições encontradas

Cargas Testadas	(P <sub>r</sub> -P <sub>i</sub> )*100/P <sub>i</sub>	(I <sub>r</sub> -I <sub>i</sub> )*100/I <sub>i</sub>	(F.P.r-F.P.i)*100/F.P.i	(P <sub>r</sub> -P <sub>v</sub> )*100/P <sub>v</sub>
Lâmpada LED Philips	-3,336	0,036	-10,152	-14,077
Lâmpada fluorescente FLC	-13,635	-1,188	-12,598	-18,119
Lâmpada fluorescente Golden	-20,604	-3,679	2,947	-19,259
Lâmpada incandescente OSRAM	-9,129	-0,121	-	-0,291
Ventilador Mondial maxi power 30	-9,433	4,885	-	-4,792
Ventilador Britânia C50 turbo 200W	-49,216	-31,843	-	-10,906
Secador de cabelo Taiff	-48,009	-42,036	-	2,189
TV Samsung	-43,567	-31,458	-	-5,650

Na tabela 8,  $P_i$ ,  $I_i$  e  $F.P.i$  são os valores de potência ativa, corrente eficaz e fator de potência indicados nos selos das cargas.  $P_v$  indica o valor de potência ativa medido pelo

*Voltson WiFi Outlet*. E  $P_r$ ,  $I_r$  e  $F.P._r$  são os valores de potência ativa, corrente eficaz e fator de potência medidos pela Régua Inteligente.

Na tabela 9, a primeira coluna indica o erro relativo entre a medição de potência ativa pela Régua Inteligente e a potência que o selo da carga indica, a segunda coluna é o erro relativo entre a medição de corrente eficaz pela Régua Inteligente e a corrente que o selo da carga indica, a terceira coluna é o erro relativo entre a medição do fator de potência pela Régua Inteligente e o fator de potência que o selo da carga indica e a última coluna é o erro relativo entre a medição de potência ativa pela Régua Inteligente e a potência medida pelo *Voltson WiFi Outlet*. Nas duas tabelas percebemos que alguns valores estão indicados por um traço, isso aconteceu pois essas cargas não indicavam em seus selos os valores de fator de potência.

Analizando a tabela de erros relativos, percebemos que o desempenho das medições da Régua Inteligente de potência ativa quando comparadas ao que foi especificado nos selos das cargas variou de 3% a 50%. Para três cargas esse valor foi alto, próximo de 50%. Mas se verificarmos a diferença entre os valores medidos para essas mesmas cargas entre a régua e o *Voltson WiFi Outlet*, os valores foram mais baixos, entre 5% e 10%. Isso pode indicar que os valores nos selos dessas cargas não representam o consumo de potência real delas. Os outros valores de potência ativa encontrados pela régua estão no pior caso próximos de 20%. Essa diferença pode ser explicada pela forma como a calibração das equações dos sensores foi feita. Essas calibrações foram feitas baseadas somente nas curvas encontradas em artigos e nos *datasheets* dos sensores, uma calibração mais precisa necessitaria da utilização de instrumentos de laboratório, como osciloscópios e multímetros.

Ainda analisando a tabela de erros relativos, percebemos que os valores de corrente eficaz medidos pela Régua Inteligente para as cinco primeiras cargas teve um desempenho muito satisfatório em relação ao indicado nos selos das cargas, variando de 0.04% até 5% aproximadamente. Os últimos três valores medidos variaram de 30% a 40% aproximadamente. A explicação para essa diferença pode ser explicada devido a apresentação equivocada dos valores de potência e corrente nos selos dessas cargas.

Os fatores de potência medidos pela régua variaram de 3% a 10% em relação aos valores indicados pelos selos das cargas. Os valores dessas medições poderiam ser melhor comparados caso instrumentos mais precisos de medição fossem utilizados.

Os resultados encontrados nas duas tabelas indicam que o primeiro caminho utilizado para o cálculo das medidas não foi ruim, tendo em mente os valores dos erros encontrados e a análise feita. Caso a calibração das equações dos sensores fosse melhor e instrumentos de medição mais precisos fossem utilizados para a comparação das medidas, melhores resultados poderiam ser alcançados.

## 7 Conclusão e trabalhos futuros

Nesse trabalho foi discutido a implementação do *hardware* e *software* de uma Régua Inteligente para o monitoramento local ou remoto do consumo de energia de equipamentos eletrodomésticos.

Primeiramente foi apresentada uma teoria que era necessária para o entendimento das partes que compunham o projeto da régua, assim como uma revisão de trabalhos relacionados ao tema aqui tratado.

Em seguida especificações do *hardware* e *software* foram feitas com base nas variáveis que se desejou medir, assim como a expectativa da utilização do servidor por usuários para poder monitorar o consumo de energia de seus dispositivos.

Com as especificações feitas, uma pesquisa no mercado foi feita para buscar e selecionar módulos de *hardware* que exercessem as funções necessárias, e também a busca e seleção de ferramentas de *software* para a construção do servidor e a interface de visualização das medições.

Por fim, validações de medições da Régua Inteligente para diferentes tipos de cargas foram apresentadas e discutidas.

Ainda resta construir as partes que compõem o servidor e validar o resultado da integração do *hardware* e *software* da Régua Inteligente. É de meu interesse continuar a implementação do servidor e das interfaces da Régua Inteligente, mas outras pessoas interessadas no assunto poderão fazer tal tarefa.

O desenvolvimento deste trabalho foi extremamente engradecedor pois pude trabalhar diversos aspectos do desenvolvimento de *hardware* e, principalmente, de *software*.



# Referências

- ABUBAKAR, I. et al. Calibration of zmpt101b voltage sensor module using polynomial regression for accurate load monitoring. v. 12, p. 1076–1084, Janeiro 2017. Citado na página 43.
- ADAFRUIT. 2018. <[https://github.com/adafruit/Adafruit\\_ADS1X15](https://github.com/adafruit/Adafruit_ADS1X15)>. Citado na página 59.
- ADAFRUIT Industries. 2017. <<https://cdn-shop.adafruit.com/datasheets/ads1015.pdf>>. Citado na página 45.
- ALLEGRO MicroSystems. 2017. <<https://www.allegromicro.com/~/media/files/datasheets/acs712-datasheet.ashx>>. Citado na página 44.
- ANEEL. 2018. <<http://www.aneel.gov.br/tensoes-nominais>>. Citado na página 33.
- ANGULAR. 2018. <<https://angular.io/>>. Citado na página 58.
- APACHE. 2018. <<https://www.apache.org/>>. Citado na página 52.
- ARDUINO. 2018. <<https://github.com/esp8266/Arduino/tree/master/libraries>>. Citado na página 59.
- BASSIL, Y. A comparative study on the performance of the top DBMS systems. *CoRR*, abs/1205.2889, 2012. Disponível em: <<http://arxiv.org/abs/1205.2889>>. Citado na página 55.
- DIDHE, L. V.; KULKARNI, S. V.; BAKAL, J. W. Remote monitoring and control of home devices using android platform. *International Journal of Advanced Research in Computer Science and Software Engineering*, v. 5, n. 3, p. 303–306, Março 2015. Citado na página 23.
- DOCS Apache. 2018. <<https://httpd.apache.org/>>. Citado na página 52.
- ESPRESSIF Systems. 2018. <[https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf)>. Citado na página 48.
- ETEKCITY. 2018. <[https://static.etekcity.com/files/manual/1.0\\_EM\\_HOHIHT32D\\_SmartWifiOutlet\\_2ndGen.pdf](https://static.etekcity.com/files/manual/1.0_EM_HOHIHT32D_SmartWifiOutlet_2ndGen.pdf)>. Citado 2 vezes nas páginas 37 e 44.
- GRAHAM Dumpleton. 2018. <<https://modwsgi.readthedocs.io/en/develop/>>. Citado na página 57.
- HAN, J. et al. Smart home energy management system including renewable energy based on zigbee and plc. *IEEE Transactions on Consumer Electronics*, v. 60, n. 2, p. 198–202, Maio 2014. Citado na página 22.
- HYELESIONTEK. 2018. <<https://pt.aliexpress.com/item/6pcs-DC-DC-Step-Down-Buck-Converter-Adjustable-4-5V-24V-to-5V-3A-97-5/32805900484.html?spm=a2g03.10010108.1000016.1.6af261baPJDsA0&isOrigTitle=true>>. Citado na página 49.

- IIS. 2018. <<https://www.iis.net/>>. Citado na página 53.
- INTERPLUS Industry. 2018. <[http://www.interplus-industry.fr/index.php?option=com\\_content&view=article&id=52&Itemid=173&lang=en](http://www.interplus-industry.fr/index.php?option=com_content&view=article&id=52&Itemid=173&lang=en)>. Citado na página 43.
- ISLAM, K. et al. Huge and real-time database systems: A comparative study and review for sql server 2016, oracle 12c & mysql 5.7 for personal computer. *Journal of Basic & Applied Sciences*, p. 481–490, 2016. Citado na página 55.
- JARROD. 2016. <<https://www.rootusers.com/linux-web-server-performance-benchmark-2016-results/>>. Citado na página 55.
- LIRON. 2017. <<https://iwf1.com/apache-vs-nginx-vs-node-js-and-what-it-means-about-the-performance-of-wordpress-vs-ghost>>. Citado na página 53.
- MUSLEH, A. S.; DEBOUZA, M.; FAROOK, M. Design and implementation of smart plug: An internet of things (iot) approach. *2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, p. 1–4, Nov 2017. Citado na página 24.
- MYSQL. 2018. <<https://www.mysql.com/>>. Citado na página 55.
- MYSQL Connector/Python. 2018. <<https://dev.mysql.com/doc/connector-python/en/>>. Citado na página 56.
- NETCRAFT. 2018. <<https://news.netcraft.com/archives/2018/06/13/june-2018-web-server-survey.html>>. Citado na página 52.
- NGINX. 2018. <<https://www.nginx.com/>>. Citado na página 52.
- NODE.JS. 2018. <<https://nodejs.org/en/>>. Citado na página 55.
- OLADIMEJI, S. et al. Development of a low-cost gsm-bluetooth home automation system. *International Journal of Intelligent Systems and Applications*, v. 8, p. 41–50, 08 2017. Citado na página 23.
- PYTHON. 2018. <<https://www.python.org/>>. Citado na página 56.
- S, A.; MAHADEVASWAMY, U. B. Design and development of a smart device for energy monitoring and control of domestic appliances: An android application. *International Journal of Image, Graphics and Signal Processing*, v. 10, n. 1, p. 36–46, Janeiro 2018. Citado na página 24.
- SAIKIA, A. et al. Comparative performance analysis of mysql and sql server relational database management systems in windows environment. *International Journal of Advanced Research in Computer and Communication Engineering*, p. 160–164, 2015. Citado na página 55.
- TABELAS de medição de cargas teste. 2018. <[https://github.com/fabianojardim/smart-plug/blob/master/valores\\_medicao\\_carga.xlsx](https://github.com/fabianojardim/smart-plug/blob/master/valores_medicao_carga.xlsx)>. Citado na página 61.
- TABELAS de potência de cargas. 2018. <[https://github.com/fabianojardim/smart-plug/blob/master/potencia\\_media\\_cargas.xlsx](https://github.com/fabianojardim/smart-plug/blob/master/potencia_media_cargas.xlsx)>. Citado 3 vezes nas páginas 5, 33 e 34.

TANENBAUM, D. J. W. A. S. *Computer Networks*. 5. ed. [S.l.]: Courier/Westford, 2011. ISBN 0132126958. Citado na página 31.

VOLTSON WiFi Outlet. 2018. <<https://www.etekcity.com/productcate/59/list>>. Citado na página 61.

YOUR Cee. 2018. <<https://pt.aliexpress.com/item/5-pcs-4-Channel-3-3V-to-5V-5V-to-3-3V-IIC-UART-SPI-TTL/32451889202.html?spm=a2g0s.9042311.0.0.2742b90aOnYRox>>. Citado na página 47.