

**UNIVERSIDADE FEDERAL DO PARANÁ**

ANDRÉ HENRIQUE VETERI BASTOS

**O USO DA INTERNET DAS COISAS EM  
PONTOS DE ENERGIA EM ÁREAS  
COMUNS**

CURITIBA - PARANÁ

2019

ANDRÉ HENRIQUE VETERI BASTOS

# **O USO DA INTERNET DAS COISAS EM PONTOS DE ENERGIA EM ÁREAS COMUNS**

Monografia apresentada à disciplina de Trabalho de Conclusão de Curso B como requisito parcial à conclusão do Curso de Engenharia Elétrica (Ênfase em Sistemas Eletrônicos Embarcados), Setor de Tecnologia, da Universidade Federal do Paraná.  
Orientador: Prof. Dr. José Carlos da Cunha

CURITIBA - PARANÁ  
2019

**Folha de aprovação**

ANDRÉ HENRIQUE VETERI BASTOS

**O USO DA INTERNET DAS COISAS EM PONTOS  
DE ENERGIA EM ÁREAS COMUNS**

Projeto apresentado como Trabalho de Conclusão de Curso, de Engenharia Elétrica com Ênfase em Sistemas Eletrônicos Embarcados, pela seguinte banca examinadora:

---

Orientador: José Carlos da Cunha

---

Rogers Demonti

---

Theoma Muriel Sanches Otobo

# LISTA DE ILUSTRAÇÕES

Figura 1 – Tabela de aplicações . . . . .	14
Figura 2 – Sequência para a realização da criptografia . . . . .	15
Figura 3 – <i>MySQL</i> com datas mais avançadas . . . . .	29
Figura 4 – Características dos excelentes aplicativos. . . . .	32
Figura 5 – O Photon . . . . .	35
Figura 6 – O circuito . . . . .	36
Figura 7 – Tensão de operação . . . . .	36
Figura 8 – Correntes de acionamento do relay . . . . .	37
Figura 9 – Retificador de meia onda com filtro capacitivo . . . . .	38
Figura 10 – Página inicial. . . . .	40
Figura 11 – Página de registro do usuário do aplicativo. . . . .	41
Figura 12 – Página inicial. . . . .	41
Figura 13 – Acesso incorreto. . . . .	42
Figura 14 – Página de escolha do ponto de acesso. . . . .	42
Figura 15 – Página de calendário. . . . .	43
Figura 16 – Janela de agendamento. . . . .	44
Figura 17 – Página todos os usuários cadastrados no dia. . . . .	44
Figura 18 – Detalhes do agendamento. . . . .	45
Figura 19 – Caso do não agendamento prévio. . . . .	45
Figura 20 – Página de roteamento. . . . .	46
Figura 21 – Página de ativamente do circuito. . . . .	46
Figura 22 – Caso do agendamento prévio. . . . .	47
Figura 23 – Página de custos. . . . .	48
Figura 24 – Circuito em funcionamento . . . . .	51
Figura 25 – Circuito desligado . . . . .	52
Figura 26 – Circuito retificador montado . . . . .	53
Figura 27 – Medição do sensor de corrente . . . . .	54

# SUMÁRIO

<b>1</b>	<b>RESUMO</b> . . . . .	<b>7</b>
<b>2</b>	<b>ABSTRACT</b> . . . . .	<b>8</b>
<b>3</b>	<b>INTRODUÇÃO</b> . . . . .	<b>9</b>
<b>3.1</b>	<b>Objetivos</b> . . . . .	<b>10</b>
3.1.1	Objetivo Geral . . . . .	10
3.1.2	Objetivos Específicos . . . . .	10
<b>4</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> . . . . .	<b>12</b>
<b>4.1</b>	<b>Conceitos utilizados</b> . . . . .	<b>12</b>
4.1.1	Internet of Things . . . . .	12
4.1.2	Criptografia . . . . .	13
4.1.2.1	Criptografia básica . . . . .	13
4.1.2.2	Assinatura digitais . . . . .	15
4.1.2.3	Testabilidade e segurança de software . . . . .	16
4.1.3	Comunicação via <i>Socket</i> . . . . .	17
4.1.3.1	Desenvolvimento profissional de software . . . . .	17
4.1.3.2	Engenharia de software . . . . .	18
4.1.3.2.1	Etapas para desenvolvimento sistemático . . . . .	18
4.1.3.2.2	Tipo de aplicações . . . . .	18
4.1.3.3	Ética na engenharia de software . . . . .	19
4.1.4	Processo de software . . . . .	19
4.1.4.1	Modelos de processo . . . . .	19
4.1.4.1.1	Modelo cascata . . . . .	20
4.1.4.1.2	Desenvolvimento incremental . . . . .	20
4.1.4.1.3	Engenharia de software orientada a reuso . . . . .	20
4.1.4.2	Atividades do processo . . . . .	21
4.1.4.2.1	Especificação . . . . .	21
4.1.4.2.2	Projeto e implementação . . . . .	21
4.1.5	Desenvolvimento ágil de software . . . . .	21
4.1.5.1	Métodos ágeis . . . . .	21
4.1.5.2	Desenvolvimento ágil e dirigido a planos . . . . .	22
4.1.6	Programação orientada a objetos . . . . .	22
4.1.6.1	Classes . . . . .	22
4.1.6.2	Métodos construtores . . . . .	22

4.1.6.3	Sobrecarga do construtor . . . . .	23
4.1.6.4	Encapsulamento . . . . .	23
<b>4.2</b>	<b>Linguagens utilizadas . . . . .</b>	<b>24</b>
4.2.1	HTML ( <i>Hypertext Markup Language</i> ) . . . . .	24
4.2.2	CSS ( <i>Cascading Style Sheets</i> ) . . . . .	24
4.2.3	Bootstrap . . . . .	25
4.2.4	PHP ( <i>Hypertext Preprocessor</i> ) . . . . .	25
4.2.5	Framework . . . . .	27
4.2.6	Laravel . . . . .	27
4.2.7	Banco de dados relacional . . . . .	27
4.2.7.1	MySQL . . . . .	28
4.2.7.1.1	Características principais do MySQL . . . . .	28
4.2.7.1.2	Sintaxe do MYSQL . . . . .	28
4.2.7.2	Android . . . . .	30
4.2.7.2.1	Código aberto . . . . .	30
4.2.7.2.2	Java . . . . .	30
4.2.7.2.3	Tela multitouch . . . . .	31
4.2.7.2.4	Aplicativos incorporados . . . . .	31
4.2.7.2.5	Programação orientada a objetos em relação ao Android . . . . .	31
4.2.7.2.6	Como destacar seu aplicativo Android . . . . .	31
4.2.7.2.7	IDE Android Developer Tools . . . . .	32
4.2.7.2.8	Componentes TextView e ImageView . . . . .	32
4.2.7.2.9	Recursos do aplicativo . . . . .	33
4.2.7.2.10	Acessibilidade . . . . .	33
4.2.7.2.11	Internacionalização . . . . .	33
4.2.7.2.12	Métodos de ciclo de vida de Activity . . . . .	33
4.2.8	WebView . . . . .	33
<b>4.3</b>	<b>Dispositivos utilizados . . . . .</b>	<b>34</b>
4.3.1	Photon . . . . .	34
4.3.2	Módulo relé . . . . .	35
4.3.3	Relé . . . . .	35
4.3.4	Medidor de corrente não invasivo . . . . .	36
4.3.5	Retificador de meia onda com filtro capacitivo . . . . .	37
<b>5</b>	<b>DESENVOLVIMENTO DO PROJETO . . . . .</b>	<b>39</b>
<b>5.1</b>	<b>Front-end . . . . .</b>	<b>39</b>
5.1.1	Visão Geral . . . . .	39
5.1.2	Área de login . . . . .	40
5.1.3	Área de registro . . . . .	40
5.1.4	Página Inicial . . . . .	41

5.1.5	Página de Agendamento . . . . .	42
5.1.5.1	Página de escolha do ponto de acesso . . . . .	42
5.1.5.2	Página de agendamento . . . . .	43
5.1.6	Página de Ativamento . . . . .	44
5.1.6.1	Tentativas de acesso . . . . .	45
5.1.6.2	Página de acesso . . . . .	45
5.1.6.3	Página desativar . . . . .	46
5.1.7	Página de Custos . . . . .	47
<b>5.2</b>	<b>Back-end . . . . .</b>	<b>47</b>
<b>5.3</b>	<b>Protótipo . . . . .</b>	<b>50</b>
<b>6</b>	<b>RESULTADOS E DISCUSSÕES . . . . .</b>	<b>55</b>
<b>7</b>	<b>CONCLUSÃO . . . . .</b>	<b>57</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>59</b>

# 1 RESUMO

Em busca do aumento das vendas, empreiteiros focaram em construção de empreendimentos menores do que as construções pré-existentes. A redução de espaço gerou áreas de uso compartilhado entre os moradores, as quais são causa de muitas discussões entre os condôminos pela utilização ou má utilização dos serviços nela encontrados. Para apresentar uma solução para essa problemática está desenvolvendo-se este trabalho, idealizando corroborar com o problema de compartilhamento dos serviços, propondo-se a criação de um sistema que tem como destino a organização do tempo de utilização dos componentes de uso compartilhado pelos moradores, buscando a individualização dos componentes e o registro de seus respectivos consumos. Para desenvolver o aplicativo e o site programou-se utilizando as linguagens c (microcontrolador), html, css, php (site), sql (site e aplicativo) e java (aplicativo). Também foi desenvolvido um projeto onde através do uso de lâmpadas, representando as saídas, microcontrolador, como executor do chaveamento dos relés, sensor de corrente, responsável pelo registro da utilização de corrente, será mostrada a funcionalidade do sistema.

**Palavras-chave:** redução de espaço; compartilhamento dos equipamentos; agendamento dos equipamentos; individualização dos componentes.

## 2 ABSTRACT

Looking of increased your sales, builders have built your build smaller than the old. This reduce became private area in share area between the owners. This shared area has cause of many discussions among the owners for the use or misuse of the services found there. Presenting a solution to this problem, this work is being developed, aiming to corroborate the problem of sharing services, proposing the creation of system that aims to organize the use time of components used by residents, seeking the individualization of the components and the registration of their respective consumptions. Developing the application and the site was programmed using the languages c (microcontroller), html, css, php (site), sql (site and application) and java (application). A project was also developed where using light bulbs, representing the outputs, microcontroller, as the relay switching performer, current sensor, responsible for recording the current utilization, will show the functionality of system.

**Keywords:** area reduction; equipment sharing; equipment schedules; individualization of components.

### 3 INTRODUÇÃO

Muitas pessoas são atraídas para grandes centros econômicos devido às muitas oportunidades e facilidades que por eles são ofertadas, como uma vasta gama de ofertas de trabalho, centros de estudo e de atendimento à saúde, além de praticidades, como o atendimento 24 horas, muitas vezes distanciando-se de suas residências situadas em pequenas cidades. Esse “êxodo diário” realizado por muitas pessoas dos pequenos centros para grandes centros econômicos, torna-se muito desgastante, visto a rotina atribulada que muitas pessoas levam, o que acaba forçando a procura por moradia nos grandes centros, buscando assim uma melhor qualidade de vida. Mas devido aos altos custos imobiliários e de vida, muitas dessas pessoas não conseguem se estabelecer nos grandes centros. Com isso muitas pessoas passaram a buscar moradia nas proximidades dos grandes centros, criando uma alternativa menos custosa, criando o que hoje conhecemos por “cidades-dormitório”.

Porém com o tempo, as próprias cidades-dormitório cresceram e com isso o espaço tornou-se cada vez mais preenchido, tanto nas cidades dormitório, como nos grandes centros. Vendo no aumento populacional, uma grande oportunidade de crescimento de vendas, muitos empreiteiros apostaram em residências de tamanho reduzido, os chamados apartamentos, imóveis que apresentam um melhor aproveitamento do terreno empregado. Taxados muitas vezes de “apertamentos”, o aproveitamento dos espaços essenciais de uma moradia é a prioridade no projeto, porém como o espaço reduzido tem-se que algumas áreas foram remanejadas dentro de muitos projetos. Áreas como cozinha e copa, lavanderia, garagem, por exemplo, tiveram suas características convencionais remodeladas, tornando-se, em muitos projetos, áreas de uso compartilhado entre os condôminos. Contudo, o compartilhamento de áreas e de recursos é algo complicado de ser efetivado entre muitos condôminos, em todos os lugares. Problemas relacionados ao uso de áreas compartilhadas, ao má utilização de aparelhos de uso compartilhado e cobranças indevidas complicam a relação dos usuários, sendo esse um grande problema enfrentado por muitos síndicos.

Além dos problemas de uso comentados, tem-se que com a evolução da tecnologia novos problemas surgirão, como o uso de tomadas compartilhadas para carregamento de veículos elétricos. O problema muitas vezes está relacionado à falta de controle de quem está utilizando o bem, saber se as condições de uso estão sendo seguidas, ter uma cobrança justa entre todos os moradores e a falta de responsabilidade que muitas pessoas apresentam em não assumir seus atos, ou em tenta-los corrigi-los.

É com base nessa problemática e na evolução da tecnologia que pretende-se criar um site para a liberação do uso de dispositivos eletrônicos que estão em áreas de uso compartilhado, por meio do desbloqueio da tomada na qual o bem está conectado, onde

tem-se a motivação de além da liberação, ter também o controle do usuário, do tempo de uso, e da quantidade de energia que por esse bem consumida durante o tempo de utilização. Com a forte utilização de dispositivos eletrônicos, como celulares e *tablets*, também tem-se o desenvolvimento de uma adaptação para plataforma Android, que apresenta as mesmas funcionalidades que o site, sendo que os dois possuem um sistema interligado, logo o que for desbloqueado tanto pelo site, quanto pelo aplicativo, é visto no mesmo sistema pelo administrador.

Indo além da proposta citada anteriormente, foi pensando em desenvolver um sistema de agendamento para os bens de uso compartilhado, onde o condômino tem a possibilidade de agendar a utilização, para um momento de tempo livre, onde ele não terá que se preocupar com o caso de tentar lavar sua roupa em seu tempo livre e se deparar com todas as máquinas sendo utilizadas por outros moradores, por exemplo. Porém o agendamento também tem o propósito de servir como controle de pessoas que utilizaram o bem, de agendamentos que não foram efetivamente utilizados, podendo também servir como controle de sinistros.

## 3.1 OBJETIVOS

Durante esse tópico serão discorridos sobre os objetivos gerais e específicos que motivaram a construção desse projeto durante o trabalho de conclusão de curso, apresentando a interligação de vários conteúdos que foram vistos durante todo o curso de engenharia elétrica.

### 3.1.1 OBJETIVO GERAL

O objetivo geral deste trabalho consiste no desenvolvimento de um sistema para o desbloqueio, controle e medição do consumo de um ponto de energia em área de uso comum. Este projeto tem como alvo o público geral, mas seu uso pode ser destacado em postos de abastecimentos de energia de veículos elétricos, condomínios residenciais e áreas externas de uso comum. É um dispositivo que faz a individualização de ponto de energia, visando uma cobrança individualizada, pois atualmente nos condomínios é feito o rateio da energia entre os condôminos, com o projeto que está sendo proposto, cada cliente pagará pelo que realmente usou. Além disso, com a vinda de veículos elétricos a cobrança do quilowatt-hora individualizado tornar-se-á ainda mais importante.

### 3.1.2 OBJETIVOS ESPECÍFICOS

Os principais objetivos específicos deste trabalho são:

- criar um site responsivo para o usuário;

- desenvolver página inicial para login e senha do usuário;
- fornecer página para inscrição do usuário;
- utilizar arquitetura MD5 para proteção das senhas dos usuários;
- fazer a conexão entre o site e o microcontrolador Photon;
- fazer a conexão entre o site e o banco de dados;
- disponibilizar agendamento para usuários;
- realizar o controle de acionamento dos dispositivos;
- realizar a medição individualizada para cada usuário;
- desenvolver aplicação Android do sistema para o usuário acessar via aplicativo.

Logo no decorrer deste trabalho, ter-se-á o planejamento do projeto que será desenvolvido, discorrendo sobre as linguagens que serão utilizadas no site e aplicativo, sobre os componentes que serão utilizados do protótipo, sendo essa explicação teórica e apresentada uma justificativa para as suas escolhas. Posteriormente, será apresentada a programação do site e do aplicativo, tendo cada uma das etapas do processo de programação sendo explicadas, tendo a amostragem da junção das linguagens, de forma objetiva. E por fim, para a consolidação da ideia que esse projeto está querendo empregar, tem-se o feitiço e o funcionamento do protótipo, o qual tem como objetivo acionamento e o desbloqueio dos relés, que liberarão o acendimento das lâmpadas, sendo esse acionamento, por meio do site ou aplicativo.

# 4 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será discorrido sobre os conceitos estruturais que auxiliaram no desenvolvimento do site, assim como as linguagens que foram utilizadas, tanto no desenvolvimento do site, quanto no desenvolvimento do aplicativo. Também foi comentado sobre os dispositivos que foram utilizados para a construção do protótipo que será utilizado na comprovação de funcionamento do site criado, assim como a conexão existente entre o protótipo e o site.

Com base no que foi dito, tem-se que esse capítulo foi estruturado nos seguintes tópicos: *conceitos utilizados*, *linguagens utilizadas* e *dispositivos utilizados*. Essa estruturação foi escolhida, para um melhor entendimento da fundamentação teórica que foi necessária para o desenvolvimento desse trabalho. Dessa forma o intuito de apresentação dos tópicos utilizados torna-se mais teórica do que explicativa em relação ao projeto, sendo que isso será feito no desenvolvimento do próximo capítulo.

## 4.1 CONCEITOS UTILIZADOS

Tem-se que para fundamentar a estrutura de desenvolvimento do site, foi necessária a pesquisa em relação a alguns temas estruturantes da lógica programacional, em relação a alguns tipos de métodos que são necessários para o desenvolvimento do projeto que aqui é proposto. Temas como *testabilidade de Software*, *programação orientada ao objeto*, *comunicação via Socket*, e *processos de software*, foram estudados e utilizados no desenvolvimento do *back-end* do servidor que responde ao site.

Tem-se que também foi utilizou-se os conceitos de *Internet of Things* - IoT, solucionar a problematização proposta na introdução como justificativa do projeto aqui desenvolvido. Com a evolução do conceito da IoT, cada vez mais ela é atrelada à solução dos problemas ocasionados por vias tecnológicas, com base nisso, e em conhecimentos prévios do conceito da IoT, é que apoiou-se em sua utilização nesse projeto.

### 4.1.1 INTERNET OF THINGS

Com a evolução da tecnologia e com a necessidade de comunicação entre equipamentos tecnológicos surge a *internet of things* (IoT), ou comumente chamada de internet das coisas, a qual relaciona o uso de equipamentos que de alguma forma estão conectados ou com a internet, ou entre si. Além da conexão entre equipamentos, a IoT tem como objetivo a praticidade de seu uso, áreas como segurança, saúde, educação são exemplos de áreas onde pode ser encontrada a aplicabilidade da IoT. Exemplos mais específicos da utilização da IoT podem ser encontrados na interligação de um sistema de GPS (*Global*

*Posicion System*) com uma frota de veículos de uma empresa, a qual acompanha desde o deslocamento do veículo até o transporte de seus produtos, outro exemplo que pode ser dado é o uso de micros sensores utilizados no monitoramento de pacientes pelos médicos, sendo que seus relatórios podem ser vistos no próprio hospital ou até mesmo fora dele.

A IoT permite a coordenação inteligente de equipamentos controlando diversas atividades, ou seja, a IoT controla atividades como: monitoramento de câmeras e sensores para a gestão de espaços, assim como processos produtivos. A IoT trata tanto da coleta e processamento de dados, como também da conexão de dados inteligentes. Um artigo do Banco do Brasil, explicando de uma forma mais simplista a internet das coisas, nos diz o seguinte em relação a alguns tipos de componentes atrelados a IoT:

O ecossistema da IoT envolve diferentes agentes e processos, como módulos inteligentes (processadores, memórias), objetos inteligentes (eletrodomésticos, carros, equipamentos de automação em fábricas), serviços de conectividade (prestação do acesso à internet ou redes privadas que conectam esses dispositivos), habilitadores (sistemas de controle, coleta e processamento dos dados e comandos envolvendo os objetos), integradores (sistemas que combinam aplicações, processos e dispositivos) e provedores dos serviços de IoT. (BRASIL, 2019, p.1)

A figura 1 na sequência do texto mostra uma matriz de dados relativa ao funcionamento da IoT, relacionando diversos casos de uso com ambientes de aplicação, trabalhando em si, casos específicos. Note que ao selecionar um determinado tipo de caso, o leitor consegue relacionar quais ambientes de aplicação no qual a IoT está relacionada (BRASIL, 2019).

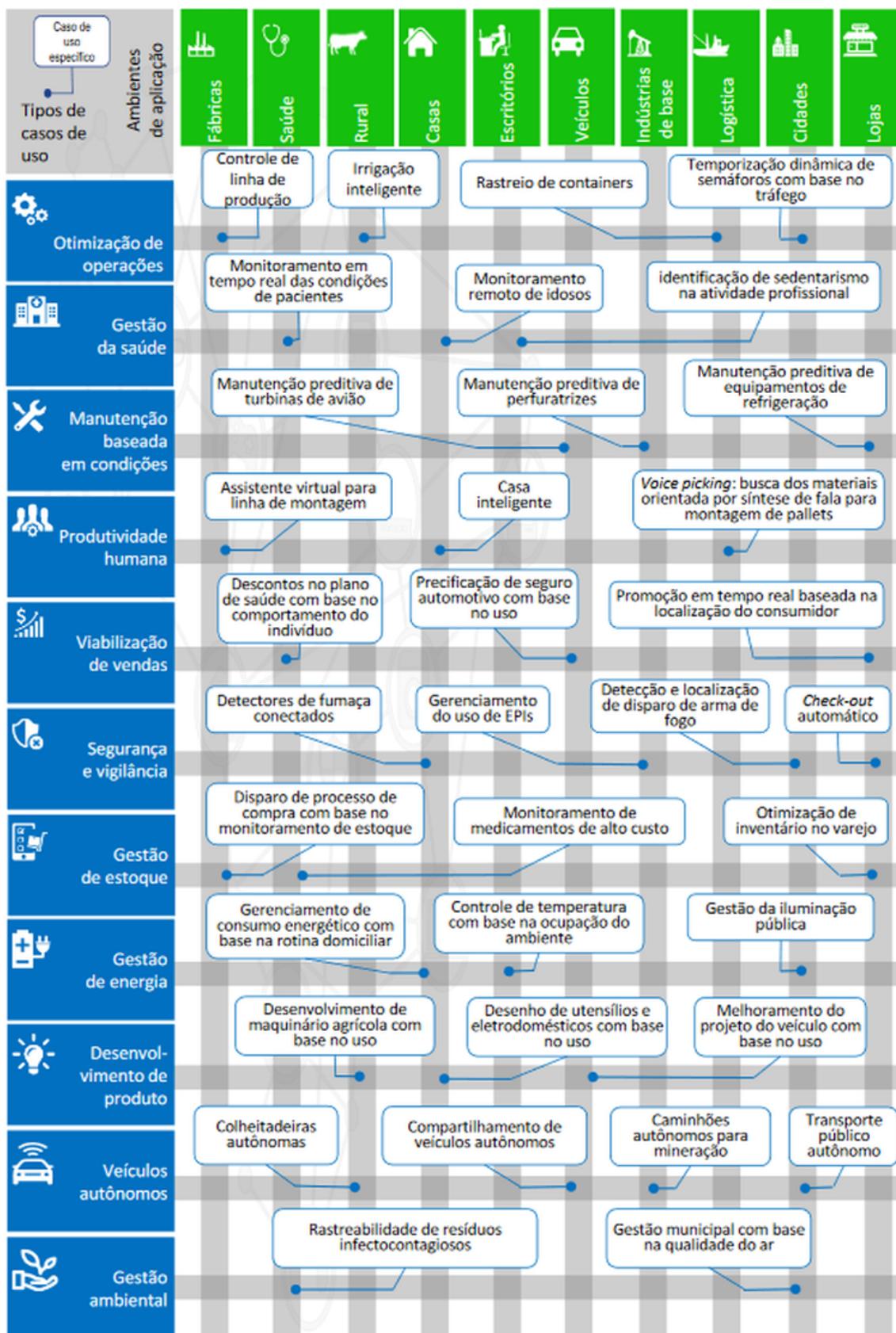
Ao selecionar o ambiente “casa”, a figura nos trás exatamente a área de aplicabilidade que está sendo buscada nesse trabalho, que é trabalhar com o projeto de “casa inteligente”, “gerenciamento de consumo energético com base na rotina familiar” e “deseenho de utensílios e eletrodomésticos com base no uso”. Por mais que esse trabalho está sendo relacionado a ambientes de uso compartilhado em condomínios e apartamentos, tem-se que sua proposta se enquadra nesse tipo de coleta de dados, visto que com o agendamento da utilização do aparelho, a partir do uso do site, tem-se o relatório de utilização, de consumo, de duração de utilização, por parte de cada um dos usuários. Dessa forma pode-se mapear picos e declínios de utilização, rentabilidade do aparelho utilizado, podendo inclusive realizar um comparativo entre os pontos de utilização (BRASIL, 2019).

## 4.1.2 CRIPTOGRAFIA

### 4.1.2.1 CRIPTOGRAFIA BÁSICA

Segundo o autor Tanenbaum (2010) o objetivo da criptografia é transformar um texto puro em um texto cifrado, para que assim apenas a pessoa autorizada possa reconvertê-lo para o texto puro. Ele argumenta que é recomendado sempre disponibilizar os dados de forma pública e utilizar a criptografia para tornar os dados incompreensíveis. Um erro

Figura 1 – Tabela de aplicações



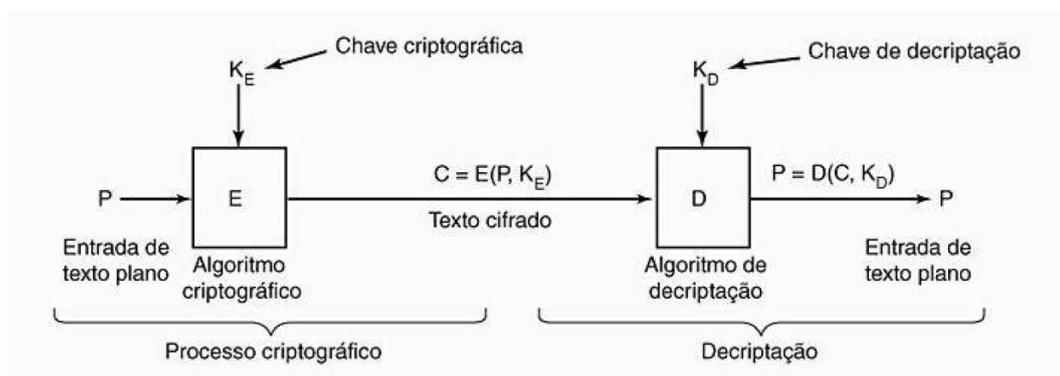
Fonte: (BRASIL, 2019)

comum é manter os dados em segredo, o que gera uma falsa segurança, assim implementando a segurança por obscuridade, que normalmente é realizada por amadores, e até mesmo algumas empresas relativamente grandes utilizam desse subterfúgio. Contudo, esse método acaba por revelar-se muito mais fraco no sentido de segurança dos dados.

O princípio de *Keckhoffs*, em síntese, pede que para a criptografia deve-se usar texto puro mais o algoritmo de criptografia e a chave de criptográfica, sempre disponibilizando os algoritmos para o público e apenas as chaves devem ser secretas. Desse modo a própria comunidade verifica se o algoritmo possui erros que comprometa a segurança e auxiliam na melhora do algoritmo (TANENBAUM, 2010).

Para realizar a criptografia inicia-se com a entrada de texto, após passa pelo algoritmo de criptografia para então ser enviado como texto cifrado. Depois de realizar esse processo o texto é decifrado e transforma-se no texto original novamente. A figura 2, na sequência mostra o funcionamento da criptografia (TANENBAUM, 2010).

Figura 2 – Sequência para a realização da criptografia



Fonte: (TANENBAUM, 2010)

#### 4.1.2.2 ASSINATURA DIGITAIS

Por meio da assinatura digital é possível comprovar a veracidade da operação, no caso que o cliente permitiu a operação. Um exemplo que Tanenbaum (2010) usa é:

Suponha que um cliente de um banco instrua o banco a comprar algumas ações para ele, enviando uma mensagem pelo correio eletrônico. Uma hora depois de a ordem ter sido enviada e executada, as ações despencam. O cliente então nega que tenha enviado a mensagem eletrônica. O banco poderia produzir uma mensagem eletrônica, é claro, mas o cliente poderia alegar que o banco a forjou para obter uma comissão. Como um juiz saberia quem dizendo a verdade?(TANENBAUM, 2010, p.134)

A resposta para esta pergunta é feita por meio da assinatura digital. Ela é uma maneira de submeter os arquivos a algoritmos de síntese em sentido único, que seja praticamente impossível de ser invertido. Essa função produz um resumo *hash* de tamanho

não variável e de forma isolada do original, podendo ter tamanho maior ou menor que o original. Atualmente as funções mais usadas são (*secure hash algorithm* – algoritmo hash seguro 1) SHA-1. Ela produz 20 bytes de resultado, a outra função é (*message digest 5* – compêndio de mensagens 5) MD5 que produz 16 bytes de resultado. Atualmente usam-se as versões atuais do SHA-1, que são SHA-256 e o SHA-512. Seus resultados são de 32 bytes e 64 bytes (TANENBAUM, 2010).

Após esses passos, faz-se a utilização de cifragem de chave pública, assim o usuário do arquivo faz uso de sua chave privada no resumo, para, dessa forma, resgatar (resumo). Isso é chamado de bloco de assinatura e é anexada à documentação e envia ao receptor. Após esse processo, a aplicação D mostrada na figura 2 é requerida para decriptar o resumo (TANENBAUM, 2010).

Em seguida, quando o arquivo e o resumo já estão disponíveis, é realizado por meio dos métodos MD5 ou SHA, esse já foi escolhido de maneira antecipada. Assim, o receptor submete a chave pública do emitente as assinatura de bloco para obter o  $E(D(\text{resumo}))$ . Caso o resumo obtido pelo cálculo não seja igual ao que contém no bloco de assinatura, o algoritmo entende que foi alterado o arquivo. Esse método é diferenciado, pois realiza também a cifragem da chave pública nos dados (TANENBAUM, 2010).

O algoritmo utilizado nesse processo é o RSA (Rivest, Shamir, Adleman) que representa o nome dos inventores desse algoritmo, que apresenta a propriedade de assinatura. Por esse motivo, não importa a sequência de aplicações, pelo fato são possível usar as assinaturas de usuários que publicam sua chave pública na rede. Contudo, outros usuários, por medo de invasão, não o fazem. Para contornar esse problema é utilizado um método de anexar as mensagem ao emissor do certificado, normalmente contém o nome do usuário e a chave pública digitalmente (TANENBAUM, 2010).

Outro método utilizado é contratar uma empresa que valide certificados, esse método é denominado (autoridade certificadora) AC. Para validar essa chave é usado o método da infraestrutura de chave pública para fazer a verificação se a chave é realmente verdadeira. Na atualidade, nos navegadores isso é resolvido de maneira arcaica, em que todos vêm pré-carregados com cerca de 40 acessos de chave pública (TANENBAUM, 2010).

#### 4.1.2.3 TESTABILIDADE E SEGURANÇA DE SOFTWARE

Na situação atual, praticamente a maioria das áreas tem a necessidade de utilizar software para controlar seus sistemas. Por esse motivo os softwares que não são restringidos por meios matérias, nem pelas leis físicas. A Engenharia de software torna-se importante nesse sentido, pois disponibiliza diversas soluções, algumas mais complexas e sofisticadas, outras rápidas e eficazes (SOMMERVILLE, 2013).

Para realizar a engenharia de software é necessário realizar a testabilidade de software, o que para um sistema de grande porte é extremamente complexa e custosa

(SOMMERVILLE, 2013).

### 4.1.3 COMUNICAÇÃO VIA SOCKET

A comunicação via *Socket* é o principal recurso presente nas diversas linguagens de programação como: lua, Java e PHP. Ela possibilita a comunicação via TCP/IP entre um computador servidor e vários computadores clientes (SOMMERVILLE, 2013).

De maneira genérica e introdutória, um *socket* essencialmente é uma conexão de dados transparente entre dois computadores em uma rede, em que um dos computadores é chamado servidor que abre um *socket* e presta atenção às conexões de outros computadores na mesma rede. Já os outros computadores denominam-se clientes, pois estes chamam o *socket* servidor para iniciar a conexão, na qual é possível haver uma comunicação de envio e resposta de informação (SOMMERVILLE, 2013).

Assim, os computadores em rede direcionam os *streams* de dados recebidos da rede para programas receptores específicos, associando cada programa a um número diferente, que é a porta do programa. Da mesma forma, quando o tráfego de saída é gerado, o programa de origem recebe um número de porta para a transação (SOMMERVILLE, 2013).

Determinados números de porta são reservados no TCP/IP para protocolos específicos, por exemplo, 80 para HTTP. Para que seja efetuada a comunicação em rede através de *socket*, é necessário ter conhecimento do número de IP ou *HOST* do computador e o número de porta do aplicativo ao qual se quer realizar a conexão (SOMMERVILLE, 2013).

O endereço IP identifica uma máquina específica na Internet e o número de porta é a maneira de diferenciar os processos que estão sendo executados no mesmo computador. É exatamente a combinação do IP com o número de porta do aplicativo que se denomina *socket* (SOMMERVILLE, 2013).

#### 4.1.3.1 DESENVOLVIMENTO PROFISSIONAL DE SOFTWARE

Em software profissional normalmente o resultado final será alterado durante um grande período. Por isso, dificilmente apenas um programador executará toda a programação, dessa forma é necessário que o profissional saiba trabalhar em equipe. Os integrantes da equipe precisam realizar a documentação e um manual para o usuário para facilitar futuras implementações (SOMMERVILLE, 2013).

Como os softwares são um produto, eles podem ser de dois tipos: Produtos genéricos e sob encomenda. O produto genérico: podem ser sistemas *stand-alone*, sistemas disponíveis para o público em geral. Produtos sob encomenda: são sistema encomendados e destinados a um cliente em específico. Por exemplo, um sistema de controle de dispositivos eletrônicos ou sistema de controle de tráfego aéreo (SOMMERVILLE, 2013).

Uma situação comum é a migração de sistema quando um produto foi desenvolvido para uma determinada empresa, e após um tempo passa para o grande público. O inverso também é possível, quando inicialmente o software é desenvolvido para um produto e uma empresa adquire os privilégios de tal serviço. Assim tornando o que era genérico em específico ou o específico em genérico (SOMMERVILLE, 2013).

Outro aspecto é a necessidade que o software deve atender, seu intuito pode ser mais seguro, mais rápido ou mais confiável. Para cada tipo deve ser usada uma abordagem diferente, para assim alcançar o objetivo do produto. Normalmente para sistema que precisam de sigilo é necessário mais segurança. Para sistemas de tempo real e necessário uma resposta mais rápida, já para sistema de precisão é necessário que sejam exatos e tenham uma grande confiabilidade (SOMMERVILLE, 2013).

#### 4.1.3.2 ENGENHARIA DE SOFTWARE

A prioridade é necessariamente a produção de software, assim são especificados, desde os estágios iniciais até a sua finalização e posterior manutenção, quando o sistema já está em pleno uso (SOMMERVILLE, 2013).

A finalidade desse processo de engenharia de software é realizar processos técnicos para o desenvolvimento de software, gerenciamento de projetos em software e, por fim, desenvolvimento de ferramentas, teorias e métodos para auxiliar na fabricação de produtos tecnológicos (SOMMERVILLE, 2013).

##### 4.1.3.2.1 ETAPAS PARA DESENVOLVIMENTO SISTEMÁTICO

O primeiro passo é especificar o software, assim é realizada a definição de como será produzido e possíveis restrições em sua operação. Na sequência é feito o desenvolvimento e programação e, após, a validação que é a testabilidade do software. A finalização se dá com a manutenção e a evolução, para que se adéque aos novos requisitos pedidos pelos clientes, visando a uma maior duração de vida para o sistema (SOMMERVILLE, 2013).

Algumas exigências para realizar a programação é haver heterogeneidade, pois dessa forma o sistema será flexível e integrado ao sistema já existente no mercado. É de fácil adaptabilidade, para não existir paradas produtivas na indústria, além de seguro e confiável, para resistir a infestações, a fim de que o consumidor tenha segurança ao adquirir o produto para sua implementação (SOMMERVILLE, 2013).

##### 4.1.3.2.2 TIPO DE APLICAÇÕES

Na sequência serão apresentados alguns tópicos sobre tipo de aplicações disponíveis para cada tipo de sistema:

- *stand-alone*, com foco em executar as rotinas do computador local;

- baseadas em transações: são operação que acontecem em um computador remoto, a exemplo de aplicações *web*;
- sistemas de controle embutido, softwares que fazem o controle de hardware;
- sistema de lotes, normalmente sistemas corporativos que processam em lote de grande porte;
- sistemas de entretenimento: sistema de comunicação entre usuários;
- modelagem e simulação: sistema utilizado em aplicação para simular eventos;
- coleta de dados: sistema que utiliza um conjunto de dados para processamento e posterior tomada de decisão;
- sistemas de sistemas: composição de sistema que gerem sistemas.

Esses sistemas são desenvolvidos em processo de compreensão e gerenciamento com a ideia definida de começo, meio e fim, assim desenvolve-se um alto grau de confiabilidade e desempenho. Dessa forma, o sistema atende às especificações do cliente da melhor forma possível (SOMMERVILLE, 2013).

#### 4.1.3.3 ÉTICA NA ENGENHARIA DE SOFTWARE

Segundo Sommerville (2013), como a engenharia de software é criada dentro de um *framework* social, que normalmente limita a liberdade criativa das pessoas desse ramo, o carácter moral e o carácter ético são primordiais para o desenvolvimento das atitudes do profissional responsável. Sendo assim, devem-se manter padrões de honestidade e integridade elevados.

Além disso, o profissional deve ser orientado para não se utilizar de conhecimento para denegrir alguém, nem comportar-se de maneira que não seja confidencial, competente e com respeito ao direito de propriedade legal e intelectual. O código de conduta é publicado por organizações de renome como o IEEE e a ACM (SOMMERVILLE, 2013).

#### 4.1.4 PROCESSO DE SOFTWARE

O processo de software é a soma de atividades que levam à produção de um produto e devem incluir especificação, implementação, validação e evolução. Nesse tópico serão trabalhados os modelos de processo e suas atividades.

##### 4.1.4.1 MODELOS DE PROCESSO

Nesse tópico serão apresentados alguns dos modelos que formam métodos de construção de software.

#### 4.1.4.1.1 MODELO CASCATA

Esse modelo é caracterizado pela separação das fases, são elas a especificação de requisitos, projeto de software, implementação e testes. Por esse encadeamento entre uma fase e outra, esse modelo é chamado de cascata. As fases são:

- análise e definição de requisitos;
- projeto de sistema de software;
- implementação de testes unitários;
- integração de teste de sistema;
- operação e manutenção.

Teoricamente as fases devem ser finalizadas em sequência. Contudo, o que normalmente acontece é que as fases se sobrepõem e são auto-abastecidas pelas outras (SOMMERVILLE, 2013).

#### 4.1.4.1.2 DESENVOLVIMENTO INCREMENTAL

Nesse tipo de desenvolvimento são intercaladas as atividades de validação, desenvolvimento e especificação como uma série de incrementos. Assim, a cada nova versão, são adicionadas novas funcionalidades para aumentar a base inicial. Isso se dá por meio de upgrade de usuário e sugestões de melhorias para assim se chegar ao resultado esperado.

De certa forma existe a simultaneidade de acontecimentos de especificação, desenvolvimento e validação, ao mesmo tempo em que o usuário possui versões para teste, para assim obter o modelo finalizado. As vantagens para esse tipo de abordagem são:

- menos documentação;
- melhor relacionamento para com o cliente;
- facilita o *feedback*;
- correções menos complexas.

A desvantagem para esse tipo de abordagem é a não visibilidade da estrutura do sistema que se degrada com o incremento de novas funcionalidades (SOMMERVILLE, 2013).

#### 4.1.4.1.3 ENGENHARIA DE SOFTWARE ORIENTADA A REUSO

Segundo Sommerville (2013), como esta abordagem utiliza a ideia de elementos reusáveis o processo concentra a integração de componentes no sistema já existente ao invés de partir do zero. Dessa maneira, pode ser feito o aproveitamento dos códigos ou processos semelhantes, realizando as modificações necessárias para incorporá-los ao sistema. Os estágios do processo são:

- análise de componentes;

- modificação de requisitos;
- projeto do sistema de reuso;
- desenvolvimento de integração.

#### 4.1.4.2 ATIVIDADES DO PROCESSO

Nessa etapa tem-se um enfoque de planejamento do site, pensando num escopo facilitador de seu desenvolvimento.

##### 4.1.4.2.1 ESPECIFICAÇÃO

Nesse processo de definição e compreensão para com os serviços requisitados, busca-se não ter erros ou solucioná-los de maneira mais rápida possível (SOMMERVILLE, 2013). As etapas dessa atividades são:

- estudo de viabilidade;
- eliminação de análise dos requisitos;
- especificação dos requisitos;
- validação dos requisitos.

##### 4.1.4.2.2 PROJETO E IMPLEMENTAÇÃO

Nesta parte do processo a conversão e a especificação em sistema executável, envolve a definição de processo do projeto e a compilação de linha digitável. Contudo, pode já ser realizados os refinamentos das especificidades para melhor desempenho do software (SOMMERVILLE, 2013).

#### 4.1.5 DESENVOLVIMENTO ÁGIL DE SOFTWARE

Atualmente, com o desenvolvimento de tecnologias em constante alteração, a obrigação de alterações rápidas e confiáveis é uma prioridade. Para isso foi desenvolvido pela IBM o método incremental, dessa maneira, a especificação, projeto e implementação são, de certa, forma integrados e intercalados. O objetivo é o de reduzir a documentação e trazer uma nova versão para propor alterações ou novas funcionalidades para atender a demanda dos clientes (SOMMERVILLE, 2013).

##### 4.1.5.1 MÉTODOS ÁGEIS

Para realizar desenvolvimento ágil de software Sommerville (2013) informa que foi implementado uma equipe específica para o desenvolvimento em programação. Já a outra equipe para a parte de documentação. Dessa forma as preferências para a execução do desenvolvimento são:

- indivíduos e interação do processo e ferramentas;

- software em funcionamento do que documentação abrangente;
- colaboração do cliente para negociação de contrato;
- resposta a mudança do que seguir um plano.

#### 4.1.5.2 DESENVOLVIMENTO ÁGIL E DIRIGIDO A PLANOS

A abordagem feita para esse tipo de desenvolvimento, no sentido de atividades documentais formais, tem o objetivo de estabelecer uma relação entre os estágios e os processos. Já na abordagem ágil, todas ocorrem em simultaneidade de atividades, assim os programas são desenvolvidos em conjunto. Para esse tipo de aplicação alguns exemplos são: grandes sistemas que exigem vários desenvolvedores, sistemas em que é necessário uma grande análise para modificação ou implementação, sistemas de vida longa, isto é, quando os programadores provavelmente não irão acompanhar todo o processo (SOMMERVILLE, 2013).

#### 4.1.6 PROGRAMAÇÃO ORIENTADA A OBJETOS

A programação orientada a objetos, tem um enfoque diferente da programação imperativa, enquanto uma tem enfoque na utilização de classes e métodos, a outra foca em processos procedurais. Para a programação foi utilizado o PHP e Java, que são linguagens orientadas a objetos.

##### 4.1.6.1 CLASSES

Segundo Horstmann (2009), classe em programação orientada a objetos é um modelo usado para criação de objetos, consistindo em uma estrutura de dados que contém atributos, as quais são características dos objetos e métodos, que definem os comportamentos ou estados dos objetos. Para fazer a instanciação de uma classe, que é a criação de um objeto pertencente a essa classe, quando se instancia uma classe cria-se um objeto daquela classe e a instanciação é feita através do comando *new ()*. A classe contém três regiões:

- nome da classe: um identificador para a classe, que permite a referencia-la posteriormente;
- atributo: conjunto de propriedades da classe, também são conhecidos como variáveis da classe e pode-se especificar tanto: nome, tipo, visibilidade;
- métodos: são conjunto de funcionalidades de uma classe, ou seja todos determinam o comportamento dos objetos de uma classe.

##### 4.1.6.2 MÉTODOS CONSTRUTORES

Horstmann (2009) define que métodos construtores são responsáveis pela alocação de recursos necessários para o funcionamento de um objeto e inicialização de atributos. Ele

é chamado sempre quando se instancia uma classe. Um método construtor é um método em que as instruções serão executadas sempre que instanciado um objeto desta classe, ele possui um nome que deve ser o mesmo nome da classe e sem identificação do tipo de retorno. O construtor é invocado no momento da criação do objeto por meio do operador.

#### 4.1.6.3 SOBRECARGA DO CONSTRUTOR

Em relação à sobrecarga do construtor, [Horstmann \(2009\)](#) informa que a sobrecarga permite a existência de vários métodos com o mesmo nome. Caso tenham assinaturas diferentes, ou seja, variando-se entre eles os números, tipo de argumento, no valor de retorno e até variáveis diferentes, isso torna possível o funcionamento. Assim, ficará a cargo do compilador escolher, de acordo com as listas de argumento, os procedimentos ou métodos a serem executados. Um exemplo de sobrecarga será apresentado na sequência do texto.

```
public class Calculadora{
    public int Calculadora(int a, int b){
        return a+b;
    }

    public double Calculadora(int a, int b){
        return a+b;
    }

    public static void main(){}
        Calculadora calc = new Calculadora();
        System.out.println(calc.Calculadora(1),4);
        System.out.println(calc.Calculadora(1.1,2.5)):
    }
```

#### 4.1.6.4 ENCAPSULAMENTO

Encapsulamento, de acordo com [Horstmann \(2009\)](#), é o processo no qual se oculta dados do objeto e fornecem-se métodos para acesso a esses dados. O processo de encapsulamento é importante porque não permite que classes externas possam acessar atributos de uma classe. Desta forma, o encapsulamento providencia segurança a uma aplicação. Ele se dá quando há atributos privados que não permitem o acesso direto ou modificação direta por classes externas, sendo assim, a programação constitui-se em blocos que abstraem o funcionamento interno de uma classe. Esses blocos comunicam-se entre si por meio de métodos (*getters e setters*), facilitando a manutenção do código.

## 4.2 LINGUAGENS UTILIZADAS

Nesse tópico serão apresentadas as linguagens que foram utilizadas no desenvolvimento tanto no site, quanto do aplicativo.

### 4.2.1 HTML (*HYPERTEXT MARKUP LANGUAGE*)

Segundo Carril (2012) a linguagem HTML é uma linguagem de marcação, por isso ela utiliza anotações de texto para que seja sintaticamente distinguível. Esse princípio teve inspiração nas marcações dos manuscritos que os editores faziam nas revisões dos textos dos autores.

Guimarães (2005) informa que o HTML foi inventado por Tim Berners Lee e sua origem vem da linguagem de marcação SGML (*Standard Generalized Markup Language*), e foi desenvolvida a fim de apresentar e compor os documentos na *web*. A evolução histórica da linguagem segue a sequência de atualizações: SGML, HTML 1.0, HTML xx, XML, HTML 4.01, XHTML, HTML5. Atualmente os objetivos principais da linguagem são a de ser o *browser*, isto é, ser o navegador para apresentar visualmente os documentos, além da estruturação do documento, isto é, realizar a marcação.

A estruturação de documentos segue a sequência de um conjunto de elementos: demarcados por duas *tags*, a  $<$  (*tag* inicial) e a  $>$  (*tag* final). Sua estrutura deve seguir a seguinte tipologia:

```
<nome-do-elemento atributo(opcional)>
</nome-do elemento >
```

A linguagem HTML pode ser dividida em *elementos*, *atributos* e *entidades*, que serão explicados a seguir:

- elemento é a variável que recebe o documento, quando o elemento não possui conteúdo é chamado de "elemento vazio";
- atributo é o que caracteriza o elemento, a sintaxe e da forma: nome do atributo = "valor";
- entidade é a referência do documento, sua sintaxe segue a forma: &nome-da-entidade (GUIMARÃES, 2005).

### 4.2.2 CSS (*CASCADING STYLE SHEETS*)

Jobstraibizer (2009) informa que para automatizar a parte de criação de layout dos sites, por exemplo, alterar a cor do título de diversas páginas, foi idealizada a criação da linguagem CSS. Dessa forma, o programador se preocupa apenas com a etapa de marcação e estruturação na linguagem HTML, e utiliza o CSS como forma de design na edição dos *web sites*. A sintaxe básica que será apresentada na sequência foi retirada do site GitHub (2017).

Seletor é em síntese o elemento em HTML que será editado seguindo a sintaxe:

```
Seletor{  
  Propriedade: valor;  
}
```

Body é o comando para alteração de layout de acordo com o interesse do usuário, por exemplo, alterar a cor de fundo para cor preta. A sintaxe está sendo apresentada a seguir:

```
Body{  
  Background-color: #000;  
}
```

### 4.2.3 BOOTSTRAP

Para [Spurlock \(2013\)](#) a linguagem Bootstrap foi idealizada com o objetivo de facilitar a criação de *layout* responsivo na *web*. Seus inventores são dois funcionários Mark Otto e Jacob Thornton da rede social de relacionamento Twitter. A ideia principal da responsividade é programar a um código que seja melhor apresentado em diferentes telas, sem a necessidade de reescrever todo o código, e adaptar para as diferentes plataformas como: *tablets*, *smartphones*, *smart tv* e *smart watch*, além do *PC desktop*. O Bootstrap nada mais é do que uma biblioteca do CSS que contém diversos itens pré-programados que auxiliam na elaboração (desenvolvimento) de um site.

### 4.2.4 PHP (HYPERTEXT PREPROCESSOR)

Segundo [Converse e Park \(2003\)](#) a linguagem PHP (*Hypertext Preprocessor*) é utilizada para criação dos *scripts* dos *web sites* em união com a linguagem HTML. O motivo principal de acrescentar o PHP é o fato de introduzir fragmentos de código em páginas HTML. Além disso, a linguagem auxilia no acesso ao banco de dados do servidor. Contudo, os autores advertem que é quase impossível perceber a escrita da linguagem para o usuário final, tendo em vista que o PHP é processado no servidor. Assim, a percepção para o usuário é somente o resultado em HTML.

Diversas são as vantagens listadas pelos autores [Converse e Park \(2003\)](#) para a utilização do PHP, entre elas o fato de ser um software livre, bem como possuir código aberto, possuir diversos recursos de implementação, além de ser estável e compatível com várias linguagens. Referente à origem dessa linguagem, os autores [Converse e Park \(2003\)](#) são enfáticos ao afirmar que o criador foi Rasmus Lerdorf e que ele desenvolveu essa linguagem com o objetivo de auxiliar a verificação de pessoas que acessavam seu site. Inicialmente ele nomeou seu invento de (*Personal Home Page tools*), após três anos de sua invenção a linguagem já estava em utilização por todo o globo, além de cerca de

cinquenta mil sites possuem em sua composição o PHP. Os comandos básicos do PHP de acordo com Lobo (2007), estão sendo apresentados na sequência:

- Delimitadores do script PHP: sua utilidade é mostrar para o interpretador que o código a seguir é em PHP, e assim executá-lo. Sintaxes mais utilizadas:

```
<?php
Comandos ;
?>
<%
Comandos ;
%>
```

- Variáveis: são os endereços que recebem um valor para executar alguma operação. Sintaxe mais utilizada:

```
<?php
$paragrafo = "trabalhando com php";
?>
```

- Echo é o responsável por mostrar os dados na tela. Sintaxe mais utilizada:

```
echo "<p align='center'>";
```

- Comentário é utilizado para comentar o código e facilitar o entendimento de terceiros. Sintaxe mais utilizada:

```
/*
Aqui h
um comentario
de varias linhas
*/

// Ja neste caso um comentario de somente uma linha
```

- *Array* é o conjunto de vetores que formam uma matriz de dados. Sintaxe mais utilizada:

```
<?
$frase [1]= "trabalho";
$frase [2]= "com";
$frase [3]= "PHP";
?>
```

Vale ressaltar que, como se pode perceber, as variáveis não necessitam da indicação de tipo, o interpretador da linguagem PHP o faz automaticamente.

### 4.2.5 FRAMEWORK

De acordo com [Gamma et al. \(1994\)](#) um *framework* no âmbito de software é composto por um conjunto de funções, classes, entre outros elementos que geralmente usa-se diversas vezes dentro de um projeto, sendo assim o mesmo facilita o reuso das funcionalidades, dispensando a necessidade de reescrita de cada função a cada utilização. "*A framework is a set of cooperating classes that make up a reusable design for a specific class of software*"([GAMMA et al., 1994](#), p. 40).

### 4.2.6 LARAVEL

De acordo com [Gabardo \(2017\)](#), o Laravel é um *framework* de PHP. A linguagem foi idealizada por Taylor B. Otwell, sua função é automatizar para tornar mais fácil e rápido o desenvolvimento de processos como autenticação, roteamento, sessão, filas e cache que são os mais comuns para realizar o projeto, essas ferramentas são necessárias em praticamente todos os sites o que torna o uso do Laravel indispensável.

O autor comenta que essa demanda por velocidade de produção ocorreu devido ao fato de, atualmente, se estar num ritmo acelerado de atualização e criação de páginas para a internet, pois o desenvolvimento de nova aplicação gera a necessidade de atualização frequente. Devido a isso, aumenta-se a transformação e adequação dos sites gerando grande volume de trabalho, o que quase torna obrigatório o uso da linguagem para conseguir acompanhar a essa demanda.

Além dos itens citados, uma grande vantagem deste *framework* é a conexão com o banco de dados, que se faz de forma bem prática, sendo definidos somente alguns parâmetros para a conexão e tendo também de forma pré-implementada o necessário para a criação de novas tabelas.

### 4.2.7 BANCO DE DADOS RELACIONAL

Essa metodologia foi desenvolvida por DR.E.F.Codd e apresentada por meio de artigo no ano de 1970. O nome do artigo é "*A Relational model of data for large shared data banks*"(Um modelo de dados relacional para grandes bancos de dados compartilhados). Para exemplificar, o conceito de banco de dados relacional é a variedade de informações relacionadas, normalmente, organizadas em tabelas. Essas tabelas armazenam dados em linhas ou colunas, e os administradores verificam se determinado usuário pode ou não ter acesso às informações para, assim, realizar a permissão de acesso ([WIDENIUS, 2010](#)).

O autor destaca a importância de entender que há duas etapas bem definidas: na primeira etapa, os programas realizam o acesso e tratamento, o que é denominado sistema de gerenciamento. Já na segunda etapa é realizada a armazenagem. Como exemplo desse tipo de montagem tem-se os bancos de dados citados pelo autor, que são o Oracle, SQL serve, DB2 e o MYSQL. Assim, fica evidente que para um banco de dados é necessária a

inserção de dados, assim como a extração dos dados. Para normatização de linguagem os sistemas normalmente utilizam o SQL (*Structured Query Language*), que é uma maneira de trazer, somar e retirar dados do banco de dados (WIDENIUS, 2010).

#### 4.2.7.1 MYSQL

Segundo o manual, o MySQL foi primeiramente feito para interligar as tabelas usando as rotinas de baixo nível (ISAM), contudo, inicialmente foi criado o mSQL que possui problemas tanto de velocidade, quanto de flexibilidade para necessidades reais de uso. Para resolver esse problema, a solução foi a criação de uma nova interface SQL, não alterando a interface do API do mSQL. O motivo de não ser alterado a API foi para haver a possibilidade de facilitar a compatibilidade e para a migração dos código feitos em mSQL serem transferidos para a nova linguagem MySQL. O nome parte do nome *My* foi derivado da filha do fundador Monty que possui o mesmo nome (WIDENIUS, 2010).

##### 4.2.7.1.1 CARACTERÍSTICAS PRINCIPAIS DO MYSQL

Esta linguagem é escrita em C ou C++, facilitando assim seu uso. Devido à quantidade de pessoas que conhecem esses tipos de codificação, ela possui uma gama grande de compiladores, tanto grátis como pagos, que funciona em várias plataformas. Usa o GNU Automake, Autoconf e Libtool para portabilidade. O API aceita as seguintes linguagens: Tcl, Ruby, Python, PHP, Perl, Java, Eiffel, C e C++ (WIDENIUS, 2010).

Em relação à realização de multitarefas com o uso dos *threads*, o sistema é feito diretamente no *kernel* com o objetivo de aumentar a facilidade do uso de múltiplos processadores. Além disso, possui tanto armazenagem transacional como a não transacional. Utiliza também tabelas em disco, e sua base é em árvores-B para aumentar a velocidade da compressão dos índices. Também é possível adicionar outro tipo de mecanismo de armazenamento com extrema facilidade (WIDENIUS, 2010).

Referente às tabelas *hash*, fica evidente, com base em Widenius (2010), que estão na memória normalmente são usadas como tabelas temporárias, outra vantagem é que as funções SQL, após inicializada a pesquisa, não possui mais memória alguma alocada. Além disso, como provado pelo manual, o MySQL Serve não tem dificuldade de apresentar datas até o ano 2037 para valores do tipo *TIMESTAMP* e datas maiores para valores *DATE* ou *DATETIME*. As datas aceitas são, até o momento, do ano 9999, como ilustra a figura 3, na sequência do texto.

##### 4.2.7.1.2 SINTAXE DO MYSQL

A seguir são apresentadas sintaxes dos comandos de SQL. Esses comandos são feitos para manipular informações contidas nos banco de dados, podendo ser editadas as informações que o banco possui. Outra opção é mudar a estrutura alterando o número de

Figura 3 – *MySQL* com datas mais avançadas

```
mysql> DROP TABLE IF EXISTS y2k;
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE y2k (date DATE,
->                        date_time DATETIME,
->                        time_stamp TIMESTAMP);
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO y2k VALUES
-> ("1998-12-31", "1998-12-31 23:59:59", 19981231235959),
-> ("1999-01-01", "1999-01-01 00:00:00", 19990101000000),
-> ("1999-09-09", "1999-09-09 23:59:59", 19990909235959),
-> ("2000-01-01", "2000-01-01 00:00:00", 20000101000000),
-> ("2000-02-28", "2000-02-28 00:00:00", 20000228000000),
-> ("2000-02-29", "2000-02-29 00:00:00", 20000229000000),
-> ("2000-03-01", "2000-03-01 00:00:00", 20000301000000),
-> ("2000-12-31", "2000-12-31 23:59:59", 20001231235959),
-> ("2001-01-01", "2001-01-01 00:00:00", 20010101000000),
-> ("2004-12-31", "2004-12-31 23:59:59", 20041231235959),
-> ("2005-01-01", "2005-01-01 00:00:00", 20050101000000),
-> ("2030-01-01", "2030-01-01 00:00:00", 20300101000000),
-> ("2050-01-01", "2050-01-01 00:00:00", 20500101000000);
Query OK, 13 rows affected (0.01 sec)
Records: 13 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM y2k;
```

date	date_time	time_stamp
1998-12-31	1998-12-31 23:59:59	19981231235959
1999-01-01	1999-01-01 00:00:00	19990101000000
1999-09-09	1999-09-09 23:59:59	19990909235959
2000-01-01	2000-01-01 00:00:00	20000101000000
2000-02-28	2000-02-28 00:00:00	20000228000000
2000-02-29	2000-02-29 00:00:00	20000229000000
2000-03-01	2000-03-01 00:00:00	20000301000000
2000-12-31	2000-12-31 23:59:59	20001231235959
2001-01-01	2001-01-01 00:00:00	20010101000000
2004-12-31	2004-12-31 23:59:59	20041231235959
2005-01-01	2005-01-01 00:00:00	20050101000000
2030-01-01	2030-01-01 00:00:00	20300101000000

Fonte: (WIDENIUS, 2010)

linhas e colunas para aumentar ou diminuir o determinado banco. Além disso, há uma parte específica para os acessos aos bancos de dados, a qual influencia as relações de permissão que determinados usuários possuem (WIDENIUS, 2010).

Para consultas e localização dos dados é usado o comando *select*. Para realizar modificações nas tabelas, pode se usar o comando *Dml*, que possui três derivações: *insert*, *update* e *delete*. O primeiro adiciona linhas ou colunas à tabela; o segundo altera uma linha ou coluna já existente; e o último deleta a linha ou coluna da base de dados (WIDENIUS, 2010).

Para realizar mudança nas estruturas de tabelas existe o comando *Ddl*, tendo cinco principais divisões: *create*, *drop*, *alter*, *rename* e *truncate*. O primeiro desses comandos, o *create*, é usada para criar tabelas ou mesmo usuários; o comando *drop* remove toda uma estrutura de dados, o comando *alter* é usado para mudar de posição os dados de uma tabela, e o comando *rename* é usado para mudar o nome de uma tabela ou usuário. Por fim, o comando *truncate* é feito para deletar tabelas inteiras do banco de dados (WIDENIUS, 2010).

Em relação aos comandos de instrução TC, são dividido em três, em síntese, sua utilização é para registrar alterações feitas no banco de dados do sistema. Os comandos são *commit*, *rollback* e *savepoint*. O comando *commit* realiza salvamento sempre que uma linha é alterada; o *rollback* desfaz as alterações no banco de dados; e o *savepoint* define um ponto onde o sistema deve voltar (WIDENIUS, 2010).

Além disso, também existem as instruções *dcl*, que modificam os acessos liberados a determinados usuários para que eles tenham acesso aos dados. As instruções *dcl* são divididas em duas: *grant* e *revoke*. A instrução *grant* concede permissões aos usuários, enquanto a *revoke* retira permissões dos usuários. Outra função muito utilizada é a *SYSDATE*, cujo objetivo é devolver a data atual. Essa função é intrínseca aos bancos de dados. Ela possui uma tabela com apenas uma única linha ([WIDENIUS, 2010](#)).

#### 4.2.7.2 ANDROID

O Android, desenvolvido pelo Google, é uma plataforma para desenvolvimento de aplicativos *mobile*, que engloba celulares e *tablets*, e esta em cerca de 60 por cento dos celulares atualmente.

##### 4.2.7.2.1 CÓDIGO ABERTO

A grande vantagem do Android é, sem dúvida, o grau de abertura de sua plataforma, pois seu código fonte é aberto e gratuito. Dessa forma, é possível ver a maneira como os recursos são implementados. Outra vantagem é a possibilidade de contribuir no relato de erros encontrados, que pode ser feito por meio do site <http://source.android.com>, ou por meio do grupo de discussão *Open Source Project*. Assim fica evidente que pelo fato de ser uma plataforma aberta - ao contrário da plataforma iOS, que é patenteado pela Apple e só funciona nos dispositivos dela - Android é utilizado por diversos aparelhos de diferentes marcas, o que gera uma grande concorrência entre elas, e, por consequência, gera benefício para os clientes ([DEITEL, 2015](#)).

##### 4.2.7.2.2 JAVA

Para desenvolver os aplicativos Android é utilizada a linguagem java, pelo fato de : ser uma das mais usadas no mundo; possuir poderosas bibliotecas; ser gratuita; seu código fonte ser aberto e já possuir milhões de desenvolvedores que a conhecem. Dessa maneira, os programadores Java que possuem experiência podem aprofundar de maneira rápida o desenvolvimento de Android usando as API (interfaces de programação de aplicativo) Android do Google. Essa linguagem utiliza-se da orientação a objetos para ter acesso a várias classes que auxiliam na criação e desenvolvimento de aplicativos complexos de maneira coesa e objetiva ([DEITEL, 2015](#)).

Para a interface gráfica o Java é baseado em eventos, assim escreve-se aplicativos para responder a vários eventos indicados por um usuário, por exemplo, um toque na tela. Além desses recursos, há a possibilidade de usar os IDE do eclipse ou de o Android Studio montar a interface gráficas da maneira que ficar mais conveniente as telas com botões e caixa de texto predefinidos. Com isso é possível rotular e dimensionar esses objetos.

Além disso, pode ser criada, executada, testada e deputada através da IDE do Android de maneira conveniente e rápida (DEITEL, 2015).

#### 4.2.7.2.3 TELA MULTITOUCH

Dentre as funcionalidades dos *smartphones* estão: telefone celular; acesso ao *Mp3 player*; câmeras digitais; controle para jogos diversos; tela sensível ao toque; giroscópio e acesso à internet. Com esses recursos é possível acessar seu telefone, tocar música, executar aplicativos, navegar na *web*. A tela poderá fazer um teclado virtual para digitação de mensagens de texto, e-mails e para a inclusão de informação nos aplicativos (DEITEL, 2015).

#### 4.2.7.2.4 APLICATIVOS INCORPORADOS

Aplicativos incorporados são os aplicativos padrões que vêm instalados de fábrica no celular. Isso varia de acordo com aparelho, mas normalmente são: telefone, agenda, câmera, buscador, calendário, Play Store, fotos, e-mail e calculadora. Em relação aos *web services*, são componentes de software, além de dados armazenados em nuvem, que podem ser acessados por aplicativos (ou por outro componente de software) como sites podem ser acessados por outro computador por meio da Internet. Com eles, pode-se criar *mashups*, que permitem desenvolver aplicativos rapidamente, combinando *web services* complementares, frequentemente de diferentes organizações e possivelmente com outras formas de feeds de informação (DEITEL, 2015).

#### 4.2.7.2.5 PROGRAMAÇÃO ORIENTADA A OBJETOS EM RELAÇÃO AO ANDROID

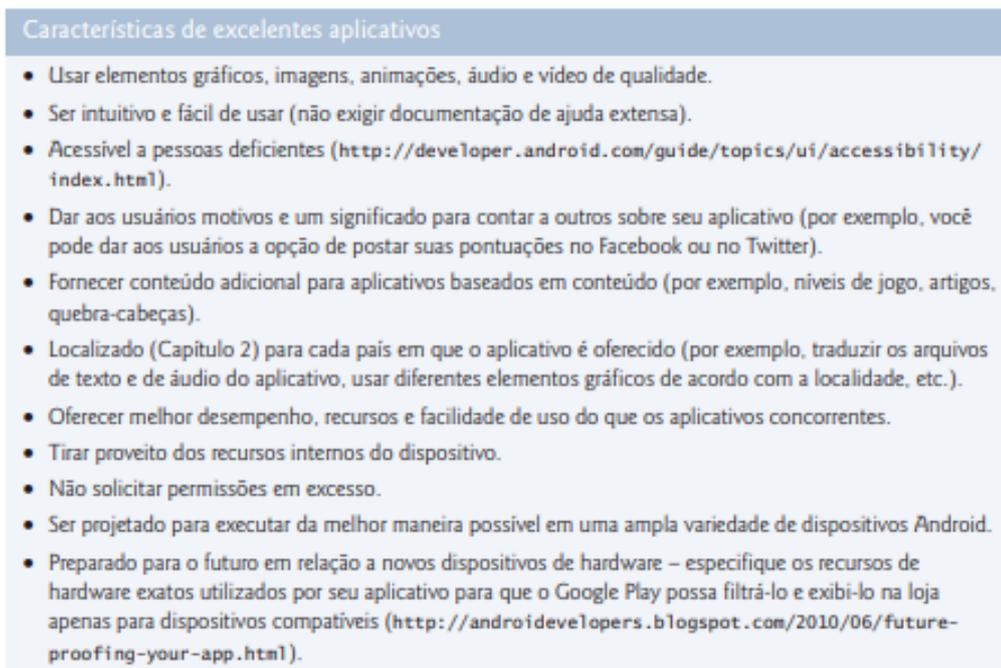
Alguns fundamentos em relação à programação orientada a objetos são necessários para criar um software de maneira econômica, rápida e correta. Por meio dessa técnica podem-se utilizar objetos que possam representar basicamente qualquer tipo de objeto com objetos tempo, áudio, vídeo, automóveis, pessoas. Além disso, os desenvolvedores verificaram que usar a estratégia de projeto e implementação modular orientada a objetos torna o desenvolvimento mais rápido, além de que, para se realizar a manutenção, torna-se mais fácil entender, corrigir e modificar, caso sejam necessárias futuras correções (DEITEL, 2015).

#### 4.2.7.2.6 COMO DESTACAR SEU APLICATIVO ANDROID

Uma das perguntas que muitos desenvolvedores de aplicativo se fazem é: *Entre tantos aplicativos como destacar o seu?* Uma maneira é torná-lo mais útil, interessante, duradouro e de fácil utilização. O nome auxilia tanto no quesito propaganda como para

buscar com maior facilidade. Outra maneira é sempre realizar uma boa descrição para atrair mais público para seu software. Com base nisso, a figura 4 a seguir lista alguns atributos importantes (DEITEL, 2015).

Figura 4 – Características dos excelentes aplicativos.



(DEITEL, 2015)

#### 4.2.7.2.7 IDE ANDROID DEVELOPER TOOLS

Essa ferramenta será necessária para construir o projeto. Por meio dela é criada a interface gráfica para o usuário, para na sequência ser usado o editor gráfico de layout do IDE, formando-se assim o texto e as suas imagens (DEITEL, 2015).

#### 4.2.7.2.8 COMPONENTES TEXTVIEW E IMAGEVIEW

Os textos criados no aplicativo são mostrados no componente *TextView*, enquanto as imagens são mostradas por meio do *ImageView*. Assim, para realizar uma modificação de tamanho de fonte ou cor, é preciso utilizar a *properties* do IDE para configurar as diversas opções disponíveis para estilo (DEITEL, 2015).

#### 4.2.7.2.9 RECURSOS DO APLICATIVO

É considerado uma boa prática definir todas as *strings* e números em arquivos de recurso, que também devem ser introduzidos em subpastas da pasta principal do projeto (DEITEL, 2015).

#### 4.2.7.2.10 ACESSIBILIDADE

O grande diferencial do Android são os recursos de acessibilidade disponíveis, por exemplo, o *Talkback* e o *Explore by Touch*. Esses recursos ajudam diversas pessoas com diversos tipos de deficiências. Para deficientes visuais e físicos o recurso *Talkback* permite que o aparelho pronuncie o texto que está na tela. Já o recurso *Explore by Touch* auxilia o usuário a verificar o aplicativo para, por meio do *Talkback*, descobrir qual aplicativo está selecionado (DEITEL, 2015).

#### 4.2.7.2.11 INTERNACIONALIZAÇÃO

Para que o aplicativo tenha grande distribuição é necessário internacionalizá-lo. Assim, atinge um grande público para consumo do software. Contudo, para isso é necessário personalizá-los para diversas localidades e idiomas (DEITEL, 2015).

#### 4.2.7.2.12 MÉTODOS DE CICLO DE VIDA DE ACTIVITY

Os estados possíveis de uma atividade são: ativas ou em execução, em pausa e parada. Dessa forma a atividade permuta entre os estados disponíveis. Normalmente uma atividade ativa é visível na tela do aplicativo, além de estar em primeiro plano e com o usuário interagindo com ela. Na forma pausada é visível na tela, mas sem o foco na atividade, um exemplo é uma caixa de diálogo com alerta em exibição. Já a forma parada não é visível na tela e provavelmente será encerrada pelo sistema quando esse necessitar da memória para realizar um novo processo (DEITEL, 2015).

Para controlar as transições das atividades, o *runtime* do Android utiliza diversos métodos de ciclo de vida definidos pela classe *Activity*. Dentre esses métodos destaca-se o *onCreate* que sobrescreve em cada atividade quando ela está iniciando. Além disso, existem os métodos *onStart*, *onPause*, *onRestart*, *onResume*, *onStop* e *onDestroy*. É necessário em cada ciclo de vida utilizar um desses métodos para sobrescrever, para, assim, chamar a superclasse (DEITEL, 2015).

### 4.2.8 WEBVIEW

O sistema *WebView* utiliza a tecnologia Google Chrome para poder fazer a abertura de página web sem o uso de navegadores, mas sim por meio do próprio aplicativo.

Para realizar esse processo, os aplicativos abrem as janelas de maneira interna, sem a presença do navegador. Além disso, outra vantagem do *WebView* é que ele já vem instalado nos aparelhos Android, sendo sempre recomendado que ele esteja atualizado para receber correções de segurança ou eventuais erros das versões anteriores. (GOOGLE, 2018)

## 4.3 DISPOSITIVOS UTILIZADOS

Nessa parte serão abordados os dispositivos que foram utilizados na composição do protótipo, que comprovará a funcionalidade do site e aplicativo desenvolvidos durante esse trabalho.

### 4.3.1 PHOTON

O microcontrolador possui os botões de *reset*, que permite reinicializar o dispositivo, e de *setup*, que se usados em conjunto podem fazer o Photon voltar à sua configuração de fábrica. Entre eles a um led de status, que dependo da cor mostra o estado de execução do microcontrolador. No dispositivo a porta usb para fornecer energia para o Photon, sendo que ela também pode ser usada como porta de comunicação com o computador (MONK, 2018).

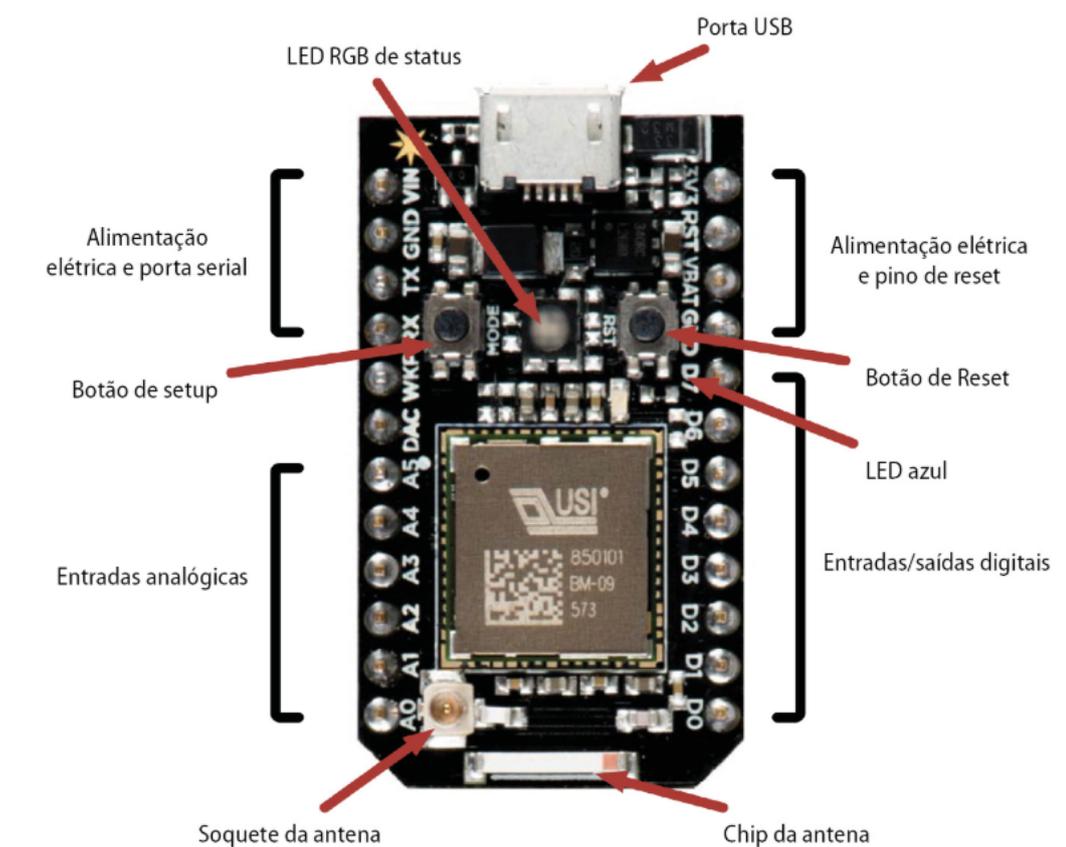
Movendo-se ao redor da placa no sentido horário, você encontra os pinos de alimentação elétrica (3V3, VBAT, GND) e de Reset ou reinicialização (RST). O Photon converte a alimentação elétrica fornecida através da porta USB, ou pelo pino VIN (na parte de cima do lado esquerdo do Photon), em uma tensão de 3,3V (volts) usada na placa. Essa tensão está disponível no pino 3V3, na parte bem de cima do lado direito. A lógica opera com 3,3V em vez dos 5V que você talvez esteja acostumado a utilizar em um Arduino. O pino RST pode ser usado com um botão Reset, que ao ser pressionado faz a tensão nesse pino baixar para 0 volts (terra elétrico) ou GND. Isso pode ser útil caso você aloje seu projeto em algum tipo de recipiente, como em uma caixa de proteção. Fora desse caso, é improvável que você use esse pino (MONK, 2018, p.7).

Para colocar o Photon em modo hibernação é possível usa o pino VBAT e acoplar uma bateria, assim o microcontrolador conserva sua memória e pode continuar do ponto que está antes da hibernação. Os pinos D0 a D7 são usado com entrada e saída digital, o diferencial está nos pinos D0 a D3 que pode atuar como pinos de saída analógica através da técnica de modulação *Pulse Width Modulation* (PWN), no pino D7 a um led azul que pode ser acionado (MONK, 2018).

No Photon há uma antena interna para fazer a conexão wifi, mas também existe um soquete onde pode ser acoplado uma antena para um melhor desempenho. Os pinos A0 a A5 são entradas analógicas com tensão 3,3 v, onde normalmente colocam-se os sensores. O destaque é para os pinos A4 e A5 que podem ser usados como saída PWN. O pino DAC é um pino para saída analógica que pode assumir valor de 0 até 3,3 v (MONK, 2018).

O pino WKP é usado para entrar em modo de operação, após ter entrado em modo hibernação. Os pinos TX e Rx são usado na comunicação serial, além do pino de *Graduated Neutral Density filter* (GND), que é o terra, e o pino VIN que pode receber energia no lugar da porta *Universal Serial Bus* (USB). A figura 5 a seguir mostra uma imagem completa do Photon ([MONK, 2018](#)).

Figura 5 – O Photon



Fonte: ([MONK, 2018](#))

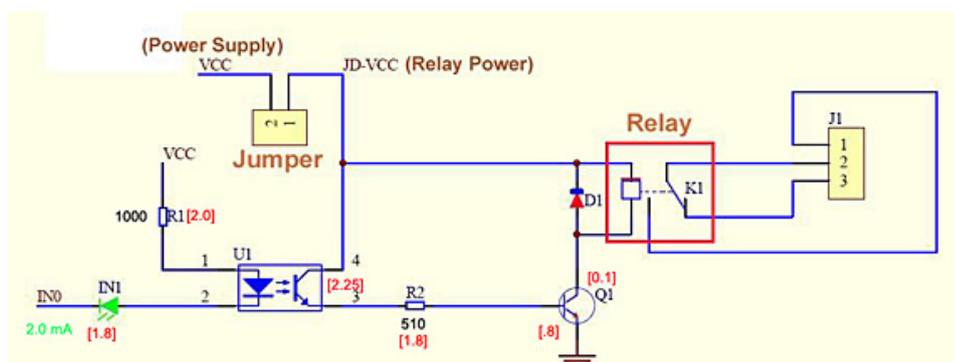
### 4.3.2 MÓDULO RELÉ

Para acionar a cargas foi escolhido o módulo relé. Seu circuito é mostrado na figura 6. Os componentes do módulo relé são: um led, um *octa acoplador* (conjunto de *phototransistor* e um led), um transistor, um diodo, *resistors* e um *relay*, que em português é chamado de relé. No item seguinte tem-se uma melhor explanação do que é um relé.

### 4.3.3 RELÉ

Para o acionamento das cargas será usado o dispositivo relé. O relé escolhido foi o da *hong wei*. As principais características de acordo fabricante são ([RELAY, 2019](#)) :

Figura 6 – O circuito



Fonte: (JEFFERSON, 2012)

- capacidade de chaveamento de 10 A, com design pequeno de alta densidade;
- seleção de materiais plásticos para altas temperaturas;
- relé magnético com circuito simples que relaciona baixo e produção em massa;
- padrões UL, CUL, TUV como mostra a figura 7;
- Como será usado em 3 v tem-se, de acordo com fabricante, que o valor da corrente é de 120 mA, como mostra a figura 8.

Figura 7 – Tensão de operação

#### 4. RATING

CCC	FILE NUMBER: CQC03001003729	7A/240VDC
CCC	FILE NUMBER: CQC03001003731	10A/250VDC
UL /CUL	FILE NUMBER: E167996	10A/125VAC 28VDC
TUV	FILE NUMBER: R50056114	10A/250VAC 30VDC

Fonte: (RELAY, 2019)

### 4.3.4 MEDIDOR DE CORRENTE NÃO INVASIVO

Para fazer a medição de corrente foi pensado em usar um medidor de corrente não invasivo, isto é, ele não necessita ser conectado diretamente no circuito, sendo assim será utilizado o medidor de corrente modelo SCT-013 séries. As especificações do fabricante são as seguintes:

Figura 8 – Correntes de acionamento do relay

**6. COIL DATA CHART (AT20 ° C)**

Coil Sensitivity	Coil Voltage Code	Nominal Voltage (VDC)	Nominal Current (mA)	Coil Resistance (Ω) □	Power Consumption (W)	Pull-In Voltage (VDC)	Drop-Out Voltage (VDC)	Max-Allowable Voltage (VDC)
SRD (High Sensitivity)	03	03	120	25	abt. 0.36W	75%Max.	10% Min.	120%
	05	05	71.4	70				
	06	06	60	100				
	09	09	40	225				
	12	12	30	400				
	24	24	15	1600				
	48	48	7.5	6400				
SRD (Standard)	03	03	150	20	abt. 0.45W	75% Max.	10% Min.	110%
	05	05	89.3	55				
	06	06	75	80				
	09	09	50	180				
	12	12	37.5	320				
	24	24	18.7	1280				
	48	48	10	4500				

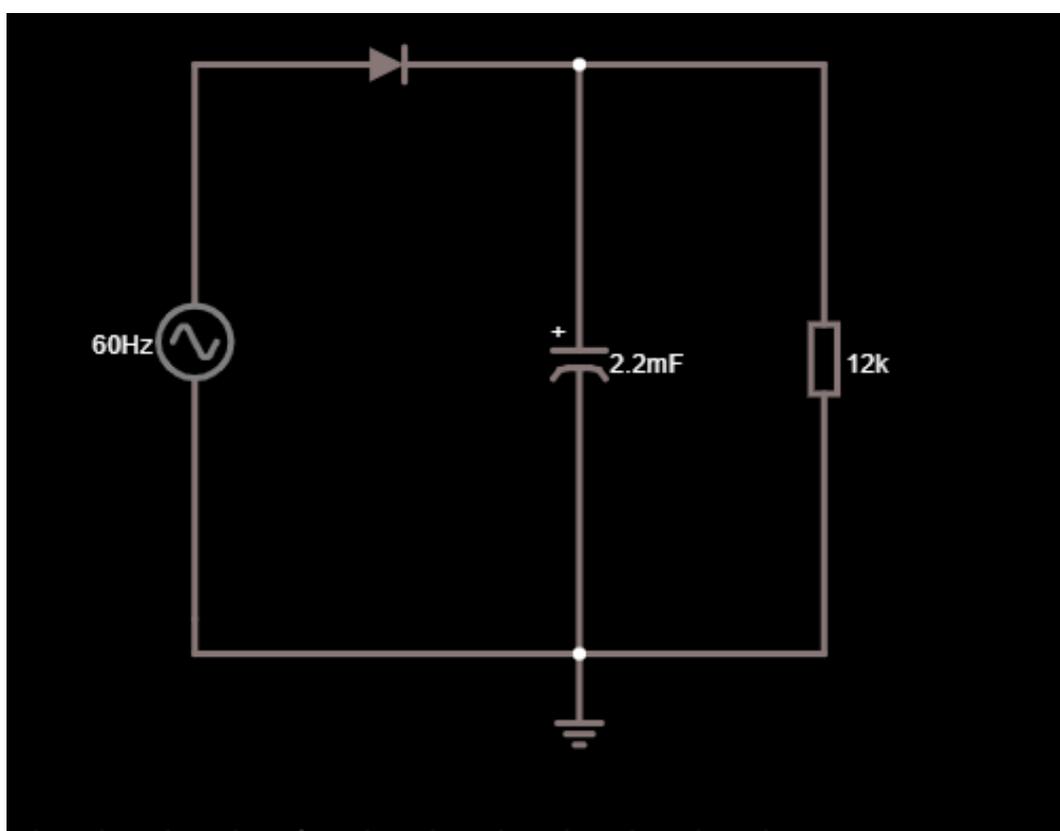
Fonte: (RELAY, 2019)

- corrente de entrada: 0-30 A;
- sinal de saída: 1 Volt para cada 30 amperes;
- material do Core: Ferrite;
- dielétrico: 6000 V AC/1 min;
- taxa anti-chama: UL94-V0;
- plug de saída: 3,5 mm;
- dimensão abertura: 13 x 13 mm;
- temperatura de trabalho: -25 a +70°C;
- comprimento do cabo: 150 cm.

#### 4.3.5 RETIFICADOR DE MEIA ONDA COM FILTRO CAPACITIVO

Para a linearização da corrente emitida pelo sensor de corrente foi feito um circuito retificador composto por diodo mais capacitor e resistor como mostra a figura 9 na seqüência.

Figura 9 – Retificador de meia onda com filtro capacitivo



Fonte:O autor

# 5 DESENVOLVIMENTO DO PROJETO

Esse projeto foi desenvolvido buscando a individualização de determinados pontos de energia, que encontram-se em áreas de uso compartilhado. Para conseguir alcançar esse objetivo, tem-se que foi desenvolvido um site e um aplicativo de celular, onde o usuário, após cadastrado corretamente no sistema, consegue realizar o agendamento da utilização do ponto de energia de determinado bem de consumo que ele tenha interesse de utilizar. Feito isso, no dia do agendamento, o usuário deverá conectar-se ao site, efetuar seu login, para então realizar o ativamento do ponto de energia selecionado.

Para uma limitação do campo de atuação, tem-se que o projeto foi pensado em função da conectividade de quatro pontos de energia, relacionados ao espaço de uso compartilhado de uma lavanderia, sendo que foi pensado somente no produto máquina de lavar, a qual já possui a função de secadora. Com essa limitação, tem-se que foi melhorado um enfoque no projeto, sendo que inclusive será realizado um escopo de empresa, com base no projeto feito e nos dados aqui comentados.

Para um melhor detalhamento dos processos de desenvolvimento desse projeto, dividiu-se esse capítulo em *front-end*, *back-end* e *protótipo*, sendo que no primeiro tópico fala-se sobre a página do site, o que o usuário terá acesso, no segundo item, tem-se o comentário do processo que acontece no servidor, sendo que esse o usuário não possui acesso, e por fim, o último tópico fala sobre a aplicabilidade e funcionalidade do site no projeto desenvolvido..

## 5.1 FRONT-END

Como proposta tem-se o desenvolvimento de um site utilizando a linguagem HTML, pois utilizando esse tipo de linguagem, tem-se uma melhor abordagem, pois ela é compatível com diversos tipos de navegadores que estão disponíveis no mercado atualmente.

### 5.1.1 VISÃO GERAL

O objetivo desse projeto é o desenvolvimento de um site que disponibiliza a utilização de determinados equipamentos eletroeletrônicos, a partir de um agendamento prévio. Pela quantidade de tempo, e disponibilidade de recursos, tem-se que durante esse projeto foi feito um protótipo, visando a comprovação da funcionalidade, composto por microcontrolador, módulo relé e lâmpadas. O acionamento é feito a partir do site, que manda uma mensagem ao microcontrolador, para acionamento das portas, a partir do uso do relé, acendendo assim a lâmpada que foi selecionada e agendada.

### 5.1.2 ÁREA DE LOGIN

Para fazer o login foi feita uma página em html, que recebe os dados. Caso o usuário selecione o botão "Entrar", é ativado o método "autenticar", que verifica no banco de dados se esse usuário já foi cadastrado, através do comando a seguir:

```
1 | <form method="post" action="/autenticar">
```

A figura 10, a seguir, mostra a página de login criada.

Figura 10 – Página inicial.



Fonte: O autor

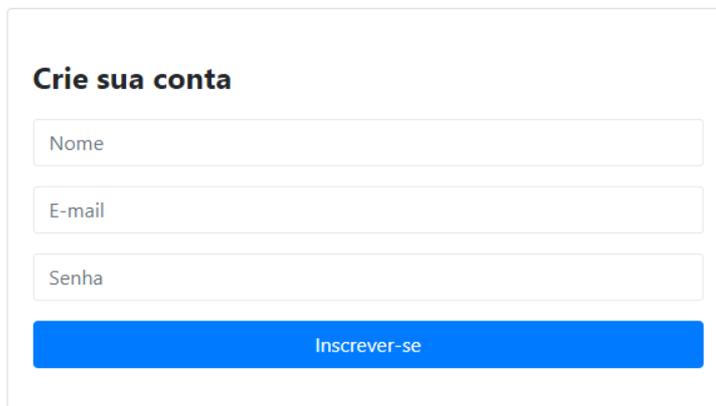
Outra possibilidade que a página de login apresenta, é a de o usuário selecionar a opção de inscrever-se, a partir do botão "Inscrever-se". Neste caso o usuário é transferido para uma nova página, na qual realizará seu cadastro como usuário, assunto esse que será melhor explanado na sequência.

### 5.1.3 ÁREA DE REGISTRO

Para registrar um novo usuário, ao clicar no botão de "Inscrever-se", o usuário é redirecionado para a página de registros (Figura 11), onde o usuário tem que preencher os campos existentes com as informações de nome, e-mail e escolha de uma senha. Tem-se que essa página possui um código semelhante ao da página de login, mas com diferenças relevantes. Uma diferença que pode ser citada é em relação aos campos, neste caso há mais campos; a outra é o objetivo da página, sendo que a linguagem utilizada, pode ser observada a seguir:

```
1 | <form action="/registrar" method="post">
```

Figura 11 – Página de registro do usuário do aplicativo.



O formulário de criação de conta, intitulado "Crie sua conta", contém três campos de entrada: "Nome", "E-mail" e "Senha". Abaixo dos campos, há um botão azul com o texto "Inscrever-se".

Fonte: O autor

#### 5.1.4 PÁGINA INICIAL

A página inicial, representada pela figura 12, foi desenvolvida para dar acesso às três páginas do site, que são: agendamento, para ativar os pontos de energia e para acessar área de custo, através dos botões "Agendar", "Ativar" e "Custos", respectivamente. Em síntese, ela é uma página de direcionamento, porém também realiza a conferência do login, pois caso o login não tenha ocorrido de maneira correta, esta página redireciona novamente para a página de login, como mostra a figura 13.

Figura 12 – Página inicial.



Fonte: O autor

Para uma melhor compreensão do funcionamento do site, tem-se a explicação de cada uma das páginas para as quais, a página inicial oferece direcionamento.

Figura 13 – Acesso incorreto.



Fonte: O autor

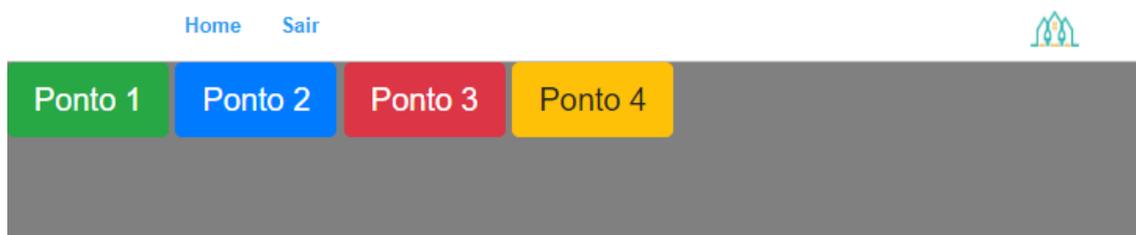
### 5.1.5 PÁGINA DE AGENDAMENTO

Ao clicar no botão "Agendar" o usuário será direcionado para as páginas do site responsáveis por fazer o agendamento. Tem-se que primeiramente ele terá que selecionar o ponto referente ao agendamento, para depois selecionar uma data e horário disponíveis.

#### 5.1.5.1 PÁGINA DE ESCOLHA DO PONTO DE ACESSO

Após o usuário selecionar o botão de "Agendar", ele é direcionado para uma página na qual ele deve escolher qual ponto ele gostaria de agendar, como mostra figura 14, essa primeira etapa foi necessária para separar qual dos quatro pontos ele vai selecionar. Essa divisão foi necessária para separar os bancos de dados e também pela estética da página de agendamento, pois com todos os pontos na mesma página, as informações estavam ficando muito tumultuadas.

Figura 14 – Página de escolha do ponto de acesso.



Fonte: O autor

### 5.1.5.2 PÁGINA DE AGENDAMENTO

Após selecionar o ponto de sua preferência, o usuário será direcionado para a página do calendário, como mostra a figura 15, onde deverá selecionar uma data e reservar um horário. Assim para realizar esse agendamento o usuário deve clicar no dia escolhido, e após essa ação uma janela de agendamento será aberta. Esta é aberta com a data padrão da data selecionada pelo usuário, indo das 00:00:00 horas até as 00:00:00 horas do dia seguinte, logo restará ao usuário selecionar o intervalo de tempo em que desejar utilizar o ponto de acesso. Por exemplo, o usuário escolhe a data 13/11/2019, então a janela de agendamento de utilização será aberta, onde ele escolherá o horário das 15:40:00 até as 17:00:00, do mesmo dia, assim o usuário terá esse período para utilizar o sistema. A figura 16 mostra o exemplo de agendamento dado nesse parágrafo. Ao confirmar clicar no botão "Confirmar" o usuário é redirecionado novamente para a página inicial, porém se ele clicar no botão "Cancelar" a janela será fechada e o usuário voltará para o calendário.

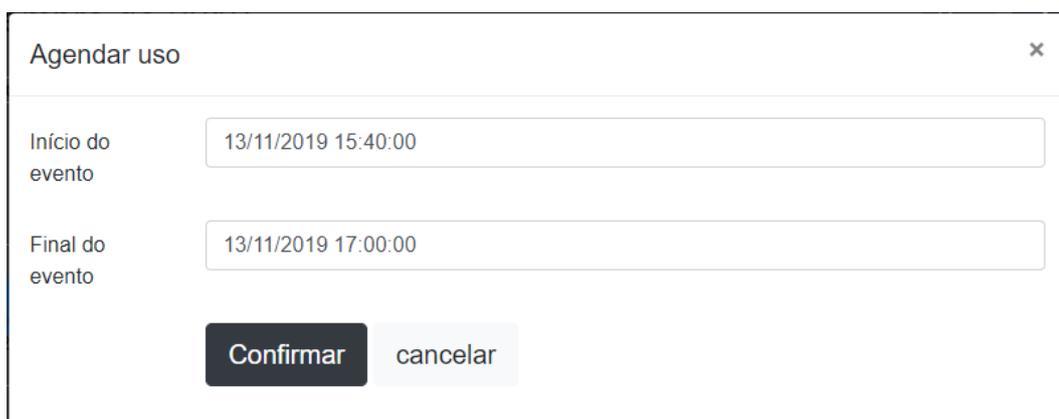
Figura 15 – Página de calendário.

dom	seg	ter	qua	qui	sex	sáb
27	28	29	30 15:10 andre	31	1	2 15:40 andre 20 andre 21:30 andre
3 11 noir 11:05 noir mais +2	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Fonte: O autor

Dentro da página do calendário é possível verificar quais usuários agendaram o equipamento selecionado, para assim escolher os horários vagos. Para casos em que uma data possuir mais que 3 agendamentos, optou-se por ocultar alguns agendamentos, visando

Figura 16 – Janela de agendamento.



A janela de agendamento, intitulada "Agendar uso", contém dois campos de entrada de texto. O primeiro campo, rotulado "Início do evento", contém o texto "13/11/2019 15:40:00". O segundo campo, rotulado "Final do evento", contém o texto "13/11/2019 17:00:00". Abaixo dos campos, há dois botões: "Confirmar" (em um fundo escuro) e "cancelar" (em um fundo claro). No canto superior direito da janela, há um ícone de "X" para fechar.

Fonte: O autor

a melhoria estética do calendário, mas se for necessário visualizar os agendamentos do dia basta clicar no campo abaixo dos agendamentos que aparecem, que no caso da figura 15 é "mais +2", e então serão mostrados todos os cadastrados do dia, como mostra a figura a seguir 17. Outra funcionalidade do calendário é o detalhamento da reserva, sendo possível verificar quem agendou, o horário de início e de término, como mostra a figura 18.

Figura 17 – Página todos os usuários cadastrados no dia.

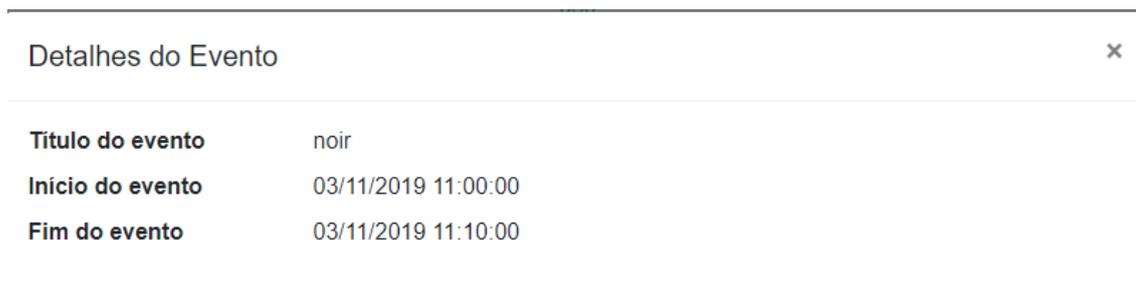


Fonte: O autor

### 5.1.6 PÁGINA DE ATIVAMENTO

Ao clicar no botão "Ativar", da página inicial, o usuário já deverá ter agendado uma data de utilização na página de agendamento, apenas dessa forma ele poderá ser direcionado para a próxima página.

Figura 18 – Detalhes do agendamento.



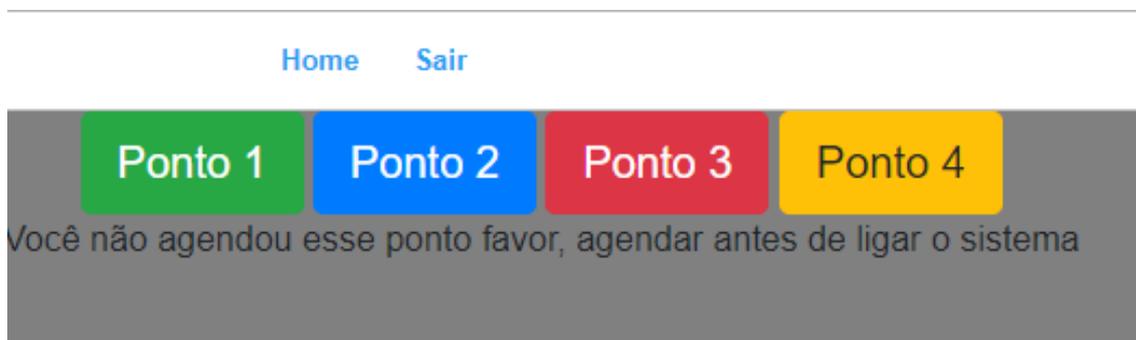
Detalhes do Evento	
<b>Título do evento</b>	noir
<b>Início do evento</b>	03/11/2019 11:00:00
<b>Fim do evento</b>	03/11/2019 11:10:00

Fonte: O autor

#### 5.1.6.1 TENTATIVAS DE ACESSO

Caso o usuário não tenha agendado o uso do equipamento e tente acessar determinado ponto, ele será redirecionado para a tela de escolha de pontos, com a seguinte frase: **"Você não agendou esse ponto, favor agendar antes de ligar o sistema"**, como mostra a figura 19, a seguir. Tem-se que ele é redirecionado para a página de agendamentos, para que assim ele possa realizar o ativamento do ponto de energia.

Figura 19 – Caso do não agendamento prévio.



Fonte: O autor

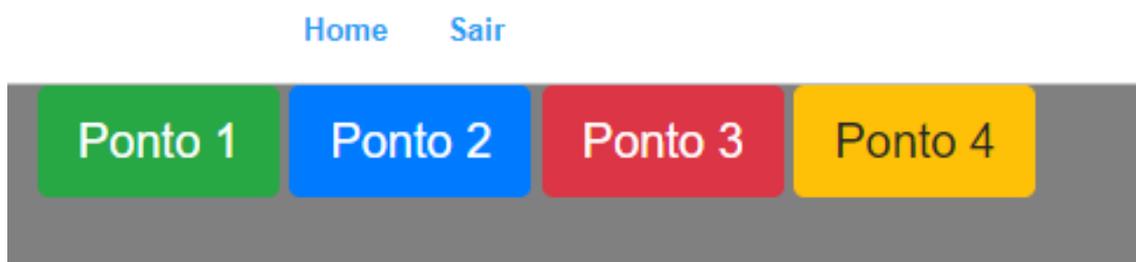
Mesmo que o usuário tenha efetuado o agendamento, para a data na qual ele esteja acessando o site, mas esteja entrando num horário diferente do qual ele agendou, ele não poderá utilizar o ponto de energia, e será redirecionado para a página de agendamento com a mesma frase, como mostrado anteriormente.

#### 5.1.6.2 PÁGINA DE ACESSO

Após o usuário fazer o agendamento do equipamento, ele poderá utilizá-lo apenas na data e horário agendadas. Assim na data prevista o usuário deverá acessar o site e na página inicial selecionar o botão "Ativar", então ele será direcionado para a página de

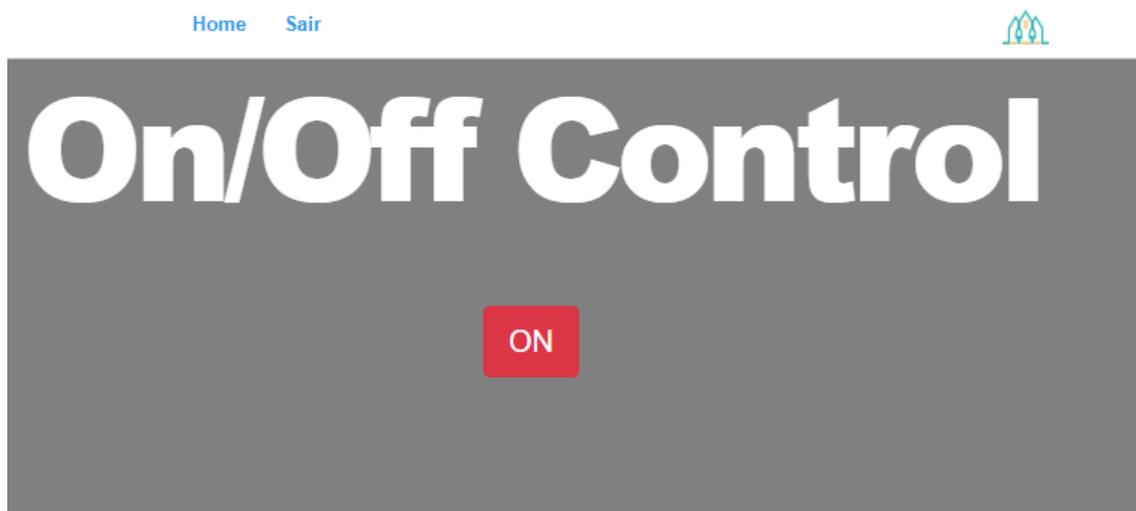
roteamento 20, onde deverá clicar no botão do ponto de acesso que foi agendamento. Então ele será direcionado para a página que ativa o circuito, como mostra figura a seguir 21, onde há apenas o botão "ON", neste caso quando ele selecionar este botão o equipamento será ativado e seu login é marcado como acessado, no banco de dados.

Figura 20 – Página de roteamento.



Fonte: O autor

Figura 21 – Página de ativamente do circuito.



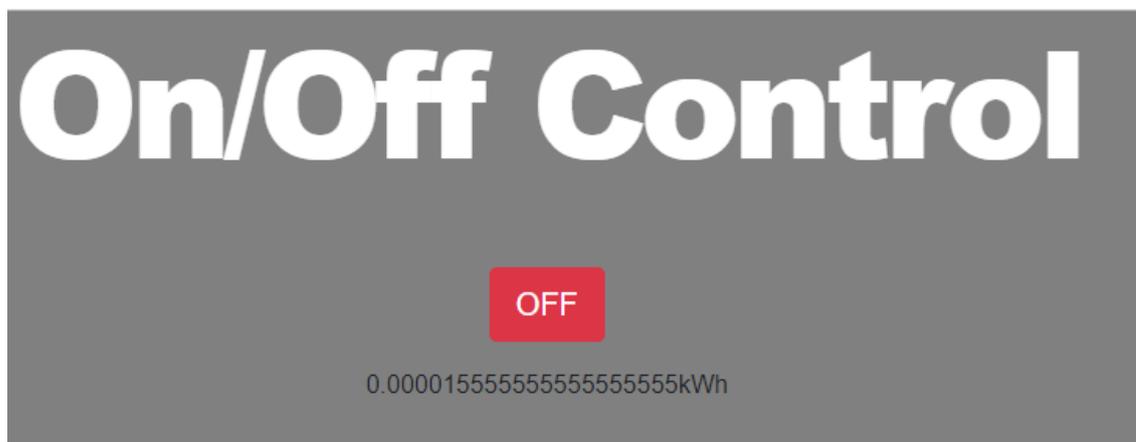
Fonte: O autor

### 5.1.6.3 PÁGINA DESATIVAR

Após ativado o circuito o usuário será redirecionado para a página de desligar, onde permanecerá durante todo o tempo de utilização do equipamento. Findada a atividade, o usuário deverá selecionar o botão "OFF" e então o sistema será desligado. Além disso, durante o tempo de utilização, será mostrado o valor do consumo da energia atual do equipamento, como mostra a figura 22. Assim quando o usuário sair do sistema será

contabilizado o gasto de energia utilizada durante o consumo. Este valor junto com o nome do usuário será salvo em banco de dados para uma posterior exibição na página de custo.

Figura 22 – Caso do agendamento prévio.



Fonte: O autor

### 5.1.7 PÁGINA DE CUSTOS

Nesta opção o usuário tem acesso ao seu histórico de utilização, onde estão relacionados os dados de pontos de acesso, data e horário final da utilização, além do valor do consumo de energia utilizado naquele acesso, como mostra a figura 23, na sequência.

## 5.2 BACK-END

O *back-end* faz o papel de servidor do site que está sendo planejado. Tem-se que as informações que são colocadas no site, são levadas ao banco de dados, o qual as analisa, segundo uma função, determinada pelo comando do site, e assim tem-se a "linkagem" entre o banco de dados e o site.

Por exemplo, quando o usuário digita o seu login, tem-se que as informações captadas no site, que são "nome do usuário" e "senha", são enviadas através da linguagem PHP e processadas através da linguagem SQL, no banco de dados, que faz a comparação e verificação entre os dados de acesso pré existentes (gravados no banco de dados) com os dados novos. Se acaso for verificado que aquele dados estão ali cadastrados, tem-se que o back-end devolve as informações ao site, liberando o usuário para acessar uma nova página, enquanto se os dados estiverem errados, ou o usuário não constar cadastrado no banco de dados, então o usuário é impedido pelo back-end de prosseguir para uma nova página. Essa explicação trata-se do método autenticar, mostrado a seguir:

Figura 23 – Página de custos.

		<a href="#">Home</a>	<a href="#">Sair</a>
Data/Hora	Ponto	Energia	
<b>November 2018</b>			
2018-11-20 19:38:45	3	0.00000888889 kWh	
<b>Total:</b>		<b>0.00000888889 kWh</b>	
<b>October 2019</b>			
2019-10-20 19:39:10	3	0.000233333 kWh	
2019-10-20 19:41:13	3	0.000604444 kWh	
2019-10-20 19:42:16	3	0.0000133333 kWh	
<b>Total:</b>		<b>0.0008511103 kWh</b>	
<b>November 2019</b>			
2019-11-20 19:42:47	3	0.0000133333 kWh	
2019-11-20 19:43:44	3	0.0000111111 kWh	
2019-11-20 19:44:04	3	0.0000222222 kWh	
2019-11-20 19:44:32	3	0.0000333333 kWh	
2019-11-20 19:45:22	3	0.00004 kWh	
2019-11-20 21:46:33	2	0.0000155556 kWh	
2019-11-20 21:48:52	1	0.0000133333 kWh	
2019-11-20 21:49:07	2	0.00000888889 kWh	
2019-11-20 21:49:54	4	0.00000888889 kWh	
2019-11-20 21:59:25	3	0.0000133333 kWh	
2019-11-20 22:01:51	3	0.00000666667 kWh	
2019-11-20 22:02:54	1	0.00000666667 kWh	
2019-11-20 22:03:13	1	0.00000666667 kWh	
<b>Total:</b>		<b>0.00019999989 kWh</b>	

```
1 | <form method="post" action="/autenticar">
```

O método registrar recebe os parâmetros "nome", "e-mail" e "senha" e submete esses dados ao banco de dados, que este verifica se os campos nome, e-mail e senha estão preenchidos, sendo que se acaso algum dos campos não estiver preenchido e o usuário tente clicar em "inscrever-se" ele recebe como devolutiva a seguinte mensagem de erro: **Erro ao tentar realizar o cadastro. Verifique se os campos foram preenchidos corretamente..**

Quando o usuário preencher todos os campos, com informações da maneira correta, por exemplo e-mail caracteres válidos, nome e senha com no mínimo três caracteres e também o e-mail não pode já estar cadastrado na base de dado. Nesse caso antes de realizar o cadastro, a senha recebe o processo de tratamento de codificação pelo código MD5, e então são salvos no banco os campos: nome, e-mail e senha. Além da chave primária, que é utilizada para identificar o usuário, esta é adicionada sempre com um número a mais que o usuário anterior. O código na sequência relata a ação "registrar" que é disparada quando o usuário seleciona o botão de "Inscrever-se".

```
1 | <form action="/registrar" method="post">
```

Para ativar o banco de dados, tem-se a utilização da linguagem PHP, que faz a conexão do uso do banco de dados MySQL. Tem-se que a linguagem SQL é utilizada tanto para a conversão dos dados que vem do site, quanto dos dados de programação que foram feitos, visando a conexão entre o site e o banco de dados. Por exemplo, quando o usuário entra em agendar um horário de utilização de um determinado ponto de acesso, tem-se a ativação de uma função dentro do banco de dados, assim quando o usuário recebe a resposta do site, que está agendado o ponto de acesso, é por que essa função foi compilada de forma correta dentro do banco de dados, liberando assim o acesso do ponto e redirecionando o usuário para a página inicial.

Em relação ao processo de programação do *back-end*, tem-se algumas dificuldades que foram aparecendo durante o desenvolvimento do projeto. A utilização do banco de dados e a programação em SQL, necessitavam de uma imersão de conhecimento por parte do autor, o qual não tinha a fundamentação concreta desse conteúdo, o qual não havia sido tão bem explorado durante a disciplina de programação. Outro item a ser acrescentado em relação às dificuldades que apareceram, foi a diferença entre os formatos de data utilizados no Brasil e nos EUA, sendo que o primeiro utiliza o formato DIA/MÊS/ANO e o segundo ANO/MÊS/DIA, onde site tinha como formato o padrão utilizado pelo Brasil, enquanto o banco de dados possuía o padrão dos EUA. Para a resolução desse problema foi utilizada a linguagem PHP, que faz a mudança do padrão brasileiro para o padrão americano, para após isso conseguir ser utilizada a linguagem SQL, para salvar os dados de agendamento no banco de dados.

Outra dificuldade encontrada foi em relação ao acionamento no protótipo, teve-se o problema que ou era cadastrado o usuário que clicou no botão "ON", ou era ligado o

sistema e acionada a porta no microcontrolador. Para resolver essa situação foi feito um código na linguagem JavaScript que utilizou uma função que ativada dois formulários, sendo que os formulários ativam duas funções simultâneas, uma trabalhando o registro do usuário no site, enquanto a outra trabalha o registro do pedido no microcontrolador Photon. Agora será trabalhado o desenvolvimento do protótipo.

## 5.3 PROTÓTIPO

Para a comprovação da funcionalidade do projeto, optou-se realizar uma simulação de utilização do site, em cima de um protótipo composto dos seguintes elementos: microcontrolador, módulo relé, lâmpadas, bucais, cabo de luz de 1,5 mm, plug e medidor de corrente. O protótipo consiste no microcontrolador, conectado ao módulo relé, o qual está conectado nos quatro pontos de acesso, que serão representados pelas lâmpadas e bucais. O microcontrolador está conectado a uma tomada, a qual está conectada em um ponto de energia. O funcionamento do protótipo ocorre da seguinte forma:

- o usuário realiza o agendamento prévio da utilização de determinado ponto de acesso, por exemplo o agendamento do ponto 1;
- na data e horário selecionadas para a utilização, o usuário acessa o sistema e realiza o ativamento do ponto de acesso no qual ele realizou o agendamento;
- uma mensagem é enviada através do site para o banco de dados, o qual repassa a informação para o microcontrolador, o qual ativa a porta escolhida;
- o *octa acoplador* é acionado pelo microcontrolador, deixando então a corrente passar, a qual é liberada pelo transistor, chegando então ao relé, que é uma chave eletromecânica;
- o relé então é exposto a uma corrente, a qual o faz mudar de status, destravando então a entrada de energia que está atrelada a porta selecionada, assim a corrente consegue chegar ao bucal (representado o ponto de acesso do agendado);
- com a chegada da corrente, tem-se o acendimento da lâmpada como mostra a figura 24 na sequência.

O mesmo acontece com o desligamento do ponto de acesso, uma mensagem é enviada do site para o banco de dados, que transmite ao microcontrolador que foi solicitado o desligamento do sistema, então o microcontrolador derruba a corrente que está passando pelo *octa acoplador*. Logo a corrente não consegue mais chegar ao ponto de acesso, implicando no desligamento do sistema como é visto na figura 25.

Foi feito um circuito retificador de meia onda com filtro capacitivo para a linearização da corrente, transformando corrente alternada em contínua, esse circuito é composto

Figura 24 – Circuito em funcionamento



Fonte:O autor

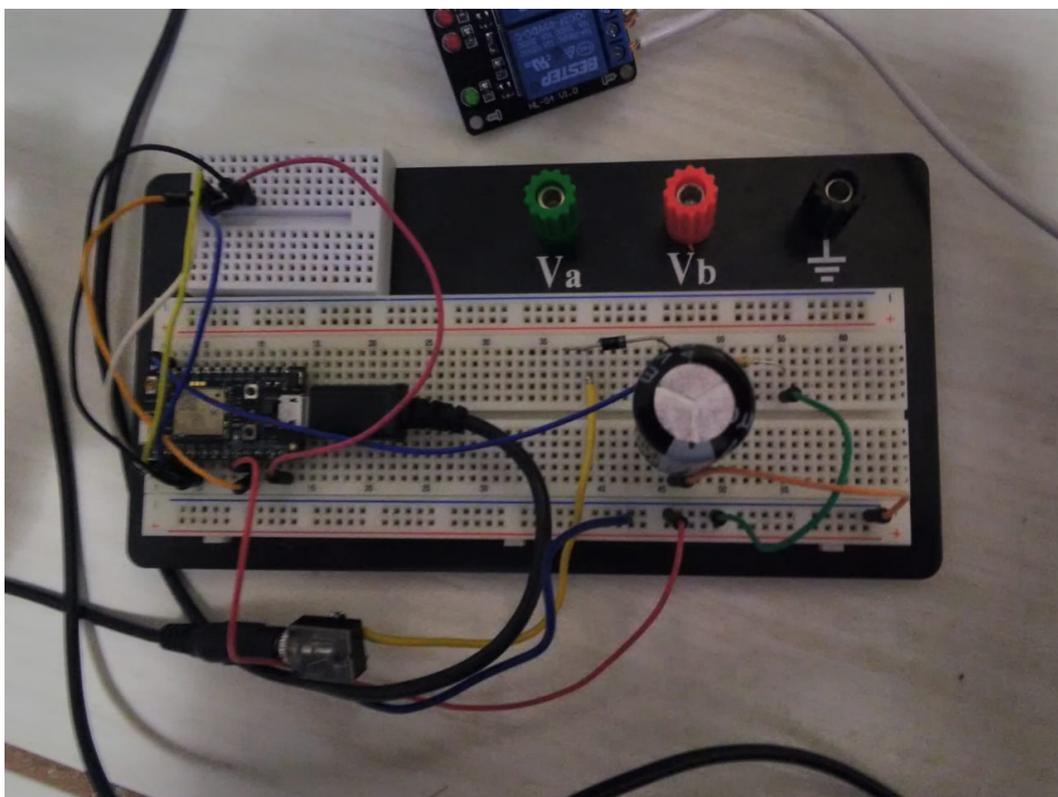
Figura 25 – Circuito desligado



Fonte:O autor

por um diodo mais um capacitor e um resistor, eles em conjunto fazem a linearização, pois o diodo realiza o corte da parte negativa da senoide da corrente, e o capacitor mantém a tensão entre as operações da senoide o circuito montado é representado pela figura 26. Então o valor é medido pelo microcontrolador e enviado para o site da empresa Particle, o qual recebe a leitura e transmite as informações para o servidor via método post, sendo que neste consta o usuário, o ponto de acesso, o somatório da corrente utilizada e o horário de término de utilização do ponto. Após o cadastramento no banco de dados, é processado o valor obtido, multiplicado por 127, por causa do valor de tensão nominal, dividindo por 1.000, para transformar em kilowatt e depois por 3.600, para transformar o valor para horas, obtendo assim o valor do quilowatt-hora, gerando assim o valor de custo, comentado no final do tópico anterior.

Figura 26 – Circuito retificador montado



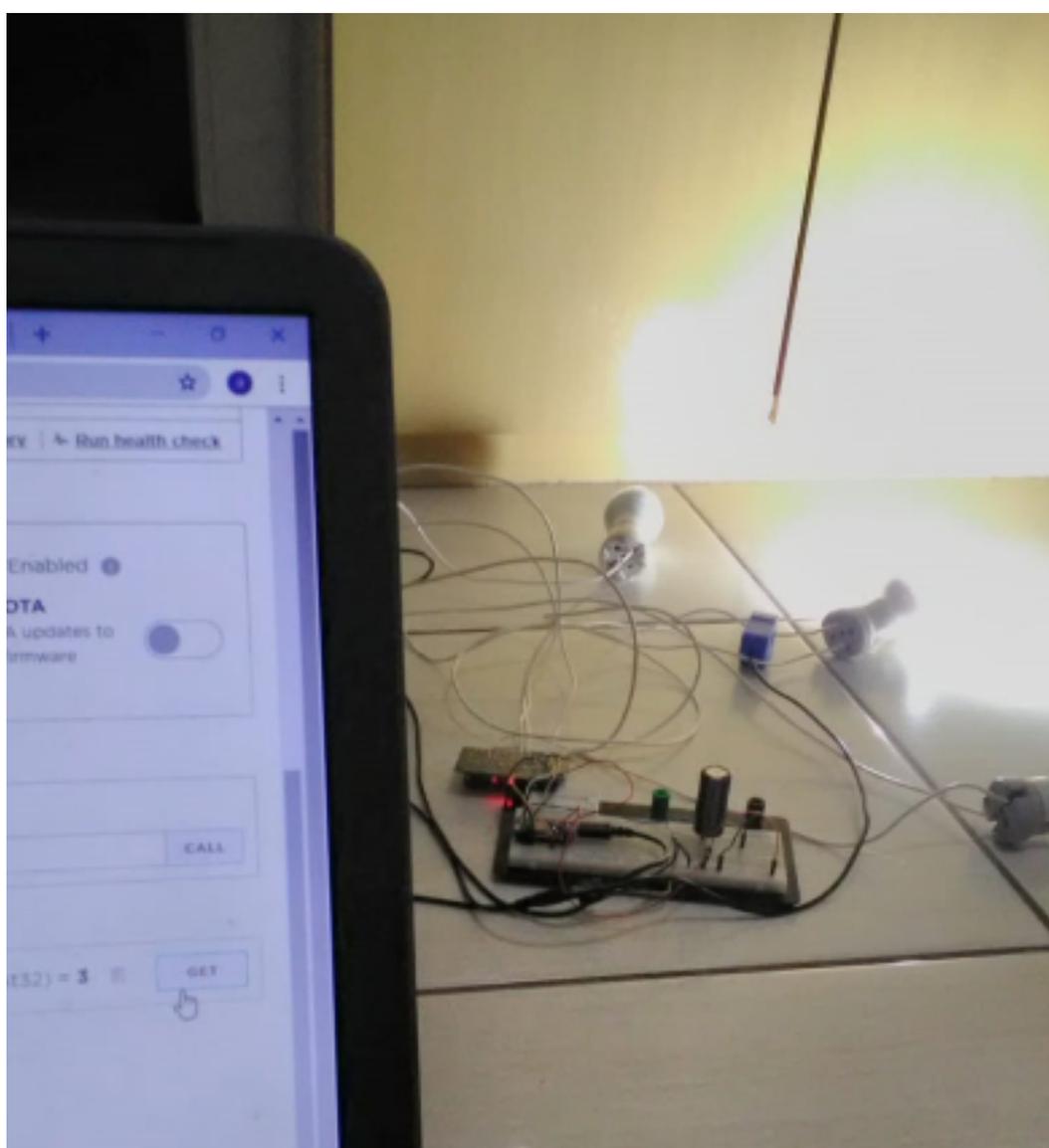
Fonte:O autor

Foi feita também a operação em software sem o uso do sensor de corrente, através do controle de entrada e saída, a grande vantagem do software é o fato do usuário verificar online o valor gasto, fora a diminuição do custo de aquisição do sensor. Este custa em média cinquenta reais, e seus eventuais problemas como, por exemplo, o rompimento do cabo que leva a corrente medida pelo sensor até o microcontrolador, além do problema da aferição do sensor, pois neste caso temos o valor pequeno para a tensão medida no microcontrolador em torno de 2,1 milivolt esse valor é medido pelo microcontrolador com

o valor de 3 como mostra a figura 27. Esse número é dado pelo sensor através da regra de 3, pois esse mede 4096 para 3,3 v então para 2,1 mv o valor seria de 2,6 como o valor aceita apenas inteiros então é passado para o valor maior 3. Vale ressaltar que essa tensão oscila mais devido ao retificador e a sensibilidade do sensor essa pode ser desconsiderada.

Pode-se considerar essa medição como correta, pois como o produto tem uma potencia de 8 watts, e a rede elétrica tem tensão de 127 volts fazendo a divisão obtém-se o valor de 0,063 Amperes como o sensor mede na proporção 1 para 30 isso é 1 volt para cada 30 amperes obtém-se o valor de 2,1 milivolts.

Figura 27 – Medição do sensor de corrente



Fonte:O autor

## 6 RESULTADOS E DISCUSSÕES

O trabalho de conclusão de curso, aqui desenvolvido, tem como resultado a criação de um sistema de agendamento de utilização de pontos de energia, coordenando a criação de um site, com o desenvolvimento de um protótipo que visa o desbloqueio de energia de aparelhos, a partir do conceito de internet das coisas. O desenvolvimento do site se deu a partir da prática da programação, sendo essa feita em várias web linguagens, junto com isso tem-se a utilização de um banco de dados, o qual trabalha no back-end, servindo como servidor, para o site.

Além da utilização das web linguagens, tem-se a transposição do site para um aplicativo de celular, compatível com a plataforma Android, usando a tecnologia bootstrap, que realiza a adequação do tamanho da tela, de acordo com qual a plataforma o usuário estiver utilizando, por exemplo, dentro de um computador, um tablet e um celular, tem-se os mesmos recursos, já que houve o redimensionamento da tela.

Com o site o aplicativo concluídos, deu-se uma maior atenção ao desenvolvimento do protótipo que fundamenta todo esse trabalho. Pensando na liberação dos pontos de energia, tem-se que foi escolhido um sistema utilizando lâmpadas, as quais representam os produtos que encontram-se em áreas de uso compartilhado. Com isso, tem-se que o ativamento é feito a partir de um agendamento feito no site, fazendo com que o protótipo funcione da maneira esperada. O protótipo foi feito a partir do uso de um microcontrolador e um módulo relé que são responsáveis pelo bloqueio, ou não, da energia do sistema, impedindo assim que a energia chegue ao ponto de acesso. A partir do momento que foi feito o ativamento no site tem-se que o microcontrolador terá como objetivo a passagem de corrente pelo circuito até o ato de ligar o ponto de energia.

Durante essa etapa do processo encontraram-se dificuldades de comunicação entre o microcontrolador e o banco de dados, sendo que não era possível efetuar duas funções simultaneamente. Para isso foi utilizada a linguagem javascript, que ativa as duas funções simultaneamente em php, sendo que ela opera cada uma das funções com um espaço de milésimos de segundo entre cada uma. O funcionamento se dá da seguinte maneira: primeiro é efetuado o ligamento do aparelho enviando um comando ao site da Particle com *access token* mais o número do usuário após essa operação é esperado cerca de quatrocentos milissegundos, e na sequência é feita a função "acesso" que realiza o cadastramento referente ao uso daquele ponto específico. Outro problema encontrado em relação ao uso do microcontrolador foi uma atualização feita pela sua desenvolvedora, a qual travou a comunicação entre o site e o microcontrolador, sendo que para esse problema ser resolvido foi atualizado o microcontrolador para a versão mais recente e atualizado o método post, sendo esse método responsável por realizar a requisição para o site. Essa atualização deu-

se através do uso da biblioteca **ajax biblioteca**, sendo ela disponibilizada pela Google, assim o sistema voltou ao seu pleno funcionamento.

Um outro tipo de problema que pode ser apontado, é a medição do consumo de energia que vai para o dispositivo photon, pelo fato de ser muito pequena essa gerava algumas incorrências, além disso o sistema deveria receber o valor medido de segundo em segundo o que sobrecarregava o microcontrolador. Então foi colocado apenas a medição via software que utiliza contador que soma segundo a segundo o valor de potência que está sendo apresentado na especificação do produto e divide o valor por 1000 e 3600 para transformar para kilowatt-hora.

## 7 CONCLUSÃO

Durante o desenvolvimento desse trabalho, pode-se perceber o aumento da importância da IoT para a solução de problemas, atuando em áreas não tecnológicas até na tecnologia de ponta. A solução apresentada para a redução do espaço de moradia, trata de um grande exemplo de uma área não tecnológica, mas que a partir do uso da tecnologia tem tido grandes mudanças, atrelando ao conforto e segurança a seus moradores, através do emprego da IoT no seu planejamento e desenvolvimento. Além disso, a IoT consegue trabalhar na otimização do tempo, sendo esse um dos maiores benefícios para o trabalhador atual.

Tem-se então que foi relacionando a IoT, a redução dos espaços nas residências do futuro, e primando o aproveitamento do tempo livre, que escolheu-se trabalhar o tema do agendamento prévio de utilização de itens eletrônicos dispostos em áreas de uso compartilhado, como lavanderias, por exemplo. O desenvolvimento do site, aplicativo e do protótipo foram pensados em relação a valorização de espaço, tempo e qualidade de vida dos condôminos. Além disso outra problemática que pode ser resolvida com o desenvolvimento desse projeto foi a relação interpessoal entre os condôminos dos condomínios, visto que a pessoa só poderá utilizar o bem se o agendar primeiramente, desta forma reserva seu horário de utilização, assim não há a perda de tempo indo até o local, e verificando que outros usuários já ocuparam todos equipamentos. Além disso, pode-se dimensionar melhor o uso dos equipamentos para esses não terem um tempo ocioso tendo assim um uso contínuo do mesmo.

Outra vantagem para o consumidor é que ele não teria que desembolsar uma grande quantia de dinheiro para comprar um eletrodoméstico, pois ao invés de um eletrodoméstico para cada pessoa em sua área privativa, teria uma área de uso compartilhado que atenderia a mais pessoas com o mesmo equipamento. Pelo fato de muitas pessoas terem acesso a esse bem poderia gerar o problema da má utilização e está pessoa não pagar da forma adequada ao problema que esta gerou em caso com esse tem-se que o administrador do site poderá descobrir quem utilizou o equipamento pela última vez e assim descobrir o causador do problema.

Uma alternativa importante que pode ser levantada a partir desse projeto, devido a individualização do sistema, é o mapeamento de escolha de equipamentos pelos usuários, por exemplo, de marcas distintas, podendo assim gerar uma pesquisa de mercado, implicitamente, entre os usuários do projeto, em relação à marca dos produtos selecionados por eles. Logo uma análise do perfil de cada um dos inquilinos pode vir a ser desenvolvida. Assim gerar uma pesquisa de mercado, que mostra o quanto uma determinada marca está inserida dentro do mercado consumidor, pode-se até fazer análises qualitativas em relação

aos consumidores como a classe econômica, idade e sexo, e sua preferência a determinada marca em relação a outra

# REFERÊNCIAS

- BRASIL banco do. *Internet das coisas*. 2019. Disponível em: <<https://www.bb.intranet10.com.br>>. Acesso em: 10 jul 2019. Citado 2 vezes nas páginas 13 e 14.
- CARRIL, M. *HTML - Passo a Passo*. [S.l.]: clube de autores, 2012. Citado na página 24.
- CONVERSE, T.; PARK, J. *PHP a Bíblia*. [S.l.]: ELSEVIER, 2003. Citado na página 25.
- DEITEL, P. *Android para programadores*. [S.l.]: Bookman, 2015. Citado 4 vezes nas páginas 30, 31, 32 e 33.
- GABARDO, A. C. *Laravel para ninjas*. [S.l.]: Novatec, 2017. Citado na página 27.
- GAMMA, E. et al. *Design Patterns: Elements of reusable object-oriented software*. [S.l.]: Grady Booch, 1994. Citado na página 27.
- GITHUB. *Sintaxe do CSS*. 2017. Disponível em: <<http://tableless.github.io/iniciantes/manual/css/sintaxe-css.html>>. Acesso em: 03 set 2017. Citado na página 24.
- GOOGLE, P. store do. *WebView do sistema Android*. 2018. Disponível em: <<https://play.google.com/store/apps/details?id=com.google.android.webview>>. Acesso em: 01 junho 2018. Citado na página 34.
- GUIMARÃES, C. *Introduç ao a Linguagens de Marcaç ao: HTML, XHTML, SGML, XML*. 2005. Disponível em: <[www.ic.unicamp.br/~celio/inf533/docs/markup.html](http://www.ic.unicamp.br/~celio/inf533/docs/markup.html)>. Acesso em: 01 set 2017. Citado na página 24.
- HORSTMANN, C. S. *Conceitos de computação em Java*. [S.l.]: Bookman, 2009. Citado 2 vezes nas páginas 22 e 23.
- JEFFERSON. *Módulo de relês*. 2012. Disponível em: <<https://www.automalabs.com.br/modulo-de-reles>>. Acesso em: 23 jul 2012. Citado na página 36.
- JOBSTRAIBIZER, F. *Criação de sites com CSS*. [S.l.]: Digerati, 2009. Citado na página 24.
- LOBO, E. J. R. *Criação de sites em PHP*. [S.l.]: Digerati Books, 2007. Citado na página 26.
- MONK, S. *Internet das Coisas: Uma Introdução com o Photon*. [S.l.]: Bookman, 2018. Citado 2 vezes nas páginas 34 e 35.
- RELAY, S. *RELAY ISO9002*. [S.l.]: SONGLE RELAY, 2019. Citado 3 vezes nas páginas 35, 36 e 37.
- SOMMERVILLE. *Engenharia de soft*. [S.l.]: pearson, 2013. Citado 7 vezes nas páginas 16, 17, 18, 19, 20, 21 e 22.

---

SPURLOCK, J. *Bootstrap : Responsive Web*. [S.l.]: O'REILLY, 2013. Citado na página 25.

TANENBAUM, A. S. *Sistemas Operacionais Modernos*. [S.l.]: pearson, 2010. Citado 3 vezes nas páginas 13, 15 e 16.

WIDENIUS, M. *Manual de Referência d MySQL 4.1*. 2010. Disponível em: <<https://dev.mysql.com>>. Acesso em: 01 out 2017. Citado 4 vezes nas páginas 27, 28, 29 e 30.