



**ANTONIO MENEGHETTI FACULDADE - AMF**  
**CURSO DE SISTEMAS DE INFORMAÇÃO**

**JEAN EMÍLIO WILHELM**

**SISTEMA DE ALARME DE PORTA UTILIZANDO ARDUINO COM**  
**MONITORAMENTO POR DISPOSITIVO MÓVEL: UM ESTUDO DE**  
**CASO NO RECANTO MAESTRO**

RESTINGA SECA/RS

2019

**JEAN EMÍLIO WILHELM**

**SISTEMA DE ALARME DE PORTA UTILIZANDO ARDUINO COM  
MONITORAMENTO POR DISPOSITIVO MÓVEL: UM ESTUDO DE CASO NO  
RECANTO MAESTRO**

Trabalho de Conclusão do Curso de Sistemas de  
Informação

Orientador: Prof<sup>o</sup>. Dr. Felipe Becker Nunes

RESTINGA SECA/RS

2019

## AGRADECIMENTOS

Em primeiro lugar, agradeço ao meu professor e orientador **Felipe Becker Nunes** por aceitar conduzir o meu trabalho de pesquisa e ter realizado este com tanta maestria, dedicação e paciência. Apesar de todos os obstáculos sempre me incentivou e auxiliou para a execução deste trabalho.

Agradeço também, aos meus **amigos, colegas e familiares** que tiveram muita paciência ao longo desta etapa, sempre me encorajando e apoiando em minhas decisões, fazendo com que esta fase me propiciasse muito conhecimento que auxiliarão muito em minha vida, principalmente profissional.

A todos os meus **professores** do Curso de Sistemas de Informação, por compartilhar de suas experiências e, com muita maestria, transmitir seus conhecimentos.

Por fim, a **Faculdade Antonio Meneghetti** por todo suporte prestado durante esses anos de graduação, auxiliando em meu desenvolvimento técnico, pessoal e profissional.

A todos, meu mais sincero obrigado!

FACULDADE ANTONIO MENEGHETTI

Jean Emílio Wilhelm

SISTEMA DE ALARME DE PORTA UTILIZANDO ARDUINO COM  
MONITORAMENTO POR DISPOSITIVO MÓVEL: UM ESTUDO DE CASO NO  
RECANTO MAESTRO

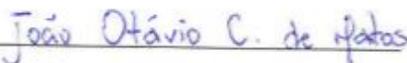
Trabalho de Conclusão de Curso-Monografia, apresentado como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação, Curso de Graduação em Sistemas de Informação, Faculdade Antonio Meneghetti-AMF.

Orientador: Prof. Dr. Felipe Becker Nunes



Prof. Dr. Felipe Becker Nunes

Orientador do Trabalho de Conclusão de Curso  
Antonio Meneghetti Faculdade



Prof. Ms. João Otávio Cadó de Matos

Membro da Banca Examinadora  
Antonio Meneghetti Faculdade



Prof. Ms. Mateus Schott

Membro da Banca Examinadora  
Antonio Meneghetti Faculdade

Restinga Seca, RS, 10 de dezembro de 2019

“Depois de ações eficazes, inicie uma reflexão silenciosa. A partir da reflexão silenciosa virá uma ação ainda mais eficaz.”.

Peter F. Druck

## RESUMO

Um dos problemas que assombra qualquer brasileiro hoje em dia, em qualquer região é a segurança. Em uma crescente contínua, a automação residencial é um termo que está se tornando popular entre os brasileiros e cada vez mais acessível. Esse tipo de automação tem como objetivo proporcionar conforto aos seus usuários, uma vez que ações simples como controle de iluminação, climatização, segurança, entre outros, podem ser realizadas de forma prática e rápida através do controlador do sistema.

Frente a este fato, o autor identificou a segurança residencial como um aspecto que deixa a desejar no local que reside, além disso, entendeu que não é uma preocupação apenas sua, mas sim de demais moradores da localidade de Recanto Maestro, que possui um campus universitário frequentado por milhares de pessoas diariamente. A partir disto, propôs a realização de um estudo de caso nesta localidade, desenvolvendo um sistema de alarme de porta a partir de ferramentas *open source*, tais como Arduino para o controle de porta, e as tecnologias React, React Native e Node.js, além do banco de dados MySQL para as aplicações de cadastros básicos e controle do alarme.

Utilizando o método de pesquisa qualitativa realizou-se um estudo teórico sobre as tecnologias e ferramentas que foram utilizadas, partindo então para o desenvolvimento do projeto. A análise dos dados aconteceu a partir de resultados da execução de cenários de testes, previamente descritos, bem como a comparação com alguns projetos semelhantes. Os resultados obtidos neste projeto foram parciais, uma vez que não foi possível concluir o desenvolvimento do aplicativo *mobile* que faria o controle do alarme de porta. Com relação aos resultados coletados nas outras partes do sistema, entendeu-se como satisfatória, gerando também ideias para projetos futuros.

**Palavras-chave:** Alarme Residencial; Automação Residencial; Segurança.

## ABSTRACT

One of the problems that haunts any Brazilian today, in any region is security. In a continually growing, home automation is a term that is becoming popular among Brazilians and increasingly accessible. This type of automation aims to provide comfort to its users, since simple actions such as lighting control, climate control, safety, among others, can be performed quickly and conveniently through the system controller.

In light of this fact, the author identified residential security as an aspect that is lacking in the place where he lives. He also understood that it is not only his concern but that of other residents of the locality of Recanto Maestro, which has a campus. attended by thousands of people daily. From this, he proposed to conduct a case study in this locality, developing a door alarm system from open source tools such as Arduino for door control, and React, React Native and Node.js technologies, plus the MySQL database for basic registration and alarm control applications.

Using the qualitative research method, a theoretical study was carried out about the technologies and tools that were used, starting from the project development. The data analysis was based on the results of the execution of test scenarios previously described, as well as the comparison with some similar projects. The results obtained in this project were partial, as it was not possible to complete the development of the mobile application that would control the door alarm. Regarding the results collected in other parts of the system, it was considered satisfactory, also generating ideas for future projects.

**Keywords:** Residential Alarm; Home Automation; Security.

## LISTA DE ILUSTRAÇÕES

<b>Figura 1.</b> Utilização de Web Frameworks por desenvolvedores em 2019	27
<b>Figura 2.</b> Placa Wemos D1 R2 WiFi ESP8266	46
<b>Figura 3.</b> Configurações Arduino IDE	47
<b>Figura 4.</b> Sensor Magnético com fio	48
<b>Figura 5.</b> LEDs, Resistores, jumper e protoboard	49
<b>Figura 6.</b> Design do protótipo realizado no Fritzing	50
<b>Figura 7.</b> Protótipo utilizado para os testes	51
<b>Figura 8.</b> Stack de Arquitetura do Projeto	52
<b>Figura 9.</b> Diagrama de Caso de Uso - Sistema Web	56
<b>Figura 10.</b> Diagrama de Caso de Uso - Aplicativo Mobile	57
<b>Figura 11.</b> Modelo Relacional	58
<b>Figura 12.</b> Estrutura do Backend	59
<b>Figura 13.</b> Rotas Para Cada Função	60
<b>Figura 14.</b> Insert no banco de dados	61
<b>Figura 15.</b> Insert no banco de dados	62
<b>Figura 16.</b> Cadastrar Pessoa	63
<b>Figura 17.</b> Listagem de Pessoas	64
<b>Tabela 1.</b> Cenários de Testes - Aplicativo <i>Mobile</i>	40
<b>Tabela 2.</b> Cenários de Testes - Sistema Web	42

## LISTA DE ABREVIATURAS

AR - Automação Residencial  
BDD - *Behaviour Driven Development*  
CRUD - *Create-Retrieve-Update-Delete*  
CSS - *Cascading Style Sheets*  
GND - *graduated neutral density filter*  
IDE - *Integrated Development Environment*  
HTML - *Hypertext Markup Language*  
JS - JavaScript  
MVC - *Model-View-Controller*  
RN - React Native  
PHP - *Hypertext Preprocessor*  
SGBD - *Data Base Management System*  
SQL - *Structured Query Language*  
TI - Tecnologia da Informação  
USB - *Universal Serial Bus*  
XML - *Extensible Markup Language*  
WiFi - *Wireless Fidelity*

## SUMÁRIO

ANTONIO MENEGHETTI FACULDADE - AMF	1
RESUMO	6
ABSTRACT	7
LISTA DE ABREVIATURAS	9
<b>INTRODUÇÃO</b>	<b>12</b>
1.1 OBJETIVOS	14
1.1.1 OBJETIVO PRINCIPAL	14
1.1.2 OBJETIVOS ESPECÍFICOS	14
1.2 JUSTIFICATIVA	14
<b>ABORDAGEM TEÓRICA</b>	<b>17</b>
2.1 AUTOMAÇÃO RESIDENCIAL	17
2.1.1 SEGURANÇA NA AUTOMAÇÃO	19
2.2 MICROCONTROLADORES	20
2.2.1 ARDUINO	22
2.3 DESENVOLVIMENTO DE SOFTWARE	25
2.3.1 FRONTEND	25
2.3.1.1 REACT	26
2.3.1.2 REACT NATIVE	28
2.3.2 BACKEND	29
2.3.2.1 JAVASCRIPT	30
2.3.2.2 NODE.JS	31
2.3.3 SGBD	32
2.3.3.1 MYSQL	32
2.4 TRABALHOS RELACIONADOS	33
<b>METODOLOGIA</b>	<b>36</b>
3.1 DESIGN DO ESTUDO	37
3.2 INSTRUMENTOS DE COLETA DE DADOS	39
3.3 ANÁLISE DOS DADOS	44

<b>DESENVOLVIMENTO</b>	<b>45</b>
4.1 COMPONENTES E ESQUEMÁTICO	45
4.1.1 WEM 45	
4.1.2 SENSOR MAGNÉTICO COM FIO	47
4.1.3 COMPONENTES COMPLEMENTARES	48
4.1.4 ESQUEMÁTICO E PROTÓTIPO	49
4.2 SISTEMA BACKEND	53
4.3 SISTEMA FRONTEND	63
<b>ANÁLISE E DISCUSSÃO DOS RESULTADOS</b>	<b>66</b>
<b>CONSIDERAÇÕES FINAIS</b>	<b>68</b>
<b>REFERÊNCIAS</b>	<b>70</b>

# 1 INTRODUÇÃO

De acordo com a Associação Brasileira de Automação Residencial (AURESIDE, 2016), cerca de 300 mil residências brasileiras possuem automação. Este número é considerado baixo se levado em consideração o número de residências existentes. Apontado por esse estudo como um dos principais motivos desse baixo número, o alto custo destes projetos é um dos principais desafios a serem superados pelos estudantes e profissionais de Tecnologia da Informação (TI).

O conceito de automação é muito utilizado quando há a necessidade no aumento da produtividade ou diminuição do esforço humano ao realizar determinada atividade. Do latim, *automatus*, refere-se a “que se move sozinho”, ou seja, agilizam as tarefas, geralmente sem a interferência humana. O termo automação se tornou popular por volta de 1950, geralmente associada a automação industrial e muito utilizada para explicar a robotização dos processos industriais. Por volta de 1960 a automação foi associada então a residências e temas futuristas, a partir do desenho animado norte americano *The Jetsons*, que traziam consigo o dia-a-dia de uma família futurista. Naquele tempo, isso não passava de imaginação. Segundo Muratori, diretor executivo da Associação Brasileira de Automação Residencial e Predial (Aureside), a “automação residencial é a integração dos diversos sistemas existentes em um lar.”

Esse tipo de automação tem como objetivo proporcionar conforto aos seus usuários, uma vez que ações simples como controle de iluminação, climatização, segurança, entre outros, podem ser realizadas de forma prática e rápida através do controlador do sistema. O quanto esses recursos poderão ajudar o usuário dependerá de seu estilo de vida, dos gostos pessoais e de seus recursos disponíveis (MARTE, 1995, p. 15).

No Brasil, a automação residencial vem tomando seus primeiros passos nos últimos anos. Sendo que, atualmente, representa um mercado de consumo que fatura aproximadamente R\$500 milhões de reais por ano e tem apresentado um crescimento contínuo de 30% ao ano. O principal fator que alavanca essa massa de consumo é o anseio por maior conforto e segurança (TECHINBRAZIL, 2015).

A segurança residencial pode ser feita a partir de diversos sensores, como sensores de fumaça, umidade, fogo, movimento, entre outros. Estes quando detectam algum sinal de ativação, neste caso, o sensor de movimento detectar algum movimento, pode acionar um alarme, ou outro dispositivo configurável, e disparar uma mensagem ao usuário. Também é possível realizar o controle em tempo real com o uso de dispositivos móveis, caso hajam vários instalados e conectados a um sistema. A natural busca por um lugar seguro pra se morar, em face dos crescentes problemas sociais que afligem o mundo de hoje, vem demandando o desenvolvimento de sistemas de monitoramento e segurança residencial, cada vez mais onipresentes, inteligentes e com mais recursos (PEREIRA, 2007).

Segundo o IBGE (2018), no Brasil cerca de 181 milhões de pessoas acessam a internet pelo celular. Com base nesta informação, compreende-se que o usuário conseguirá identificar de forma mais rápida uma possível invasão quando um sistema lhe enviar uma notificação. Este sistema de segurança, vinculado a um aplicativo intuitivo, pode sugerir ações a serem tomadas em possíveis casos de invasões, como por exemplo, disponibilizar uma lista de contatos de emergência que podem ser acionado a partir de poucos comandos.

Além disso, a integração entre os processos a serem controlados e o aproveitamento total da central de automação, em questão de capacidade do sistema, são fatores importantes a serem considerados, visto que é necessária apenas uma central para controlar várias aplicações, o que determina um melhor custo-benefício do que implementá-las separadamente (BEGHINI, 2013).

Portanto, o propósito deste projeto é realizar a análise e estudo de tecnologias e ferramentas *open-source* e atuais do mercado. A partir deste, realizar o desenvolvimento de um protótipo para o controle de portas e monitoramento a partir de um aplicativo *mobile*.

Para isso, foi realizado um estudo de caso na localidade de Recanto Maestro (Restinga Seca - RS), onde foi identificado a necessidade de realizar o controle de porta de uma residência, visto que o local possui fluxo de pessoas muito grande, visto que é o campus da Faculdade Antonio Meneghetti. A localidade foi escolhida pelo autor por possuir sua residência nesta localidade, assim então podendo validar a ideia a partir de testes reais, possibilitando então

realizar o aprimoramento do protótipo, melhorar a usabilidade do produto, e também comprovar a confiabilidade do mesmo, podendo vir a se tornar um produto a ser comercializado futuramente.

## **1.1 OBJETIVOS**

### **1.1.1 OBJETIVO PRINCIPAL**

O presente trabalho tem como objetivo realizar a análise e estudo de tecnologias e ferramentas *open-source* e atuais do mercado, assim como desenvolver um sistema de controle de porta, utilizando a plataforma Arduino, bem como realizar seu monitoramento a partir de um aplicativo *mobile*.

### **1.1.2 OBJETIVOS ESPECÍFICOS**

- Realizar estudo de novas tecnologias e ferramentas;
- Utilizar *software e hardware open source* para garantir o baixo custo do projeto;
- Realizar a integração das diferentes plataformas e tecnologias utilizadas na execução do projeto;
- Promover a segurança com o monitoramento remoto dos acessos da residência;
- Implantar um protótipo funcional, a fim de coletar dados para análise e aprimoramento do protótipo com base em testes reais.

## **1.2 JUSTIFICATIVA**

Durante a graduação na área de Tecnologia da Informação é possibilitada a criação de inúmeras aplicações, para diferentes dispositivos e com diversos objetivos. O autor, diante deste

cenário, tem como desafio unir estes diversos projetos desenvolvidos em seu período de graduação, a fim de criar um produto útil, com ferramentas *open-sources*, que promova comodidade e segurança no contexto que está localizado.

A necessidade de ambientes automatizados se torna cada vez maior, a partir da evolução da tecnologia. Segundo Volpato (2012), dentro do contexto de automação residencial, uma das áreas mais importantes tratadas, é a questão da segurança nas casas e apartamentos. Muitas tecnologias integradas com sistemas computacionais vêm sendo desenvolvidas visando garantir maior segurança nos ambientes residenciais.

Equipamentos como alarmes, câmeras de vigilância, interfones, portões eletrônicos, entre outros, estão cada vez mais presentes nas residências das pessoas. Aliar o controle de dispositivos de segurança de forma remota ao devido conforto que essa teoria defende ainda é um desafio nos dias de hoje.

Aliado a isto, seu alto custo faz com que inúmeras residências optem por não aderir a esta evolução. Uma das motivações do autor é a possibilidade de iniciar a automação de sua própria residência e a partir dos testes, realizar o aprimoramento, bem como a continuidade dos estudos, a fim de obter cada vez mais a redução nos custos e manter o sistema confiável.

Apesar deste crescimento, a automação residencial possui um sério entrave relacionado à falta de padronização do protocolo utilizado. Até o momento não encontrou-se uma solução que seja ótima e atenda globalmente às necessidades do mercado, por isso muitos fabricantes preferem investir em soluções proprietárias para comercializarem em conjunto com os seus produtos, fatos, inclusive, que faz com que o valor agregado do produto aumente muito (SILVA, 2014).

Sistemas automatizados estão sendo cada vez mais presentes no nosso cotidiano. Imóveis modernos, tanto comerciais como residenciais, geralmente contam com sistemas integrados de controle que possibilitam a realização de inúmeras atividades com facilidade, podendo ainda ser citados vantagens pessoais como conveniência, controles de iluminação e temperatura, verificação e segurança de janelas ou portas remotamente.

De forma adjunta à necessidade real de haver um monitoramento em tempo real, os *smartphones* são cada vez mais comuns, e com isso a criação de aplicativos se torna mais atrativa para jovens estudantes de TI. Perfil este do autor que acompanha a crescente onda em aplicativos

móveis que visam realizar o monitoramento remoto de diversas aplicações, a fim de facilitar o dia a dia dos usuários. De acordo com a 30ª Pesquisa Anual de Administração e Uso de Tecnologia da Informação nas Empresas, realizada pela Fundação Getúlio Vargas de São Paulo (FGV-SP), há hoje 230 milhões de celulares ativos no País. Já o número de computadores, notebooks e tablets em uso no Brasil é de 180 milhões<sup>1</sup>.

O mercado de aplicativos para *smartphones* mostra-se em crescimento rápido. Segundo Martins (2013), este mercado é disputado por diferentes plataformas tecnológicas, incluindo sistemas operacionais e plataformas de desenvolvimento, gerando uma variedade de soluções no mercado. Com sistemas operacionais mais avançados, é possível desenvolver aplicativos melhores, com cada vez mais recursos e serviços disponíveis ao usuário.

Frente a este fato, um dos principais desafios propostos foi compreender uma solução para a necessidade de segurança residencial ainda não suprida na localidade de Recanto Maestro, que possui um campus universitário frequentado por milhares de pessoas vindas de diversas cidades vizinhas, que assim nos revela um ambiente de insegurança pelo alto fluxo de pessoas desconhecidas, frequentando o local. Assim então, foi realizado um estudo de novas tecnologias open source, tornando o projeto autêntico e adaptável com maior facilidade e custo mais acessível para a idealização do protótipo.

---

<sup>1</sup> Brasil tem 230 milhões de smartphones em uso. Disponível em: <https://epocanegocios.globo.com/Tecnologia/noticia/2019/04/brasil-tem-230-milhoes-de-smartphones-em-uso.html>

## 2 ABORDAGEM TEÓRICA

Neste capítulo, serão abordados de forma introdutória, alguns conceitos importantes para a compreensão funcional e técnica do projeto, bem como a qual conceito tecnológico a pesquisa proposta está inserida. Inicialmente será explanado sobre a automação residencial e os seus benefícios para a segurança, seguido dos conceitos de microcontroladores e sensores usados na construção do *hardware*. Na sequência, as tecnologias usadas para a criação do *software*: React, React Native, JavaScript e Node.js, bem como seu SGBD.

### 2.1 AUTOMAÇÃO RESIDENCIAL

Conforme Bolzani (2004), a automação residencial (AR) é um ramo da automação predial que tem como objetivo o controle de processos no universo doméstico, com isso, reduzindo a necessidade de intervenção humana no gerenciamento de equipamentos eletroeletrônicos através de sistemas de controle.

Cada nova tecnologia traz acoplado um novo vocabulário. Quando o assunto é residência inteligente, não é diferente: casa automática, casa 24 inteligente, automação residencial, *retrofitting*, domótica, etc., mas tudo pode ser resumido a uma só palavra: conforto. Se um sistema eletrônico instalado em um ambiente não oferecer conforto ao usuário, em semanas ele vai ser desligado e deixado de lado [...]. Os equipamentos devem unificar os controles e processos tornando tudo mais simples. Mas é o desejo do usuário que deve prevalecer e não a do PC. A casa automática pode ajudar nas tarefas diárias que tomam muito tempo ou evitar preocupações tais como esquecer as janelas abertas quando a previsão do tempo avisou que ia chover (BOLZANI, 2004).

A automação residencial está dividida em três graus ou classes de integração, que podem determinar o grau de automação de uma edificação (PINHEIRO, 2004):

- **Sistemas Autônomos:** possuem funcionalidades somente de acionar ou desligar dispositivo ou subsistemas. Nessa classificação nenhum dispositivo ou subsistema tem relação um com o outro (TERUEL, 2008);

- Sistemas Integrados: possuem múltiplos subsistemas integrados a um único controlador. A limitação deste sistema está no fato que cada subsistema deve ainda funcionar unicamente na forma a qual seu fabricante pretendia. Basicamente trata-se apenas de controle remoto estendido a diferentes locais. O processamento pode ser centralizado na central de automação ou distribuído pela rede (TERUEL, 2008);
- Sistemas de automação complexos: nível total de automação na residência, trazendo assim um alto grau de complexidade e a necessidade da residência ser projetada com o intuito de ser totalmente automatizada (QUINDERÉ, 2009).

O que muitas pessoas desconhecem, é que nos dias de hoje, já existem muitos dispositivos em suas residências. Aparelhos como micro-ondas, cafeteiras, geladeiras e televisores possuem diversos *microchips* com diversas funções. Porém para acessar esses aparelhos é necessário se deslocar ao mesmo ou através de controle remoto. A automação residencial vem com o objetivo de disponibilizar o acesso, controle e monitoramento destes equipamentos a distância.

A infraestrutura doméstica padrão é composta por funções independentes, redes não compatíveis, falta de uniformidade e equipamentos limitados, responsáveis por problemas como: multiplicidade de rede e cabos, manutenção cara e complicada, impossibilidade de automação global e dificuldade de interligar novos serviços e redes (Bolzani, 2004).

Ainda conforme Bolzani (2004), a proposta de infraestrutura automatizada centralizada relaciona alguns pontos positivos, entre eles está o maior conforto e automatização de serviço, barateamento de equipamentos e processos, simplificação da rede, acesso a informação de qualquer ponto da casa, economia de energia e auxílio na administração da residência com constante supervisão do conjunto de equipamentos.

Outro ponto positivo da proposta de automação é a segurança. Aliada ao conforto, a automação residencial fornece a possibilidade de implantar travas controladas por senhas ou por aplicações, identificação de fumaça, fogo, água, mudanças de temperatura e umidade, sinalizando possíveis sinistros e vazamentos. Além disso, é possível realizar o monitoramento em tempo real

com câmeras e identificação de movimento, a partir da captação de sinais infravermelhos com sensores, sinalizando com alarmes sonoros ou enviando informações às centrais.

### **2.1.1 SEGURANÇA NA AUTOMAÇÃO**

Com o incremento do uso das redes, da difusão da internet e da padronização de algumas soluções de hardware e software, criou-se um ambiente com consideráveis possibilidades de ataques virtuais aos sistemas de TI. Na visão de especialistas, as redes inteligentes introduzirão novos riscos de segurança relacionados com as suas necessidades de automação, comunicação e novas tecnologias (BAUMEISTER, 2010).

Sistemas auxiliados por sensores permitem detectar fuga de gás, inundações, incêndios, invasões e outros tipos de ameaças logo em seu princípio. Quando esses sistemas possuem automação e controle a distância, é possível tomar decisões sobre estes incidentes, facilitando o atendimento ao mesmo ou até controlando a situação a partir de alarmes sonoros, corte na energia elétrica ou notificação aos responsáveis.

Os controladores gerenciam os dispositivos automatizados (sensores e atuadores), monitoram informações dos sensores, podendo enviar comandos para que um atuador ative ou desative algum equipamento. De maneira geral podem possuir interfaces independentes, na forma de um controle remoto, ou serem sofisticadas centrais de automação (ALMEIDA, 2009).

Os sensores, são os dispositivos que detectam estímulos, medem e monitoram grandezas físicas e eventos (temperatura, umidades, etc.), convertendo-as em um valor passível de manipulação por sistemas computacionais. São eles que encaminham as informações aos controladores sobre algum evento, para que os controladores possam enviar os comandos adequados para os atuadores (ALMEIDA, 2009).

Esses sensores, podem ser combinados de diversas formas, monitorando de forma automatizada os mais variados aspectos em residências, condomínios e até mesmo ambientes de fácil acesso, como áreas comerciais e ambientes externos de nossas residências. Combinados e

controlados por uma ou mais centrais de comando, estes trabalham enviando os dados, independente da detecção ou não de não conformidades com as aplicações de controle, ou, até mesmo empresas terceirizadas que prestam serviço de monitoramento à invasão e incidentes, muito comum em condomínios e hotéis.

Segundo Brugnari e Maestrelli (2010), os atuadores atendem a comandos que podem ser manuais ou automáticos, ou seja, qualquer elemento que realize um comando recebido de outro dispositivo, com base em uma entrada ou critério a ser seguido. Existem atuadores para portas, janelas, persianas, fechadura magnética, sirene, indicadores luminosos, etc. O barramento é o meio físico responsável pelo transporte das informações (rede elétrica, telefônica etc.) (CASADOMO, 2010).

Com essa informação fornecida pelos sensores, identificou-se a necessidade do desenvolvimento de interfaces que tratem e traduzam essa informação disponibilizando dados para monitoramento em tempo real desses dispositivos e dos locais que estão instalados. As interfaces são os dispositivos ou mecanismos (navegador de internet, celular, painéis, controles remotos, interruptores etc.) que permitem ao usuário visualizar as informações e interagir com o sistema de automação (CASADOMO, 2010).

## **2.2 MICROCONTROLADORES**

Segundo Reis (2015), um microcontrolador é um chip que contém praticamente um pequeno computador embutido, de baixo custo, incluindo núcleo de processamento, memórias RAM/ROM/Flash, periféricos de E/S, conversores de sinais, gerador de clock, entre outros. Os microcontroladores são projetados para uso em aplicações embarcadas e outros sistemas

específicos, tais como brinquedos, sistemas de controle remoto, máquinas industriais e comerciais e também eletrodomésticos.<sup>2</sup>

Aureliano (2017) resume microcontroladores como componentes capazes de executar programas que fazem o uso de seus recursos internos, como a leitura e escrita digital na forma de valores de tensão em suas portas, além da comunicação com componentes externos, permitindo expansão de suas capacidades<sup>3</sup>.

Tipicamente, um microcontrolador caracteriza-se por incorporar no mesmo encapsulamento um microprocessador (com a finalidade de interpretar as instruções de programa e processar dados), memória de programa (com a finalidade de armazenar de maneira permanente as instruções do programa), memória de dados (com a finalidade de memorizar os valores associados com as variáveis definidas no programa), uma série de pinos de entrada/saída (com a finalidade de realizar a comunicação do microcontrolador com o meio externo) e vários periféricos (tais como temporizadores, controladores de interrupção, temporizadores cão de guarda (WatchDog Timers – WDTs), comunicação serial, geradores de modulação por largura de pulso ou de PWM (Pulse Width Modulation), conversores analógico/digital etc.), fazendo com que o hardware final fique extremamente complexo (MARTINS, 2005).

De acordo com Penido e Trindades (2013), um microcontrolador é um circuito integrado capaz de realizar operações lógicas e possui internamente os seguintes dispositivos:

- Uma CPU, na qual a finalidade é interpretar as instruções de programa;
- Uma memória PROM cuja finalidade é gravar as instruções do programa;
- Uma memória RAM (Random Access Memory ou Memória de Acesso Aleatório) empregada com a finalidade de memorizar as variáveis utilizadas pelo programa;
- Um barramento de I/O para controlar dispositivos externos ou receber impulsos de sensores, interruptores;
- Um conjunto de dispositivos auxiliares ao funcionamento, ou seja, gerador de clock, contadores, USART para comunicação.

---

<sup>2</sup> Introdução aos Microcontroladores. Disponível em:

<http://www.bosontreinamentos.com.br/eletronica/eletronica-geral/introducao-aos-microcontroladores/>

<sup>3</sup> Microcontroladores. Disponível em: <https://fiozera.com.br/microcontroladores-914a59cbf7de>

Além disso, os microcontroladores, são capazes de realizar cálculos aritméticos através de amostras, e/ou dados recebidos, tomar decisões, também podem ser implantados em aplicações de inteligência artificial, Internet das Coisas (IoT), processamento de imagem e som e uma infinidade de aplicações, tornando-se um dos grandes responsáveis pela evolução tecnológica cotidiana (AGNOL, 2018).

Com o avanço da tecnologia na área, foi possível reduzir os custos de produção dos microcontroladores, fazendo com que se tornasse uma das melhores opções de investimento quando comparada a capacidade de processamento de dados com o baixo custo a ser investido nesta aplicação.

A partir desta breve compreensão sobre microcontroladores, daremos início à compreensão da Plataforma Arduino, uma placa que consiste basicamente em um controlador programável, com entradas e saídas, a fim de enviar e receber dados dos sensores.

### **2.2.1 ARDUINO**

O Arduino surgiu na Itália, recentemente em 2005, pelo professor chamado Massimo Banzi, visando ensinar de forma interativa eletrônica e programação de computadores para seus alunos de design, para que eles usassem em seus projetos de arte, interatividade e robótica (AGNOL, 2018). Ainda segundo o autor, porém, ensinar programação e eletrônica à pessoas que não são da área era uma tarefa difícil, e outra dificuldade era a inexistência de placas poderosas e baratas no mercado (AGNOL, 2018).

Pensando nisso, Massimo e David Cuartielles decidiram criar sua placa própria, com ajuda do aluno de Massimo, David Mellis, que ficou encarregado de desenvolver a linguagem de programação do Arduino. Várias pessoas conseguiram utilizar o Arduino e fazer diversas atividades de forma fácil e mais em conta sendo utilizada em atividades didáticas até os dias atuais (BEGHINI, 2013).

Com o conceito de hardware e software livre, o Arduino se difundiu com muita facilidade, se tornando uma das principais plataformas de prototipação de eletrônicos autônomos.

O fato de ser *open source* e fácil aquisição, fez com que se criassem diversos fóruns que debatem o uso do Arduino, auxiliando em problemas que pessoas possuem durante seu desenvolvimento, bem como as mais diversificadas ideias sobre seu uso.

Isto auxiliou para que cada vez mais ficasse conhecido e deixando alunos de TI e demais áreas com curiosidade sobre o uso dessa plataforma e as atividades que podem ser automatizadas de diversas formas. Para realizar estas atividades, o Arduino possui como sua peça principal, um microcontrolador de 28 pinos produzido pela Atmel, uma das maiores fabricantes de microcontroladores. Esse microcontrolador possui uma memória para receber dados e toda a eletrônica necessária para os pinos de entrada e saída (MONK, 2013).

O Arduino, portanto, consiste na junção deste microcontrolador a uma placa de desenvolvimento de hardware aberto. Todos os arquivos, projetos e diagramas da placa estão disponíveis para toda e qualquer pessoa que queira produzir e comercializar sua placa Arduino (MONK, 2013).

Uma outra vantagem proporcionada pelo Arduino é uma grande variedade de extensões, chamadas de Shields, que agregam funcionalidades a placa. Alguns exemplos de Shields que pode-se listar são Ethernet, Motor, USB Host e Wireless SD (ARDUINO, 2014). Nesta sessão de vantagens oferecidas pelo Arduino, é um dever de qualquer autor citar os sensores e atuadores.

Moraes (2003) explica sensores como sendo transdutores, ou seja, são conversores de grandezas físicas em sinais elétricos correspondentes. Sensores são dispositivos que trabalham com medidas de grandezas físicas, como: temperatura, pressão, presença, umidade, intensidade luminosa, entre outros. As grandezas medidas pelos sensores são combinadas a fim de obter informações sobre o meio físico, onde estão presentes.

Em geral os sensores atuam transformando partes de uma grandeza física normalmente em um sinal elétrico, que por sua vez pode ser interpretado por certos equipamentos eletrônicos (BORGES & DORES, 2010). Esses sensores são classificados em dois tipos:

- Sensores Externos: Lidam com observações de aspectos do mundo exterior ao robô, como por exemplo, sensores de contato, de proximidade, de força, de

distância, de laser, de ultrassom, de infravermelhos e sensores químicos (RIBEIRO, 2004).

- Sensores Internos: Estes fornecem informação sobre os parâmetros internos do robô, como por exemplo, a velocidade ou sentido de rotação de um motor, ou o ângulo de uma junta, como por exemplos potenciômetros, codificadores (encoders) e os sensores inerciais como acelerômetros, giroscópios, inclinômetros e bússolas (RIBEIRO, 2004).

Atuadores, assim como os sensores são um tipo de transdutor que transforma uma forma de energia em outra, porém, ao invés de transformar partes de uma grandeza física em um sinal elétrico, ele transforma um sinal elétrico em uma grandeza física, gerando movimento, magnetismo, calor, entre outros possíveis resultados. Ou seja, atuadores são componentes que convertem energia elétrica, hidráulica ou pneumática em energia mecânica, gerando uma força que resulta em um movimento. Muitas vezes os atuadores são usados junto dos sensores e placas Arduino para fechar e abrir trancas de portas ou em rodas para protótipos de carros e robôs pequenos.

Por fim, compreende-se que o Arduino é um microprocessador que possui um tipo de mecanismo capaz de controlar diversos aplicativos em anexo a sua placa, gerando informação e ações a partir destas. Todas essas ações realizadas são providas de um código desenvolvido em uma IDE (Integrated Development Environment), que é específica do Arduino.

O Arduino, utiliza uma linguagem de programação simples, adaptada da linguagem Processing, que é chamada de Wiring. O que torna essa linguagem simples é o fato de quando pressionado o botão de upload, o código escrito é traduzido para a linguagem C (Banzi, 2010). Essa IDE possibilita o desenvolvimento do código para qualquer placa Arduino que desejar e está disponível para os mais diversos sistemas operacionais, incluindo Windows 8.1 e 10, Mac OS X e Linux, abrangendo assim o mais variado público de desenvolvedores interessados.

## 2.3 DESENVOLVIMENTO DE SOFTWARE

A história da interação humano-computador tem se afastado da ideia fria e reducionista do mundo sobre o computador, apresentando camadas mais amigáveis em cima disso. As duas ideias mais importantes neste processo tem sido o uso de linguagens de computador, que mapeiam bem para o nosso cérebro por se parecer com as linguagens que usamos para conversar uns com os outros, e interfaces gráficas de apontar e clicar (ou toque), que nós facilmente entendemos por imitar o mundo tangível fora da máquina (HAVERBEK, 2014).

Programação, ao que parece, é difícil. As regras fundamentais são claras e simples. Mas programas, criados em cima destas regras básicas, tendem a tornar-se complexos o suficiente para introduzir suas próprias regras e complexidades. Você está construindo seu próprio labirinto, e pode simplesmente perder-se nele (HAVERBEK, 2014).

Uma divisão clássica para quem trabalha com desenvolvimento web é entre Front-end e Back-end. De maneira simples e clara, classificamos em Front-end tudo aquilo que você vê e o que você interage, ou seja, é a interface gráfica. Já o Back-end é o que está por trás dessa interface, que trabalha em conjunto com um servidor.

### 2.3.1 FRONTEND

Na web, o front-end trabalha no lado do cliente, no navegador. Os links, as imagens, os botões e os textos são o resultado de uma mistura de dois códigos escritos em duas linguagens: o Hyper Text Markup Language (HTML) e o Cascading Style Sheets (CSS). O HTML organiza todo o conteúdo que há dentro de uma página e ajuda o navegador a saber como exibir essas informações. O CSS é utilizado para especificar estilos visuais e ainda, é responsável por toda a parte estética (CHAN, 2016).

Junto do HTML e do CSS, uma linguagem de programação se faz necessário conhecer, o JavaScript. Essa linguagem, basicamente, da vida ao sistema a partir da interação com o

usuário. Para Pereira (2018), programar um sistema web apenas com a linguagem HTML seria como olhar para uma pessoa e ver apenas o esqueleto. É estranho. Sem vida. Sem atração. Por isso que costumamos dizer que a base para a programação Front-End é o tripé: HTML + CSS + JavaScript.

Ainda segundo Pereira (2018), elas são a base inicial de conhecimento de um desenvolvedor Front-End, que é o profissional que vai trabalhar com elas de forma tão uniforme, que vai garantir que o site criado seja atrativo e de fácil leitura em todos os lugares: web, *smartphone*, *tablets*, entre outros canais onde o site puder ser aberto pelos navegadores.

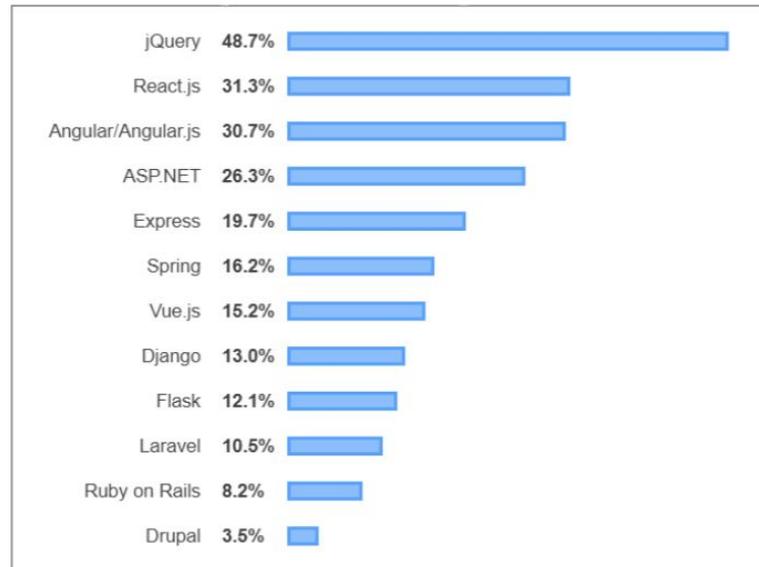
A partir desta afirmativa, compreende-se a importância desta união e funcionamento. Com esse objetivo, existem diversas bibliotecas e tecnologias que auxiliam no seu funcionamento, dentre elas está o React e o React Native. Ambos baseados em JavaScript e o segunda capaz de gerar código nativo para IOS e Android, se tornando um grande diferencial.

### 2.3.1.1 REACT

De acordo com Kostrzewa (2018) e Pandit (2018), o React foi criado em 2011 pela equipe de desenvolvedores do Facebook, como uma forma de resolver um problema que lhes causava considerável dor de cabeça: a necessidade frequente de atualização dos elementos que compõem a timeline da plataforma durante a navegação do usuário. Assim, desenvolveram uma biblioteca cujo propósito principal era renderizar, de maneira mais eficaz, as mudanças ocorridas na view, ou seja, a camada visual da aplicação. O restante do modelo MVC (Model-View-Controller), arquitetura utilizada em aplicações Web e muito consolidada, se tornou responsabilidade de bibliotecas e *plug-ins*, além de outras ferramentas externas.

Por ser uma ferramenta *open-source* e contar com uma comunidade de desenvolvedores individuais, além de sua mantenedora (Facebook), o React é uma ferramenta em expansão contínua, seja no uso como nas funcionalidades lançadas anualmente. A plataforma StackOverflow (2019) conduziu uma pesquisa sobre os *Web Frameworks* mais utilizados por

desenvolvedores e atingiu 63.585 desenvolvedores de todo o mundo. Como resultado, o React aparece em segundo lugar, contabilizando 31,3% de votos, de acordo com a Figura 1.



**Figura 1: Utilização de Web Frameworks por desenvolvedores em 2019**

**Fonte: StackOverflow (2019)**

De acordo com Silva e Sousa (2019), o React opera no paradigma de linguagem declarativa, o que possibilita maior legibilidade e facilidade em dar manutenção no código. Isso porque o React faz uso do compilador JSX, que é essencialmente JavaScript, porém voltado para uma linguagem de componentes, o que facilita não só a organização, mas também a visualização da montagem dos componentes na hora de desenvolver

Para Branks (2017), o React enfatiza a programação funcional sobre programação orientada a objetos. Essa mudança de pensamento, pode levar a benefícios em áreas como testabilidade e desempenho. Uma das principais características do React é essa sintaxe conhecida como JSX.

JSX funciona como uma extensão para o ECMAScript (a especificação da linguagem JavaScript), gerando códigos JavaScript através de *transpilation*, oferecendo uma sintaxe similar ao XML/HTML. Possui algumas poucas diferenças na estruturação de tags, como, por exemplo, a substituição das propriedades *for* e *class*, por serem palavras reservadas do JavaScript, as quais

precisam ter outros nomes nesta sintaxe. O seu princípio fundamental é juntar JavaScript e *layout* HTML no mesmo código (ANTONIO, 2015).

Segundo Occhino (2015), engenheiro diretor do grupo React do Facebook, o lema “Aprenda uma vez, escreva em qualquer lugar”, significa que o desejo de fazer com que a experiência que os desenvolvedores têm para desenvolver para Web com o ReactJS, seja a mesma para desenvolver mobile. Seguindo esse conceito, em 2015 o Facebook lançou o React Native, *framework* para desenvolvimento mobile.

### 2.3.1.2 REACT NATIVE

React Native (RN) é um projeto desenvolvido pelos engenheiros do Facebook que consiste em uma série de ferramentas que juntas possibilitam a criação de aplicativos móveis nativos para Android e iOS, utiliza as ferramentas de front-end mais modernas e o desenvolvimento é baseado em JavaScript (CABRAL, 2016).

Diferente de outras alternativas para aplicações mobile que utilizam webviews para renderizar a aplicação, o React Native invoca as APIs de renderização nativas em Objective C, para iOS, ou Java para Android. Portanto, a aplicação será renderizada utilizando componentes nativos da plataforma (EISENMAN, 2016).

As principais vantagens da utilização desse framework são: conter uma experiência do usuário muito mais fluida, uma vez que, ele gera códigos nativos; os carregamentos e requisições são mais rápidos, pois, não necessita conter uma webview para intermediar esse processo; possui uma melhor integração entre funções do dispositivo, como câmera, gps, giroscópio, etc.; possui maior segurança em relação a aplicativos web mobile e uma performance em geral relativamente superior (PEDRASSANI, 2018).

O desenvolvimento é extremamente mais rápido, possuindo um reaproveitamento de código gigantesco que não existiria se fosse desenvolvido em object-c para iOS e java para Android (KUPKA, 2017). Além de tudo, o React Native é contemplado com uma comunidade extremamente colaborativa, caso ocorra problemas com desenvolvimento, com plugins, bibliotecas ou incompatibilidades, torna-se extremamente mais fácil e rápida a forma de encontrar suas soluções.

Além disso, muitas vezes a comunidade disponibiliza componentes prontos para serem utilizados, e mesmo que não sejam encontrados, a criação de um componente nativo pode ser feita uma única vez e reutilizada posteriormente (OLIVEIRA, 2018). Assim, o React Native, hoje, pode ser considerado uma sensação dentre as tecnologias de desenvolvimento de software *mobile*.

Uma vez que este possibilita o desenvolvimento mais fácil, prático e com mais suporte, por parte comunidade, empresas do mundo inteiro optam por aprender e utilizar essa tecnologia, criando diversas aplicações seguras e usadas a nível internacional. Algumas aplicações criadas que podem ser citadas é o próprio Facebook, Airbnb, Baidu Mobile, entre outras.

Mesmo ainda não sendo uma plataforma totalmente estável e ‘madura’, o React Native é sem dúvidas uma das maiores promessas para o desenvolvimento *mobile* em um futuro próximo, pois o número de desenvolvedores web é cada vez maior e é muito mais prático para esse desenvolvedores utilizar uma linguagem que já estão familiarizadas do que aprender linguagens e paradigmas de programação novos para criar um aplicativo *mobile* (KUPKA, 2017).

### **2.3.2 BACKEND**

Uma aplicação servidora ou aplicação de backend em computação é o sistema responsável por armazenar e/ou processar e fornecer dados para que outras aplicações ou sistemas possam consumir de diversas formas. Uma aplicação servidora, geralmente é aquela em que os dados de todo o sistema são armazenados através de bancos de dados e trabalhados por outros processos computacionais, para que possam gerar novas informações posteriormente (BACKENDLESS, 2017).

É também comum encontrar toda a estrutura de regras de negócio de um sistema sob a responsabilidade de uma aplicação backend, uma vez que esta é a que disponibiliza de mais recursos computacionais quando a comparamos com aplicações cliente, como páginas web sendo

requisitadas de notebooks de uso pessoal, aplicativos móveis rodando em smartphones comuns e outros (PLURALSIGHT, 2015).

### **2.3.2.1 JAVASCRIPT**

O JavaScript nasceu por volta de 1995, seu principal objetivo era manipular as validações de entradas que anteriormente eram feitas com linguagens no lado do servidor como Java e PHP. O JavaScript fez a atual geração de aplicativos baseados na web possível em navegadores de e-mail, mapas e redes sociais, e também é usado em locais mais tradicionais por fornecer diversas formas de interatividade (HAVERBEK, 2014).

O JavaScript, comumente abreviado para JS é uma linguagem de programação, caracterizada por ser leve, interpretada e fundamentada em objetos. É mundialmente conhecida por ser uma linguagem de script para páginas Web e vem ganhando força no uso em ambientes sem browsers, como o Node.js para aplicações de serviços ou como o Apache CouchDB para banco de dados. O JS, além de ser uma linguagem de script que oferece suporte a diferentes paradigmas de programação (orientada à objetos, funcional, imperativa), baseia-se na dinâmica de protótipos, que permite o uso de herança entre objetos (MOZILLA, 2017).

O JS é uma das linguagens de programação mais conhecidas ao redor do globo, o que contribuiu para o massivo número de frameworks criados para a linguagem JavaScript. Um framework JS, descreve como uma aplicação deve ser construída, e faz com que o código seja reutilizável e mais organizado, o que implica em mais flexibilidade e escalabilidade desta aplicação (MARIANO, 2017).

Para Elliott (2014), JavaScript é a mais importante linguagem de programação da Terra. Quase todo mundo com um computador ou um smartphone tem todas as ferramentas que precisa para executar um programa JavaScript que ele mesmo criou. Um dos fatores que fizeram com que o JavaScript se torna-se muito conhecido e utilizado é que estudantes e iniciantes no ramo da tecnologia, com foco no desenvolvimento de software, buscavam

desenvolver sistemas a partir apenas uma linguagem de programação. Problema esse solucionado com o JavaScript, possibilitando o desenvolvimento do software com a mesma bases de linguagem de programação.

### 2.3.2.2 NODE.JS

Node.js é um runtime - aplicação que possibilita o processamento, a renderização e a execução de elementos escritos em linguagem não suportada nativamente pelo sistema - orientado a eventos assíncronos desenvolvido sobre o motor JavaScript V8 do Google Chrome, o Node foi projetado para criar aplicativos de rede escalonáveis<sup>4</sup>. Além disso, ele usa uma arquitetura voltada a eventos não bloqueáveis. As requisições são interpretadas dentro de um loop de eventos em uma única thread e, isso é feito de forma assíncrona não bloqueável, o que torna a aplicação mais rápida e eficiente, principalmente quando se tem um número extenso de requisições (SANTOS, 2016)

O Node se utiliza de um mecanismo com uma lógica de entrada/saída dirigida a eventos não bloqueante, para dar suporte a execução de múltiplas operações e tarefas em seu ambiente, o que a torna muito leve e eficiente. O universo de bibliotecas disponíveis para aplicações Node é o npm, que é o maior ecossistema de bibliotecas *open source* do globo (NODEJS, 2017). Outra vantagem da linguagem, é sua constante evolução e aprimoramento, uma vez que empresas como Google, Mozilla, Apple, entre outras, usam a linguagem em seus navegadores.

Uma das principais características do Node.js é o uso da linguagem JavaScript como recurso de desenvolvimento, que é uma das linguagens mais populares do mundo. Além de amplamente usada, a linguagem é pode ser executada quase em qualquer Sistema Operacional (CANTELON, 2017). Além disso, com o NodeJS é possível executar aplicações em qualquer plataforma, por exemplo: pode-se escrever todo o código no Mac OS, e em outros ambientes de produção executá-lo em ambiente Windows (FELIX, 2016).

---

<sup>4</sup> Disponível em: <<https://nodejs.org/en/about/>>. Acessado em: 21 outubro 2019.

### 2.3.3 SGBD

A utilização de banco de dados está presente no cotidiano da sociedade. Eles são utilizados em formas simples, como uma agenda que armazena nomes, endereços e telefones de diversas pessoas; e, também, em sistemas mais complexos, como nos sistemas de saque de dinheiro em caixas eletrônicos. Neste último caso, há um sistema que armazena as informações da conta e atualiza o saldo (MARCHESAN, 2012).

Um sistema de gerenciamento de banco de dados (SGBD) pode ser definido como

uma coleção de programas que permite aos usuários criar e manter um banco de dados. O SGBD é, portanto, um sistema de software de propósito geral que facilita os processos de definição, construção, manipulação e compartilhamento de bancos de dados entre vários usuários e aplicações. A definição de um banco de dados implica especificar os tipos de dados, as estruturas e as restrições para os dados a serem armazenados em um banco de dados (ELMASRI, 2005).

Pela sua funcionalidade, existem diversos tipos de banco de dados com diferentes características, ferramentas privadas e *open source*. Dentre essa variedade ofertada, escolheu-se o banco de dados relacional MySQL, por ser *open source* e muito conhecido, facilitando a resposta à dúvidas a partir de suas vastas comunidades que compartilham de experiências com a ferramenta e diferentes formas de usá-la e otimizá-la.

#### 2.3.3.1 MYSQL

Apesar de ser um SGBD antigo, criado em 1994, Alecrim (2008) afirma que “o MySQL é um dos sistemas de gerenciamento de banco de dados mais populares que existe e, por ser otimizado para aplicações Web, é amplamente utilizado na Internet”. Essa aplicação é suportada por diferentes sistemas operacionais, entre eles estão o Windows e o Linux.

Um banco de dados nada mais é do que uma hierarquia de estruturas de dados complexas. Em MySQL, como em muitos outros bancos de dados, o conceito da

estrutura que mantém os blocos (ou registros) de informações é chamado de tabela. Estes registros, por sua vez, são constituídos de objetos menores que podem ser manipulados pelos usuários, conhecidos por tipos de dados (*datatypes*). Juntos, um ou mais *datatypes*, formam um registro (*record*). Uma hierarquia de banco de dados pode ser considerada como: Banco de dados > Tabela > Registro > Tipo de dados. Os tipos de dados possuem diversas formas e tamanhos, permitindo ao programador criar tabelas específicas de acordo com suas necessidades (GONZAGA & BIRCKAN, 2000).

Segundo Barros et al. (2017), a arquitetura do MySQL é composta por três camadas: 1) conexão com o cliente; 2) análise da instrução SQL e; 3) execução da instrução. Sendo que, a primeira camada é onde ocorre a conexão das aplicações com o banco de dados; a segunda é responsável, quando necessário, por otimizar a instrução, onde são construídas as *procedures*, *triggers* e *views*; a terceira camada é responsável por executar a instrução recebida da camada acima, sendo tais instruções executadas no *Storage Engine*, responsável por armazenar e recuperar todos os dados

A partir das inúmeras variedades de tabelas disponibilizadas para uso, é possível criar uma estrutura que armazena em log todas as alterações e atualizações de dados, sendo possível, a partir do sistema, realizar a recuperação de determinada informação, de acordo com a necessidade dos usuários.

## 2.4 TRABALHOS RELACIONADOS

Nesta seção, serão abordados alguns trabalhos relacionados ao tema proposto, a partir de um breve resumo justificando como estes auxiliaram no desenvolvimento deste projeto.

O primeiro trabalho a ser citado é de Bukman (2016), que propôs o desenvolvimento de um protótipo de baixo custo para a segurança residencial. Este criou dois modelos, sendo o primeiro utilizando apenas o Arduino UNO, um sensor de movimento PIR e uma sirene, adicionando uma protoboard para facilitar suas conexões. O segundo modelo foi criado a partir do mesmo Arduino UNO e componentes mencionados anteriormente, porém este não possui a protoboard, sendo então utilizado um *shield Ethernet* junto de um roteador, sendo possível enviar

as informações do Arduino à Internet. Ainda, neste segundo modelo foi integrado uma conta do Twitter para que seja enviada as informações do Arduino à outro usuário da rede social.

Assim que o Arduino detecta uma presença pelo seu sensor PIR, o mesmo envia essa informação via Twitter ao dono da residência ou outro usuário cadastrado no código do Arduino, sendo então possível ser alertado de uma possível invasão. Uma vez que a cultura da Automação Residencial é pouco difundida no Brasil, utilizar redes sociais como forma de integrar essa tecnologia com a realidade da população foi uma forma criativa de realizar esta integração, além de usar ferramentas *open source* e registrar o baixo custo investido em seus protótipos.

Outro projeto citado é o de Botke (2014), que também com Arduino propôs a automação de residências controlada por uma aplicação Web. Sua aplicação foi criada com o Arduino Mega, considerado uma evolução do Arduino UNO em diversos aspectos, *shield Ethernet*, um sensor de contato para monitoramento da janela, e três atuadores, sendo um transmissor de infravermelho para controlar condicionador de ar e dois relés para o controle de luzes. O mesmo também se propôs a emular seu sistema em dispositivos móveis com os sistemas operacionais Android e IOS e obteve sucesso em seus testes, disponibilizando mais conforto aos seus usuários que poderão realizar o monitoramento de janelas e controlar os condicionadores de ar e as luzes de sua casa, a partir de seu *smartphone*.

Por fim, o projeto de Marchesan (2012), executado em Santa Maria, no Rio Grande do Sul, cidade vizinha a que reside o autor. Marchesan propôs um sistema de monitoramento residencial utilizando também Arduino, a partir de um sensor magnético acoplado em portas e janelas de uma residência. Toda vez que um sensor identifica que seu laço elétrico com o ímã foi interrompido, irá realizar um *tweet* informando qual sensor foi ativado, a partir de um login e uma senha, referente ao acesso da rede social Twitter, previamente cadastrado no código do Arduino.

Após este, enviará uma mensagem ao sistema Web desenvolvido para realizar o monitoramento dos sensores. Este projeto é muito semelhante a segunda versão do trabalho de Bukman (2016), citado anteriormente. O que difere este trabalho do anterior é o modelo do

sensor magnético utilizado, além de não ser utilizado um buzzer e conter um sistema Web para realizar o monitoramento.

Em todos os trabalhos, o que mais toca o autor deste é a busca por conforto e segurança a partir do Arduino. São diversos trabalhos que surgem e, especialmente em âmbito estudantil, cada vez mais frequentes e com novas formas de integrar os diferentes tipos de Arduino com seus diversificados componentes, abrindo uma imensidão de possibilidades ainda a serem estudadas e validadas. Por se tratar de ferramentas *open sources*, é comum que estudantes proponham de integrar Arduino com aplicações, principalmente *mobile*, que utilizam tecnologia mais atual.

A partir disso, a principal diferença da proposta do autor em relação aos trabalhos mencionados anteriormente é a utilização de um aplicativo *mobile* para controle e monitoramento do arduino, possibilitando a ativação e inativação do alarme, assim como receber notificações quando disparado o mesmo. Além disso, é incorporado um teclado ao protótipo, possibilitando ativação/inativação do alarme de uma forma manual, para quando o usuário estiver sem seu *smartphone* ou este esteja com algum defeito ou outro problema.

### 3 METODOLOGIA

O presente trabalho é caracterizado pelo método de abordagem a partir da pesquisa quantitativa e qualitativa. Segundo Lüdke e André (1986), a pesquisa qualitativa é também denominada naturalística pois “[...] pressupõe o contato direto e prolongado do pesquisador com o ambiente e a situação que está sendo investigada, através do trabalho intensivo de campo” (p. 11-12).

Para Minayo (2007), a pesquisa qualitativa trabalha com o universo de significados, motivos, aspirações, crenças, valores e atitudes, o que corresponde a um espaço mais profundo das relações, dos processos e dos fenômenos que não podem ser reduzidos à operacionalização de variáveis. Neste projeto foram realizadas pesquisas buscando projetos relacionados ao tema e problema abordado, assim como os testes do mesmo foram executados com um número reduzido de pessoas, caracterizando-se como uma pesquisa qualitativa.

Para este foram realizados testes guiados por cenários e testes exploratórios, com o objetivo de validar a funcionalidade e aplicabilidade do protótipo em diversos cenários reais possíveis, registrando como o sistema se portou diante destes testes. Os cenários de testes são focados no comportamento do protótipo, visto que as funcionalidades do sistema seguem o padrão de escrita no formato de histórias de usuário, baseado no padrão de desenvolvimento BDD (Behaviour Driven Development) ou também conhecido como Especificação por Exemplos.

Quanto aos objetivos, foi utilizado o método de pesquisa descritiva, a qual exige do investigador uma série de informações sobre o que deseja pesquisar. Esse tipo de estudo, pretende descrever os fatos e fenômenos de determinada realidade (TRIVIÑOS, 1987). Este, se caracteriza pelo desenvolvimento de um estudo de caso na localidade em que reside o autor, bem como propõe-se a resolução de um problema enfrentado pelo mesmo.

Para completar sua validação, foi utilizado o procedimento denominado de Estudo de Caso. Fonseca (2002, p. 33) explica:

Um estudo de caso pode ser caracterizado como um estudo de uma entidade bem definida como um programa, uma instituição, um sistema educativo, uma pessoa, ou uma unidade social. Visa conhecer em profundidade o como e o porquê de uma

determinada situação que se supõe ser única em muitos aspectos, procurando descobrir o que há nela de mais essencial e característico. O pesquisador não pretende intervir sobre o objeto a ser estudado, mas revelá-lo tal como ele o percebe. O estudo de caso pode decorrer de acordo com uma perspectiva interpretativa, que procura compreender como é o mundo do ponto de vista dos participantes, ou uma perspectiva pragmática, que visa simplesmente apresentar uma perspectiva global, tanto quanto possível completa e coerente, do objeto de estudo do ponto de vista do investigador.

O projeto proposto se caracteriza como estudo de caso uma vez que a motivação em realizá-lo se deu a partir da necessidade identificada na localidade de Recanto Maestro, que possui um campus universitário frequentado por milhares de pessoas vindas de diversas cidades vizinhas, que assim nos revela um ambiente de insegurança pelo alto fluxo de pessoas desconhecidas, frequentando o local, além de ser o local onde reside o autor e se identifica com esta necessidade. Além disso, a criação de um protótipo funcional e validado, como resultado deste estudo, reforça a caracterização deste como um estudo de caso.

### **3.1 DESIGN DO ESTUDO**

Conforme explanado anteriormente, o problema de pesquisa foi identificado a partir de uma necessidade do autor, assim como de demais moradores que residem na localidade de Recanto Maestro, campus universitário com fluxo diário muito grande de visitantes e alunos. Após a definição da necessidade, iniciou-se o estudo por uma solução *open source* e com baixo custo a ser desenvolvida neste projeto.

O presente trabalho foi desenvolvido, em cinco etapas. Inicialmente foi realizado o desenvolvimento do sistema web do Administrador do produto, com React e Node.js, junto também da configuração do banco de dados de todo o projeto, utilizando o MySQL, incluindo neste as tabelas e relações necessárias para o aplicativo *mobile*. Na segunda parte foi desenvolvido o aplicativo que é disponibilizado ao usuário, com React Native e Node.js, utilizando as configurações de banco de dados, definidas na primeira etapa.

Desta forma, no desenvolvimento dos softwares foram utilizadas as tecnologias React, React Native, Node.js e o banco de dados MySQL. Estas foram escolhidas por serem tecnologias *open source*, possibilitando uma redução nos custos do projeto, e de conhecimento do autor. Este stack de tecnologias é considerado extremamente novo, porém já é muito utilizado em empresas e muito promissor.

Por possuírem diversas comunidades, que compartilham seus trabalhos com essas tecnologias, esse stack está sendo utilizado, por professores, como tecnologias para ensino. Este, o caso do autor, que obteve seu primeiro contato com a tecnologia durante a graduação. Neste mesmo contexto é possível justificar a escolha do MySQL, onde, a partir do momento que foi identificada a necessidade de utilizar um banco de dados relacional, autor optou por um sistema de banco de dados um pouco mais antigo, em relação ao seu stack, porém muito seguro, *open source*, e também de seu conhecimento.

O protótipo do Arduino foi construído utilizando a placa Wemos D1 R2, muito parecida com a placa Uno, porém a Wemos possui um módulo Wi-Fi ESP8266 já embutido em sua placa, reduzindo a necessidade de associação deste módulo ao Arduino. Nesta placa também foi associado um Sensor Magnético com fio, utilizado para acionamento ou não do alarme, além de LEDs de brilho para informar o status do Arduino.

O terceiro momento foi dedicado ao desenvolvimento e testes de hardware e software por parte do desenvolvedor no arduino, de forma isolada, ainda sem integração com as aplicações desenvolvidas nas etapas anteriores.

O quarto momento foi designado a integração de todas as aplicações, criando então um servidor local para manter ativo os serviços de banco de dados e do Node.js, assim como a integração do Arduino com o aplicativo, possibilitando, assim, o controle do protótipo pelo aplicativo *mobile*.

A partir da aplicação funcional e integrada entre seus módulos, deu-se início a quinta etapa do projeto, cujo objetivo foi realizar os testes da aplicação e correções quando necessário, bem como a estilização do projeto, adotando uma identidade visual principalmente para o aplicativo e também utilizada na aplicação Web. Os testes realizados seguiram um roteiro com os

fluxos principais de funcionamento, além de cenários com exceções, validando o funcionamento do sistema e o tratamento para erros, quando ocorressem. Os cenários de testes foram desenvolvidos com foco no comportamento do protótipo, onde a descrição destes seguiu o modelo abaixo:

<b>Cenário 00 - Título - Válido</b>	
<b>Dado</b>	Pré condições para a realização do teste
<b>Quando</b>	Realizo alguma ação no sistema em teste
<b>Então</b>	Alguma coisa deve acontecer (o que é esperado que aconteça e será verificado no teste?)

Durante todas as etapas de desenvolvimento, também foram realizadas pesquisas bibliográficas sobre os materiais e ferramentas utilizadas no projeto, com o objetivo de utilizar, no software, tecnologias atuais e com o arduino, reduzir os custos do projeto. As pesquisas também possibilitaram ao autor realizar um breve nivelamento sobre seu projeto quando comparado às publicações recentes de outros projetos.

### **3.2 INSTRUMENTOS DE COLETA DE DADOS**

Para a realização de coleta de dados foi utilizado o sensor magnético com fio, ligado ao protótipo e então acoplado a porta da residência do autor. Este sensor capta os momentos em que a porta é aberta - interrompendo a ligação elétrica dos sensores - registrando este acontecimento na base de dados da aplicação a partir de uma requisição POST. Esses dados podem ser visualizados a partir de consultas na base de dados ou então pelo usuário na tela de histórico, disponível no aplicativo *mobile*. Além destes dados, serão realizados testes guiados por cenários, conforme descrito abaixo.

<b>CENÁRIOS DE TESTES - APLICATIVO <i>MOBILE</i></b>	
<b>Cenário 01</b>	<b>Usuário Realizar Login - Válido</b>
<p><b>Dado</b> que estou no acessando o Aplicativo  E estou na tela de login  <b>Quando</b> inserir o Login e Senha de um Usuário cadastrado  E pressiono o botão “Entrar”  <b>Então</b> o sistema deverá efetuar o login, direcionando-me para a tela Home, exibindo o status do alarme, de acordo o protótipo da tela Home</p>	
<b>Cenário 02</b>	<b>Administrador Realizar Login - Inválido</b>
<p><b>Dado</b> que estou no acessando o Aplicativo  E estou na tela de login  <b>Quando</b> inserir o Login e Senha do Administrador  E pressiono o botão “Entrar”  <b>Então</b> o sistema deverá negar o acesso e emitir uma mensagem, cujo texto é “Usuário ou senha inválido”.</p>	
<b>Cenário 03</b>	<b>Ativar Alarme - Válido</b>
<p><b>Dado</b> que estou ativando o alarme  E estou logado no aplicativo com um usuário  <b>Quando</b> pressiono o botão para ativar o alarme  <b>Então</b> o sistema deverá emitir uma mensagem, cujo texto é “O alarme será ativado em 10 segundos!”.</p>	
<b>Cenário 04</b>	<b>Desativar Alarme - Válido</b>
<p><b>Dado</b> que estou desativando o alarme  E estou logado no aplicativo com um usuário  <b>Quando</b> pressiono o botão para desativar o alarme  <b>Então</b> o sistema deverá emitir uma mensagem, cujo texto é “O alarme foi desativado!”.</p>	
<b>Cenário 05</b>	<b>Acessar Contatos de Emergência - Válido</b>
<p><b>Dado</b> que estou logado no aplicativo com um usuário  <b>Quando</b> acessar o menu lateral  E selecionar a opção Contatos  <b>Então</b> o sistema deverá direcionar o usuário para a tela de listagem de Contatos, exibindo os contatos existentes possibilitando inclusão de novos contatos, assim como edição ou exclusão desses registros.</p>	
<b>Cenário 06</b>	<b>Incluir Contato de Emergência - Válido</b>

<p><b>Dado</b> que estou logado no aplicativo com um usuário  <b>E</b> estou na tela de listagem de Contatos  <b>E</b> selecionei a opção Novo Contato  <b>Quando</b> preencher as informações obrigatórias, sendo essas “Nome” e “Número”  <b>E</b> pressionar o botão “Incluir”  <b>Então</b> o sistema deverá exibir uma mensagem, cujo texto é “Contato incluído com sucesso!” e retornar para a tela de listagem de contatos.</p>	
<b>Cenário 07</b>	<b>Editar Contato de Emergência - Válido</b>
<p><b>Dado</b> que estou logado no aplicativo com um usuário  <b>E</b> estou na tela de listagem de Contatos  <b>E</b> selecionei a opção Editar Contato  <b>Quando</b> realizar as alterações necessárias, mantendo as informações obrigatórias preenchidas, sendo essas “Nome” e “Número”  <b>E</b> pressionar o botão “Salvar”  <b>Então</b> o sistema deverá exibir uma mensagem, cujo texto é “Alterações salvas com sucesso!” e retornar para a tela de listagem de contatos.</p>	
<b>Cenário 08</b>	<b>Acessar Histórico - Válido</b>
<p><b>Dado</b> que estou logado no aplicativo com um usuário  <b>Quando</b> acessar o menu lateral  <b>E</b> selecionar a opção Histórico  <b>Então</b> o sistema deverá direcionar o usuário para a tela de Histórico, exibindo as últimas alterações do sistema, assim como a data e o responsável pela alteração.</p>	
<b>Cenário 09</b>	<b>Acessar Configurações - Válido</b>
<p><b>Dado</b> que estou logado no aplicativo com um usuário  <b>Quando</b> acessar o menu lateral  <b>E</b> selecionar a opção Configurações  <b>Então</b> o sistema deverá direcionar o usuário para a tela de Configurações, exibindo as informações referente ao Nome, E-mail e Número.</p>	
<b>Cenário 10</b>	<b>Editar Configurações - Válido</b>
<p><b>Dado</b> que estou logado no aplicativo com um usuário  <b>E</b> estou na tela de Configurações  <b>Quando</b> realizar as alterações necessárias, mantendo as informações obrigatórias preenchidas, sendo essas “Nome” “E-mail” e “Número”  <b>E</b> pressionar o botão “Salvar”  <b>Então</b> o sistema deverá exibir uma mensagem, cujo texto é “Alterações salvas com sucesso!” e se manterá na tela de Configurações.</p>	

<b>Cenário 11</b>	<b>Apagar Configurações - Inválido</b>
<p><b>Dado</b> que estou logado no aplicativo com um usuário  <b>E</b> estou na tela de Configurações  <b>Quando</b> apagar as informações de pelo menos um campos “Nome” “E-mail” ou “Número”  <b>E</b> pressiono o botão “Salvar”  <b>Então</b> o sistema deverá exibir uma mensagem, cujo texto é “Preencha as informações necessárias!”, além de destacar em vermelho os campos que estiverem com informações inválidas ou sem preenchimento.</p>	

**Tabela 1: Cenários de Testes - Aplicativo *Mobile***

Os cenários descritos na tabela 1, têm como objetivo garantir a funcionalidade e integridade do aplicativo *mobile*, a partir de testes objetivos e focados na funcionalidade do protótipo. Junto destes, também são executados os testes exploratórios, que por sua vez tem como objetivo encontrar problemas na execução de determinado fluxo para uma operação, bem como defeitos visuais. Na tabela 2, serão descritos os cenários de testes direcionados ao Sistema Web.

<b>CENÁRIOS DE TESTES - SISTEMA WEB</b>	
<b>Cenário 01</b>	<b>Usuário Realizar Login no Sistema Web - Inválido</b>
<p><b>Dado</b> que estou no acessando o Sistema Web  <b>E</b> estou na tela de login  <b>Quando</b> inserir o Login e Senha de um Usuário  <b>E</b> pressiono o botão “Entrar”  <b>Então</b> o sistema deverá negar o acesso e emitir uma mensagem, cujo texto é “Usuário ou senha inválido”.</p>	
<b>Cenário 02</b>	<b>Administrador Realizar Login no Sistema Web - Válido</b>
<p><b>Dado</b> que estou no acessando o Sistema Web  <b>E</b> estou na tela de login  <b>Quando</b> inserir o Login e Senha do Administrador  <b>E</b> pressiono o botão “Entrar”  <b>Então</b> o sistema deverá efetuar o login, direcionando-me para a tela principal do Sistema.</p>	
<b>Cenário 03</b>	<b>Cadastrar Pessoa - Válido</b>
<p><b>Dado</b> que estou logado no sistema como administrador  <b>E</b> estou na tela de Cadastro de Pessoa  <b>Quando</b> inserir os dados obrigatórios</p>	

<p>E pressionar o botão “Cadastrar”  <b>Então</b> o sistema deverá efetuar o cadastro, direcionando-me para a listagem de Pessoas.</p>	
<b>Cenário 04</b>	<b>Cadastrar Produto - Válido</b>
<p><b>Dado</b> que estou logado no sistema como administrador  E estou na tela de Cadastro de Produto  <b>Quando</b> inserir os dados obrigatórios  E pressionar o botão “Cadastrar”  <b>Então</b> o sistema deverá efetuar o cadastro, direcionando-me para a listagem de Produtos.</p>	
<b>Cenário 05</b>	<b>Cadastrar Contato - Válido</b>
<p><b>Dado</b> que estou logado no sistema como administrador  E estou na tela de Cadastro de Contato  <b>Quando</b> inserir os dados obrigatórios  E pressionar o botão “Cadastrar”  <b>Então</b> o sistema deverá efetuar o cadastro, direcionando-me para a listagem de Contatos.</p>	
<b>Cenário 06</b>	<b>Cadastrar Central de Comando - Válido</b>
<p><b>Dado</b> que estou logado no sistema como administrador  E estou na tela de Cadastro de Central de Comando  <b>Quando</b> inserir os dados obrigatórios  E pressionar o botão “Cadastrar”  <b>Então</b> o sistema deverá efetuar o cadastro, direcionando-me para a listagem de Centrais de Comando.</p>	
<b>Cenário 07</b>	<b>Cadastrar Pessoa - Inválido</b>
<p><b>Dado</b> que estou logado no sistema como administrador  E estou na tela de Cadastro de Pessoa  <b>Quando</b> não inserir nenhum dado no formulário  E pressionar o botão “Cadastrar”  <b>Então</b> o sistema deverá destacar os campos obrigatórios que não foram preenchidos e manterá na mesma tela.</p>	
<b>Cenário 08</b>	<b>Cadastrar Produto - Inválido</b>
<p><b>Dado</b> que estou logado no sistema como administrador  E estou na tela de Cadastro de Produto  <b>Quando</b> não inserir nenhum dado no formulário  E pressionar o botão “Cadastrar”  <b>Então</b> o sistema deverá destacar os campos obrigatórios que não foram preenchidos e manterá na mesma tela.</p>	

<b>Cenário 09</b>	<b>Cadastrar Contato - Inválido</b>
<p><b>Dado</b> que estou logado no sistema como administrador  <b>E</b> estou na tela de Cadastro de Contato  <b>Quando</b> não inserir nenhum dado no formulário  <b>E</b> pressionar o botão “Cadastrar”  <b>Então</b> o sistema deverá destacar os campos obrigatórios que não foram preenchidos e manterá na mesma tela.</p>	
<b>Cenário 10</b>	<b>Cadastrar Central de Comando- Inválido</b>
<p><b>Dado</b> que estou logado no sistema como administrador  <b>E</b> estou na tela de Cadastro de Central de Comando  <b>Quando</b> não inserir nenhum dado no formulário  <b>E</b> pressionar o botão “Cadastrar”  <b>Então</b> o sistema deverá destacar os campos obrigatórios que não foram preenchidos e manterá na mesma tela.</p>	

**Tabela 2: Cenários de Testes - Sistema Web**

Os cenários descritos na tabela 2, têm como objetivo garantir a funcionalidade do Sistema Web a partir de testes objetivos e focados na funcionalidade do sistema. Junto destes, também são executados os testes exploratórios, que por sua vez tem como objetivo encontrar problemas na execução de determinado fluxo para uma operação, bem como defeitos visuais.

### 3.3 ANÁLISE DOS DADOS

Para a análise dos dados foi utilizado o método qualitativo de análise, onde foram verificados os resultados dos testes executados a fim de evidenciar a funcionalidade do sistema e eficiência na implantação do protótipo. Também foi realizada a comparação dos resultados obtidos neste projeto com os projetos mencionados anteriormente na seção “2.4 Trabalhos Relacionados”.

## **4 DESENVOLVIMENTO**

A proposta do protótipo de monitoramento de um alarme de porta consiste na utilização de equipamentos e ferramentas *open source* e de baixo custo, que através da integração dos componentes tem como objetivo notificar o usuário, dono da residência que possui um protótipo configurado e instalado, sobre possíveis invasões ou tentativas de acesso a sua residência sem a devida autorização, gerando assim mais conforto e segurança ao morador em relação a sua residência. A seguir são demonstrados todos os equipamentos e métodos utilizados na construção do protótipo.

### **4.1 COMPONENTES E ESQUEMÁTICO**

Todos os componentes utilizados no protótipo possuem funcionalidades específicas, além de algumas especificações para seu uso. Estes serão detalhados de forma individual, além de citar como foram utilizados e contribuem para a realização do protótipo.

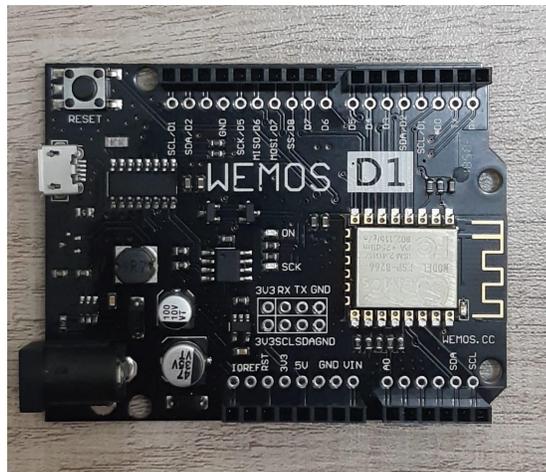
#### **4.1.1 WEMOS D1 R2 WiFi ESP8266**

Placa compatível com o Arduino Uno, mas que é controlada pelo módulo ESP8266EX, oferecendo conectividade WiFi nativa e é uma placa inspirada no Arduino Uno, uma das placas mais populares quando se fala em Arduino. Essa placa possui posicionamento dos conectores iguais aos da placa Uno e pode ser programada através da Arduino IDE, porém possui como principal diferencial em relação ao Uno seu microcontrolador principal, enquanto no Uno é o Atmega328, na Placa de Desenvolvimento ESP8266 Wemos D1 R2 é o ESP8266, um módulo que tem funções tanto de microcontrolador como de módulo wifi.

A placa Wemos D1 R2 possui algumas especificações diferentes em relação ao Arduino UNO R3, dentre estas, as principais diferenças são:

- Possui WiFi nativo;
- Possui 11 pinos digitais;
- Possui 1 pino analógico;
- 4 Mb de memória flash;
- Pinos operam em nível lógico de 3.3 Volts;
- Faixa de alimentação de 9 a 24 Volts;
- O processamento pode ser realizado em 80 MHz /160 MHz;
- Os pinos digitais e analógico não operam em 5 Volts;
- Possui a possibilidade de realizar a programação da placa via OTA (*Over The Air*), ou seja, através do WiFi você pode ser enviado os códigos para a placa;
- Possui apenas uma entrada micro USB, para envio de código via cabo.

Na Figura 2, é possível visualizar a Placa utilizada para o desenvolvimento do protótipo.



**Figura 2: Placa Wemos D1 R2 WiFi ESP8266**

**Fonte: elaborada pelo autor**

A programação do protótipo foi realizada através da Arduino IDE, utilizando algumas configurações personalizadas, específicas da placa Wemos D1 R2, conforme figura 3.

```
Placa: "LOLIN(WEMOS) D1 R2 & mini"  
Upload Speed: "115200"  
CPU Frequency: "80 MHz"  
Flash Size: "4MB (FS:3MB OTA:~512KB)"  
Debug port: "Disabled"  
Debug Level: "Nenhum"  
lwIP Variant: "v2 Lower Memory"  
VTables: "Flash"  
Exceptions: "Legacy (new can return nullptr)"  
Erase Flash: "Only Sketch"  
SSL Support: "All SSL ciphers (most compatible)"
```

**Figura 3: Configurações Arduino IDE**

**Fonte: elaborada pelo autor**

Além desta placa possuir a funcionalidade de realizar conexão WiFi, é possível realizar a integração com outros componentes a partir de suas portas digitais e analógica, como outros tipos e versões de Arduinos. Neste protótipo, foram realizadas três conexões, sendo uma delas com o sensor magnético ligando um de seus fios ao GND, que foi energizado na *protoboard*, e o outro foi ligado a porta digital D7. As outras duas conexões foram realizadas com dois LEDs, onde também foram ligados ao GND, energizado na *protoboard*, e utilizaram as portas digitais D5 e D6.

#### **4.1.2 SENSOR MAGNÉTICO COM FIO**

O modelo que foi utilizado do sensor magnético com fio, tem como propósito ser utilizado para a confecção de alarmes em portas e janelas, tornando-se eficaz e menos propício a falhas. Em um alarme mais simples, ao utilizar um sensor de movimento ou de distância, há a possibilidade de ativação do mesmo a partir da percepção de um animal doméstico, não se caracterizando como invasão. Já o sensor magnético irá detectar uma possível invasão apenas se o laço elétrico se romper, ou seja, quando o sensor estiver posicionado com uma peça na porta ou

janela e a outra em uma parede ou suporte ao lado da primeira, o alarme irá disparar somente quando a porta ou janela for aberta (movimentada). A figura 4 representa o sensor magnético utilizado neste protótipo.



**Figura 4: Sensor Magnético com fio**

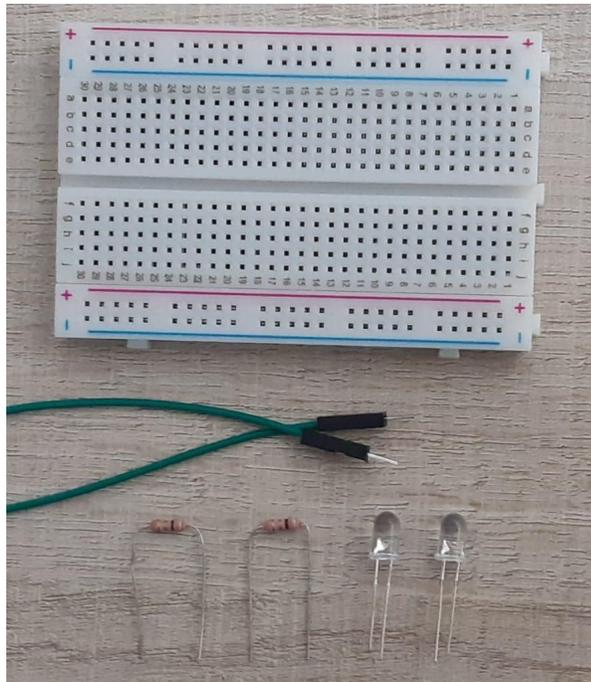
**Fonte: elaborada pelo autor**

Esse sensor, foi conectado com a placa Wemos D1 R2 a partir da *protoboard*, utilizando a saída do GND para um de seus fios e a porta digital D7 para o outro. Na placa, foi necessário apenas a configuração de uma regra para realizar a leitura do estado do sensor, onde este disponibiliza apenas dois tipos de informação, sendo eles *high* (possui conexão elétrica) e *low* (não possui conexão elétrica), facilitando a leitura e interpretação do estado do sensor.

### **4.1.3 COMPONENTES COMPLEMENTARES**

Além do sensor magnético, citado anteriormente, outros componentes foram utilizados para a confecção do protótipo. Foram utilizados dois LEDs coloridos que sinalizarão quando o alarme estiver ativado (LED de cor verde) e quando estiver desativado e pronto para uso (LED de cor azul). Para o funcionamento dos LEDs, foram utilizados também dois resistores para

controlar a corrente elétrica recebida, a fim de que não queime os LEDs com alta voltagem. Esses componentes foram ligados a uma *protoboard* a partir de sete *jumpers* do tipo macho-macho (que possui pinos para encaixe em ambos os lados dos fios), para auxílio nas ligações entre os componentes. A *protoboard* serviu, basicamente, para melhor organização dos componentes, não sendo necessária a realização de soldagem dos fios com os componentes. A Figura 5 é composta pelos componentes citados acima, utilizados no protótipo real para testes.



**Figura 5: LEDs, Resistores, *jumper* e *protoboard***

**Fonte: elaborada pelo autor**

#### 4.1.4 ESQUEMÁTICO E PROTÓTIPO

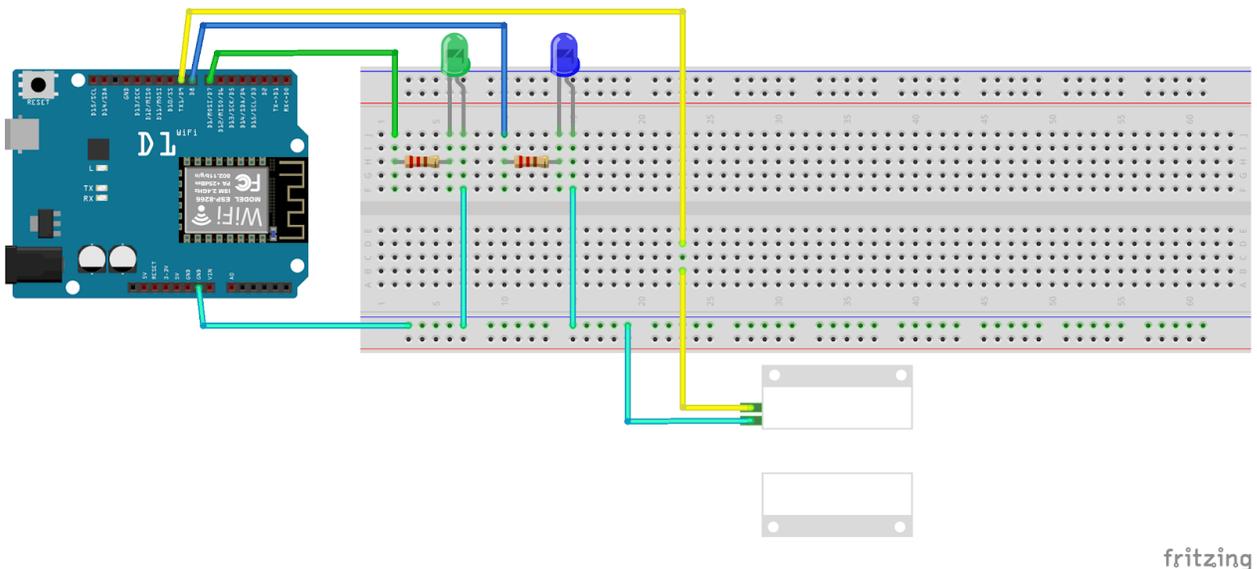
Para a construção do esquemático do protótipo foi utilizado o Fritzing<sup>5</sup>, ferramenta *open source* que auxilia no desenvolvimento de projetos de *hardware* eletrônico oferecendo uma lista

---

<sup>5</sup> Fritzing. Disponível em: <http://fritzing.org/home/>

de componentes para que seja criado o *design* do protótipo, para depois começar a ser construído na prática.

Na figura 6, é possível identificar o uso do sensor magnético com fio, na parte inferior da imagem, conectada a uma *protoboard*, equipamento para auxílio à montagem e desenvolvimento de circuitos eletrônicos fazendo com que não seja necessária a soldagem dos fios de componentes a placa. A partir da *protoboard*, os fios do sensor se conectam com a Placa Wemos D1 R2, uma placa muito semelhante a Placa UNO R3 (uma das placas mais populares do Arduino), porém está já possui um sensor WiFi ESP8266, que realiza a conexão com o servidor, que está conectado na mesma rede WiFi. É possível visualizar também dois LEDs que sinalizarão quando o alarme estiver ativado (LED de cor verde) e quando estiver desativado (LED de cor azul), esses estão ligados também à *protoboard*, e, com o auxílio de resistores, que servem basicamente para limitar a corrente elétrica em um circuito, se conectam também a placa Wemos D1 R2.



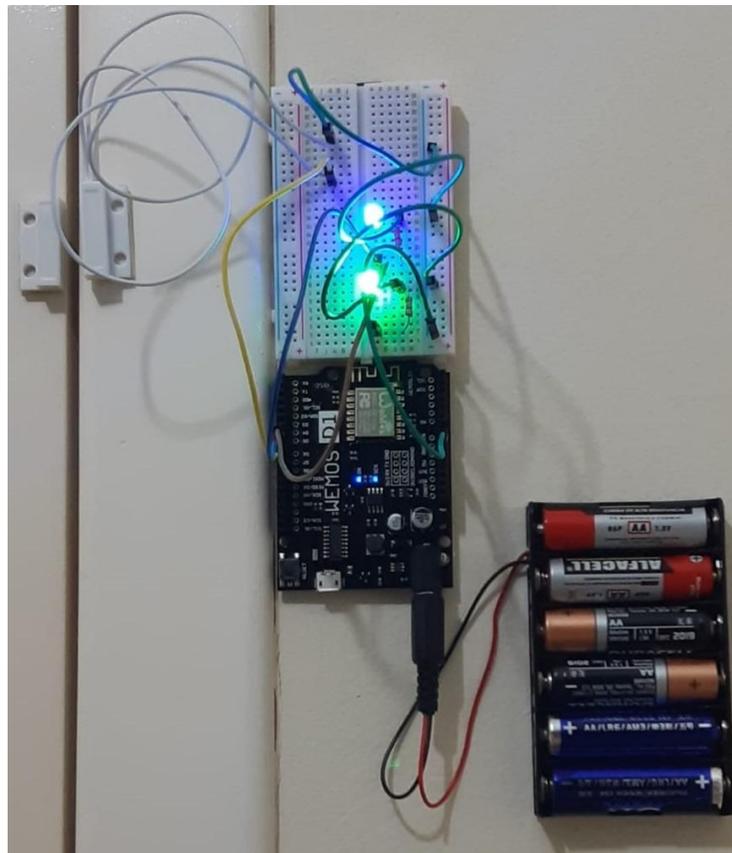
**Figura 6: Design do protótipo realizado no Fritzing**

**Fonte: elaborada pelo autor**

A partir da criação deste esquema, foi realizada a montagem do protótipo real com todos os componentes mencionados anteriormente, seguindo as mesmas ligações conforme ilustrado na

figura 6. A partir deste, foi possível iniciar o desenvolvimento do código para o funcionamento do Arduino e então submetê-lo a testes. A figura 7, a seguir, demonstra o protótipo real utilizado para os testes.

Na figura 6, é possível visualizar o protótipo que foi desenvolvido para utilização dos testes. Esse protótipo segue a estrutura demonstrada anteriormente, na figura 6, além de sua alimentação ser proveniente de pilhas, conforme possível visualizar na parte inferior direita da figura 7. Na parte superior esquerda da figura 7, é possível identificar o sensor magnético, ligado a *proto-board* (branca), posicionada no centro da figura. Além destes, também é possível visualizar os LEDs acesos, bem como os *jumpers* que auxiliam na comunicação com a Placa Wemos D1 R2, posicionada no centro da figura 7.



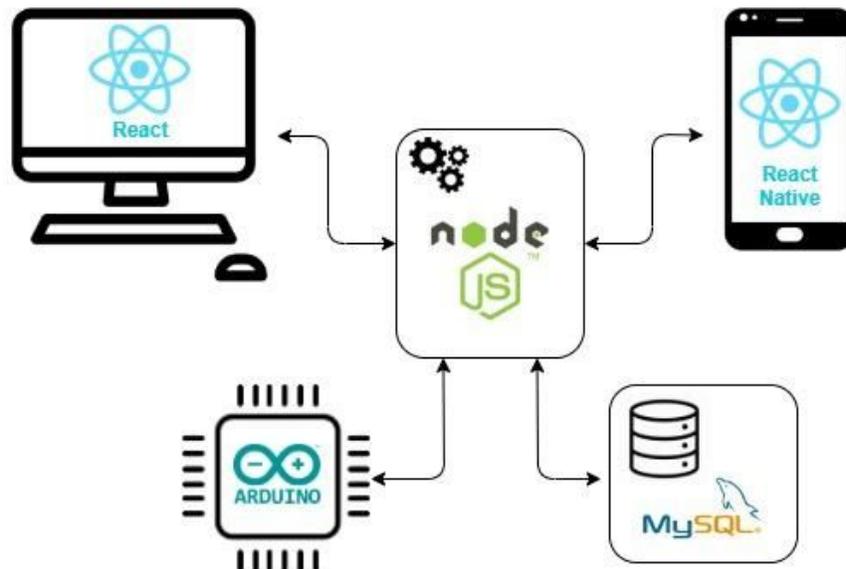
**Figura 7: Protótipo utilizado para os testes**

**Fonte: elaborada pelo autor**

O objetivo de um Arduino com sensores, é coletar dados para algum outro fim. Esses dados podem ser analisados pelo próprio Arduino e, a partir do resultado dessa análise, enviar sinais para atuadores realizarem alguma ação.

Atuadores fazem o caminho inverso de sensores, onde os atuadores transformam sinal elétrico em uma grandeza física, gerando geralmente algum movimento ou som, como por exemplo, trancar e destrancar para portas. Neste protótipo, os sensores coletam os dados e a placa Wemos envia os mesmos, sem tratamento para um servidor, conectado na mesma rede WiFi da placa. Esse sistema é responsável por realizar o tratamento de todas as informações recebidas da placa, registrando as mesmas em um banco de dados e fornecendo informações aos usuários do sistema.

Esse sistema, basicamente possui duas partes conhecidas como Backend e Frontend. O backend é responsável por armazenar e/ou processar os dados para que outras aplicações ou sistemas possam consumir de diversas formas e exibir de forma amigável a alguém. O frontend é responsável por consumir essas informações do backend e fornecer, de forma amigável ao usuário. Na figura 8, é possível identificar como essas diferentes tecnologias se comunicam.



**Figura 8: Stack de Arquitetura do Projeto**

**Fonte: elaborada pelo autor**

Na figura 8, é demonstrada a forma de comunicação das diferentes aplicações bem como a função de cada tecnologia. O React é responsável pela estrutura Web, assim como o React Native é responsável pela estrutura do Aplicativo *Mobile*. Ambas aplicações consomem e alteram os dados a partir de uma API desenvolvida em Node.js, esta que possui todas as regras de negócio e então realiza a comunicação entre todos os dispositivos e envia os comandos de Inclusão, Edição, Exclusão e Consulta ao banco de dados MySQL.

O protótipo físico (Arduino), que coleta os dados da porta, realiza a comunicação com a API da mesma forma que as aplicações *web* e *mobile*, realizando consultas para verificar o status esperado toda vez que a porta sofrer alguma alteração, ou seja, abrir ou fechar. A seguir, será detalhado a criação e funcionamento do sistema backend e, na sequência, a exibição desses dados a partir do frontend.

## 4.2 SISTEMA BACKEND

Uma vez que compreendida a necessidade pela criação deste protótipo, iniciou-se o estudo pelas definições técnicas e funcionais a serem implementadas no sistema. As definições técnicas, foram limitadas ao uso de ferramentas e tecnologias *open source*, tais como Node.js, React e React Native, além do banco de dados MySQL, todas estas apresentadas na seção 2, Abordagem Teórica. As definições funcionais foram divididas em Requisitos Funcionais e Requisitos Não Funcionais, e estes estão descritos abaixo.

Requisitos Funcionais:

- Somente o Administrador do sistema deverá ter acesso ao Sistema Web e este o fará a partir de um único login e senha;
- No Sistema Web, o Cadastro de uma Central de Comando deverá ser aceito somente após o preenchimento do campo “Descrição”;

- No Sistema Web, o Cadastro de Produtos deverá ser aceito somente após o preenchimento dos campos “Descrição”, “Local” e “Identificador da Central de Comando”;
- No Sistema Web, o Cadastro de Pessoas deverá ser aceito somente após o preenchimento dos campos “Nome”, “E-mail”, “Número” e “Senha”;
- No Sistema Web, o Cadastro de um Contato de Emergência deverá ser aceito somente após o preenchimento dos campos “Nome” e “Número”;
- No Aplicativo *Mobile*, a tela de Histórico só poderá ser acessada pelo administrador do protótipo;
- No Aplicativo *Mobile*, a tela de Perfis só poderá ser acessada pelo administrador do protótipo;
- Somente uma Pessoa pode ser administrador do Protótipo;
- No Sistema Web, o sistema não deverá permitir que o campo “E-mail” de uma Pessoa seja inválido ou esteja sem preenchimento;
- No Aplicativo *Mobile*, o Login deve ser realizado a partir do código e senha da Pessoa, rejeitando o acesso quando os dados forem inválidos ou inexistentes;
- No Sistema Web, na edição de Pessoas, deve ser possível inativar e ativar uma Pessoa.

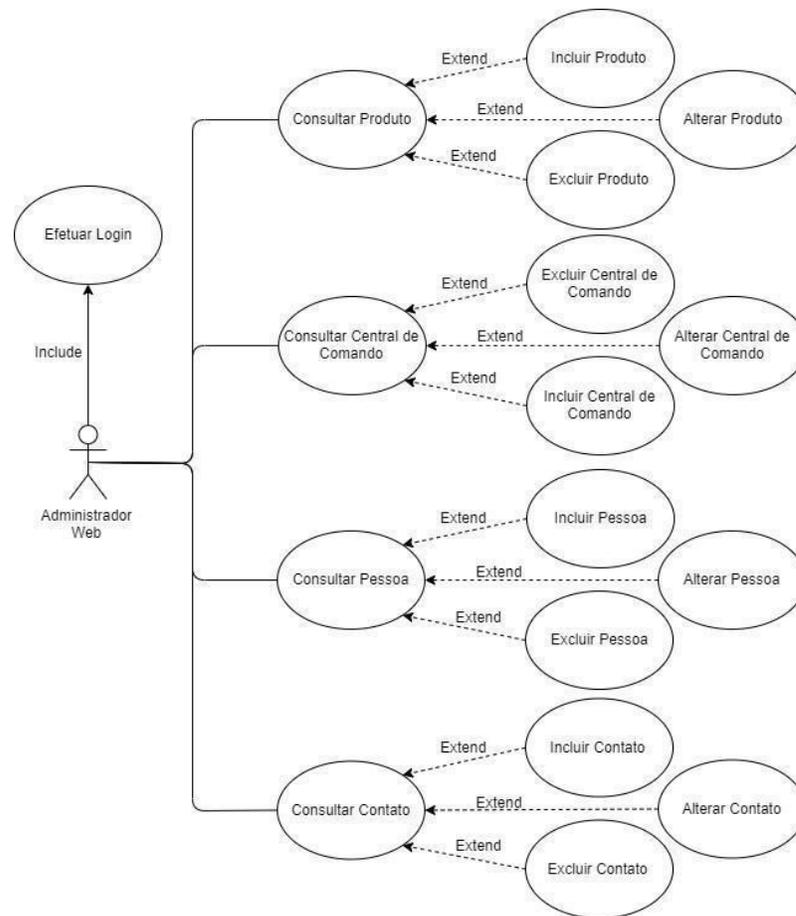
#### Requisitos Não Funcionais:

- Toda vez que um cadastro ou alteração de dados for realizado sem o preenchimento de algum campo obrigatório, o sistema deverá destacar em vermelho os campos com dados inválidos ou sem preenchido, não possibilitando a conclusão da operação;
- Desempenho: o tempo de resposta do Sistema Web é não deve passar de 30 segundos, para a realização de qualquer operação;
- O Sistema não deverá possuir restrições à qualquer sistema operacional que seja utilizado;

- O sistema deve ser responsivo, possibilitando o acesso a partir de computadores, notebooks e demais eletrônicos que possuam acesso a internet.

Levando em consideração os requisitos funcionais descritos anteriormente, o Sistema Web tem como objetivo básico a realização de todos os cadastros necessários para o funcionamento do protótipo. Já o protótipo é dividido em duas permissões possíveis, sendo a primeira permissão do Administrador do Protótipo, que será o dono do produto e decidirá quem serão os contatos de emergência, quem serão as pessoas que poderão ser cadastradas em seu protótipo para ativar e desativar o alarme (moradores da mesma residência, familiares, segurança do prédio, entre outros). As pessoas que forem cadastradas terão apenas duas opções de ações possíveis, sendo a primeira ativar e desativar o alarme, assim como poderão alterar seus dados como nome, número e e-mail. Os requisitos funcionais podem ser visualizados em diversos formatos, sendo um dele o Diagrama de Caso de Uso, apresentado a seguir na figura 9 e 10.

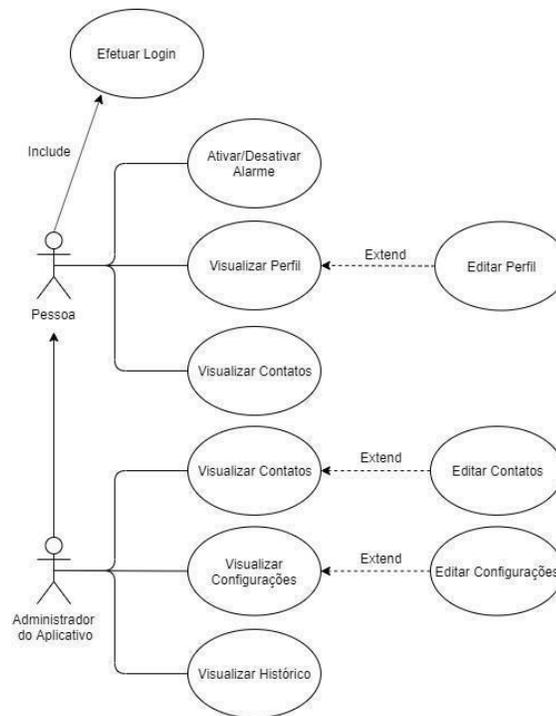
Os requisitos não funcionais, têm como objetivo disponibilizar um padrão a ser seguido para diferentes sistemas operacionais e plataformas em que os sistemas sejam utilizados, tanto para o Sistema Web, quanto para o Aplicativo *Mobile*.



**Figura 9: Diagrama de Caso de Uso - Sistema Web**

**Fonte: elaborada pelo autor**

O Diagrama de Caso de Uso demonstrado na figura 9, representa as possíveis ações que o usuário Administrador poderá executar no Sistema Web. Inicialmente o Administrador precisa efetuar o login no sistema. Após este, pode optar pelas opções de Consultar Produtos, Consultar Central de Comando, Consultar Pessoas ou Consultar Contatos. A partir de cada consulta é possível realizar a inclusão de novos registros, a alteração ou exclusão de registros já existentes. A figura 9 ilustra o Diagrama de Caso de Uso para o Aplicativo *Mobile*.



**Figura 10: Diagrama de Caso de Uso - Aplicativo *Mobile***

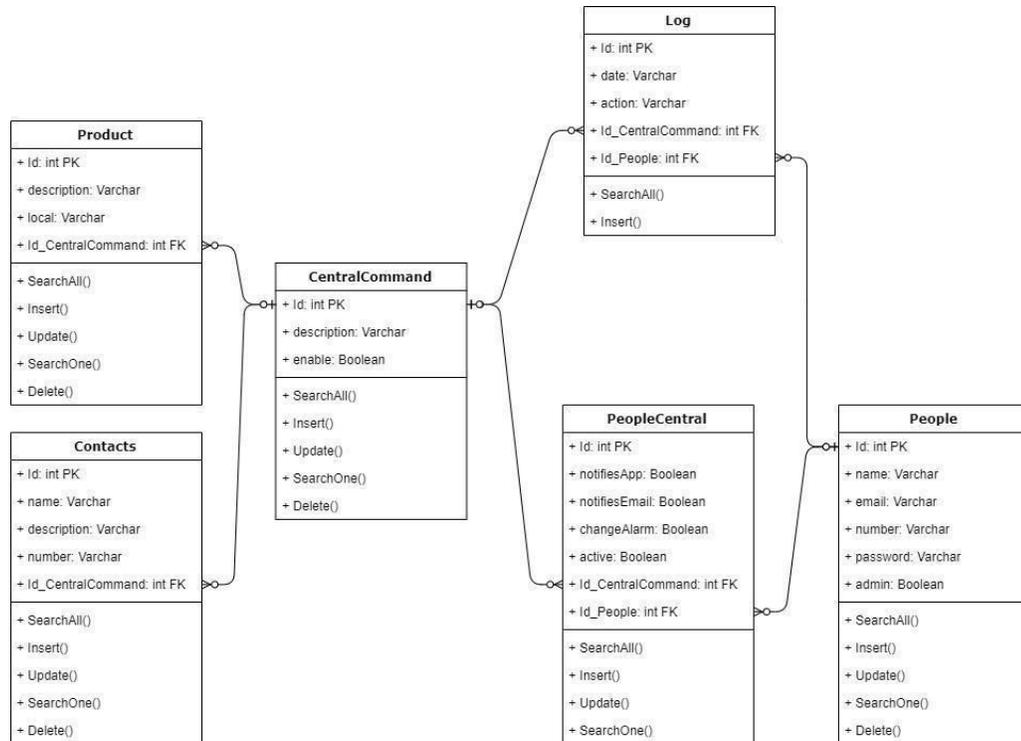
**Fonte: elaborada pelo autor**

O Diagrama de Caso de Uso da figura 10, representa as possíveis ações que uma Pessoa cadastrada como usuário poderá executar no Aplicativo *Mobile*. Inicialmente a Pessoa precisa efetuar o login no sistema. Após este, pode optar pelas opções Ativar/Desativar Alarme ou Visualizar Perfil, onde nesta última, poderá editar seus dados como nome, número e e-mail.

O Administrador do Protótipo, terá acesso a todas as funcionalidades de uma Pessoa e contará também com a opção de Visualizar o Histórico de alterações do alarme, sendo possível identificar o usuário que alterou, assim como a data e horário do mesmo. Além disso, também poderá acessar as configurações do protótipo e nesta terá a listagem de Pessoas cadastradas, possibilitando que o Administrador ative ou inative determinada Pessoa.

Os dois perfis de usuários também poderão visualizar os Contatos de Emergência, porém somente o Administrador do Protótipo poderá ativar ou inativar a exibição de determinado contato. A partir da visão funcional do sistema, foi desenvolvido um Modelo Relacional que representa a arquitetura do banco de dados. Neste modelo (figura 11) é possível identificar o

nome das tabelas, nome dos campos e o tipo de dado correspondente, bem como a ligação entre as tabelas a partir de chaves estrangeiras.



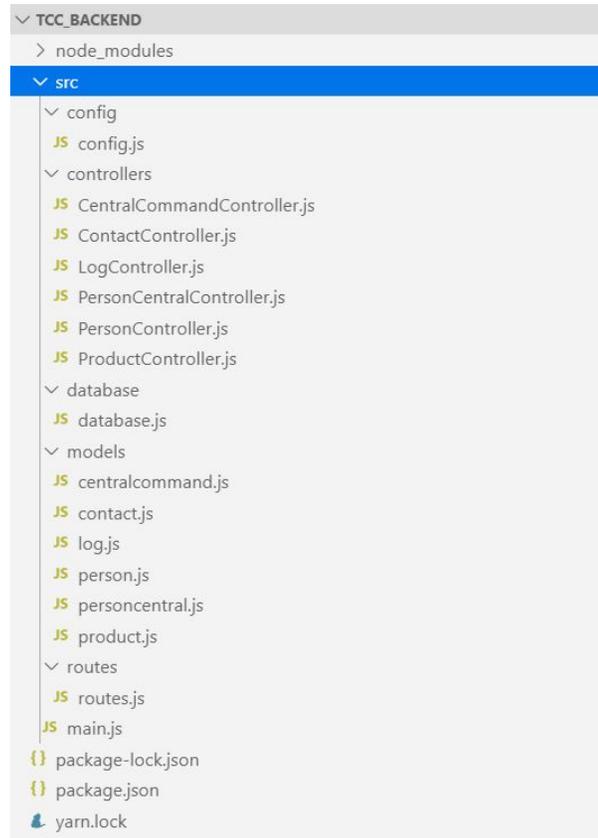
**Figura 11: Modelo Relacional**

**Fonte: elaborada pelo autor**

No modelo demonstrado na figura 11, é possível observar que cada tabela possui uma chave primária, que através das relações com outras tabelas acabam gerando chaves estrangeiras, um exemplo deste fato é a tabela PeopleCentral, criada para realizar a associação de uma pessoa (registro da tabela People) a uma central (registro da tabela CentralCommand). A justificativa para a criação desta tabela é que uma pessoa pode estar associada a mais de uma central, assim como uma central pode conter mais de uma pessoa, assim foi necessário a criação de uma nova tabela e a inclusão de campos específicos desta relação, como por exemplo, o campo “active” que serve para identificar se determinada pessoa está ativa para determinada central.

O desenvolvimento do sistema backend foi realizado com a tecnologia Node.js, conforme detalhado no decorrer deste documento. A ferramenta utilizada nesta etapa foi o Visual

Studio Code, um editor de código simples e intuitivo. A seguir serão apresentados alguns detalhes da estrutura utilizada, bem como o funcionamento de mesma (figura 12).



**Figura 12: Estrutura do Backend**

**Fonte: elaborada pelo autor**

Na figura 12, é possível identificar uma estrutura de pastas e arquivos que está organizada, basicamente, em cinco pastas sendo elas config, controllers, database, models e routes, e dentro de cada pasta possui um ou mais arquivos JavaScripts. A pasta Config é responsável por armazenar os dados de comunicação com o banco de dados, como endereço, porta, usuário e senha. A pasta database é responsável por armazenar as funções utilizadas para realizar a comunicação com o banco de dados, essa estendendo a pasta mencionada anteriormente (config) para utilizar os dados inseridos nela.

A pasta controller, é responsável por armazenar as funções que cada tabela terá, como por exemplo, as funções Insert, Update, Delete e Search. A pasta model, contém as definições utilizadas para a criação das tabelas e colunas no banco de dados. Por fim, a pasta routes, esta é responsável por armazenar todos os caminhos para entrada de dados a partir das funções definidas nas controllers, conforme disponível na figura 13, abaixo.

```
src > routes > JS routes.js > ...
21  router.get('/product', Productcontroller.SearchAll);
22  router.get('/product/:id', Productcontroller.SearchOne);
23  router.post('/product', Productcontroller.Insert);
24  router.put('/product/:id', Productcontroller.Update);
25  router.delete('/product/:id', Productcontroller.Delete);
26
27  router.get('/person', Personcontroller.SearchAll);
28  router.get('/person/:id', Personcontroller.SearchOne);
29  router.post('/person', Personcontroller.Insert);
30  router.put('/person/:id', Personcontroller.Update);
31  router.delete('/person/:id', Personcontroller.Delete);
32
33  router.get('/personcentral', PersonCentralController.SearchAll);
34  router.get('/personcentral/:id', PersonCentralController.SearchOne);
35  router.post('/personcentral', PersonCentralController.Insert);
36  router.put('/personcentral/:id', PersonCentralController.Update);
37  router.delete('/personcentral/:id', PersonCentralController.Delete);
38
39  router.get('/centralcommand', CentralCommandController.SearchAll);
40  router.get('/centralcommand/:id', CentralCommandController.SearchOne);
41  router.post('/centralcommand', CentralCommandController.Insert);
42  router.put('/centralcommand/:id', CentralCommandController.Update);
43  router.delete('/centralcommand/:id', CentralCommandController.Delete);
44
45  router.get('/log', LogController.SearchAll);
46  router.get('/log/:id', LogController.SearchOne);
47  router.post('/log', LogController.Insert);
48  router.put('/log/:id', LogController.Update);
49  router.delete('/log/:id', LogController.Delete);
50
51  router.get('/contact', ContactController.SearchAll);
52  router.get('/contact/:id', ContactController.SearchOne);
53  router.post('/contact', ContactController.Insert);
54  router.put('/contact/:id', ContactController.Update);
55  router.delete('/contact/:id', ContactController.Delete);
56
57  module.exports = router;
```

**Figura 13: Rotas Para Cada Função**

**Fonte: elaborada pelo autor**

Na figura 13, é possível visualizar as rotas utilizadas, em sua grande maioria pelo Sistema Web, cujo qual fornece as funcionalidade de um CRUD completo para cada tabela do banco de dados. Na figura 14, será exibido o código de um Insert. Código esse, que é chamado pelo frontend, a partir de uma das rotas de insert, demonstrada na figura 13, que redireciona para a execução de uma função.

```

src > controllers > JS PersonController.js > Update > exports.Update > then() callback
16 //inserir os dados no banco
17 exports.Insert = (req, res, next) => {
18   //na requisicao de insert
19   //ele retorna um json no corpo
20   //precisamos pegar cada dados e inserir na respectiva propriedade
21   const name = req.body.name;
22   const email = req.body.email;
23   const number = req.body.number;
24   const password = req.body.password;
25   const admin = req.body.admin;
26
27   //aqui passa os parametros com dados para os atributos do model
28   Person.create({
29     name: name,
30     email: email,
31     number: number,
32     password: password,
33     admin: admin
34   })
35   //then = registra o que queremos que acontece quando a Promise for resolvida
36   .then(person => {
37     if (person) {
38       res.status(status.OK).send(person);
39     } else {
40       res.status(status.NOT_FOUND).send();
41     }
42   })
43   //catch = registra o que queremos que acontece quando a Promise falhar
44   .catch(error => next(error));
45 };
46

```

**Figura 14: Insert no banco de dados**

**Fonte: elaborada pelo autor**

Na figura 14, é possível identificar a função Insert, utilizada para realizar o cadastro de informações no banco de dados. Neste caso, está sendo inserido um novo registro na tabela Person, a partir de informações preenchidas nos campos disponíveis em uma tela. O backend recebe essas informações em formato JSON e interpreta cada trecho desse JSON, associando as informações recebidas com as constantes declaradas no código. Finalizada essa leitura é realizado envio das informações para o model, retornando então status OK ou falha.

Para realizar a comunicação entre o protótipo e o sistema backend, é necessário que haja algum método de conexão entre estes, o método utilizado neste trabalho é o WiFi, visto que a placa Wemos D1 R2 conta com um módulo ESP8266 nativo, não havendo a necessidade de conexão da placa com outro sensor. O código desenvolvido para a conexão WiFi está disponível na figura 15.

```

#include <ESP8266WiFi.h> //INCLUSÃO DA BIBLIOTECA UTILIZADA PARA CONEXÃO COM WIFI

const char* ssid = "nome_rede"; //DEFINE UMA VARIÁVEL PARA ARMAZENAR O NOME DA REDE
const char* password = "senha_rede"; //DEFINE UMA VARIÁVEL PARA ARMAZENAR A SENHA DA REDE

void setup(){
  Serial.print("Conectando na rede ");
  Serial.println(ssid);

  WiFi.begin(ssid, password); //COMANDO PARA INICIALIZAR A CONEXÃO COM O WIFI, NESTA FUNÇÃO ESTÁ SENDO ENVIADO O NOME DA REDE E A SENHA DA REDE

  while (WiFi.status() != WL_CONNECTED) { //FUNÇÃO SERÁ EXECUTADA EM LOOPING ATÉ QUE A CONEXÃO SEJA CONCLUÍDA
    delay(200);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
}

void loop(){
  if (WiFi.status() != WL_CONNECTED){ //REPETIÇÃO DO MÉTODO DE CONEXÃO QUE SERÁ EXECUTADO SEMPRE QUE A CONEXÃO COM O WIFI FALHAR
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
      delay(200);
      Serial.print(".");
    }
  }
  else {
    delay(10);
  }
}

```

### Figura 15: Insert no banco de dados

Fonte: elaborada pelo autor

A figura 15, tem como objetivo, demonstrar o código utilizado para a conexão do protótipo com WiFi. Para seu funcionamento, foi utilizada a biblioteca “ESP8266WiFi.h”, esta possui as funções necessárias para a realização da conexão. Logo abaixo, foram declaradas duas variáveis, sendo uma delas com o objetivo de armazenar o Nome da Rede WiFi e a outro armazenar a senha.

A conexão é iniciada a partir da função “WiFi.begin (ssid, password);”, declarada na função setup do projeto, ou seja, toda vez que a aplicação for ligada, será realizada a tentativa de conexão com WiFi. Essa mesma função foi declarada dentro da função loop, a fim de que seja possível identificar quando ocorrerem falhas na conexão e então tentar a conexão novamente.

Esta seção, tem como objetivo demonstrar alguns aspectos funcionamento do backend do sistema, bem como a forma de trabalho e organização do mesmo. Conforme citado no início desta seção, o backend é responsável pelo armazenamento e processamento dos dados originados de outras aplicações (ou não), disponibilizando esses dados já tratados para a exibição. Na seção a seguir, será explanado sobre o frontend, bem como seu funcionamento e uso nesta aplicação.

### 4.3 SISTEMA FRONTEND

O desenvolvimento do sistema frontend foi dividido em duas partes, conforme foi possível identificar na descrição deste projeto. Inicialmente, foi utilizada a tecnologia React, no mesmo editor de texto do backend, o Visual Studio Code, para a criação do Sistema Web. Esse sistema, tem como objetivo realizar apenas os cadastros básicos para o funcionamento do protótipo, entre esses cadastros está o de Pessoa, Central de Comando, Produto e Contatos de Emergência. A seguir, serão apresentadas algumas telas do sistema, bem como a explicação sobre seu funcionamento, iniciando pela tela de cadastro de Pessoas (Figura 16).

A imagem mostra a interface de usuário para o cadastro de uma pessoa. No topo, há uma barra azul com o título "Pessoas". Abaixo, o formulário é intitulado "Cadastrar Pessoa" e contém os seguintes campos:

- Nome: um campo de texto simples.
- E-mail: um campo de texto simples.
- Número: um campo de texto com um ícone de telefone à esquerda.
- Senha: um campo de texto simples.

Um botão azul com o texto "Cadastrar" está localizado na base do formulário.

**Figura 16: Cadastrar Pessoa**

**Fonte: elaborada pelo autor**

Na figura 16, é possível visualizar a tela de Cadastrar Pessoa, com os campos necessários para a realização do cadastro, sendo estes o Nome, E-mail, Número e Senha, disponibilizando o Botão Cadastrar ao final do formulário. Todo o Sistema Web segue o mesmo padrão visual, em relação à disposição dos campos, visto que este é um sistema apenas para o uso do Administrador e foi desenvolvido com o objetivo de evidenciar as funcionalidades necessárias, conforme descrito nos Requisitos Funcionais e Casos de Uso, na seção anterior. A figura 17, está representado o retorno da consulta de pessoas vinculadas a uma mesma central.

Pessoas	
Jean Wilhelm	(55) 99650 6141 Central: Ap 301 Detalhes Remove Editar
Joao	(55) 99650 6141 Central: Ap 301 Detalhes Remove Editar

**Figura 17: Listagem de Pessoas**

**Fonte: elaborada pelo autor**

Na figura 17, é possível observar que as informações disponibilizadas a partir de uma consulta são nome, número e descrição da central em que foi vinculado. Ao lado de cada registro também é disponibilizado a opção de Visualização dos Detalhes, que retorna como resultado todos os dados da pessoa, a opção Remove, que irá inativar o usuário, mantendo o mesmo na base de dados, porém como inativo, e a opção Editar, que disponibiliza todos os campos da pessoa para edição.

A segunda parte do desenvolvimento do frontend foi realizada com a tecnologia React Native, utilizando o mesmo editor de códigos, Visual Studio Code, porém para emular o funcionamento do aplicativo, foi utilizado o Android Studio. O Aplicativo *Mobile* atualmente se encontra em fase de construção, desta forma não foram apresentadas telas ilustrativas, porém suas especificações e funcionalidades foram definidas anteriormente no Diagrama de Caso de Uso (figura 10) e a partir de alguns requisitos funcionais.

Esse aplicativo, consistirá basicamente em dois níveis de acesso, sendo Administrador do Aplicativo e Pessoa. Inicialmente, a pessoa deverá realizar o login no sistema a partir de seu código de usuário e senha. Na tela inicial do aplicativo estará disponível a opção de Ativar/Desativar Alarme, bem como um menu lateral com opções para visualização e edição de perfil, além de visualização dos Contato de Emergência. O Administrador, por sua vez, poderá executar todas as ações disponíveis à Pessoa, assim como também poderá editar os Contatos de

Emergência, Visualizar o Histórico de alterações de estado do alarme e poderá também acessar as configurações do protótipo, que disponibiliza a lista de pessoas associadas a esse protótipo, podendo inativar ou ativar uma pessoa, de acordo com a necessidade.

A partir desta seção, serão analisados os resultados obtidos até o momento, levando em consideração os testes executados com o protótipo, com o Sistema Web e com o Backend. Além destes, será realizada uma análise comparativa dos resultados obtidos em relação aos resultados dos trabalhos relacionados, citados na seção 2.4.

## 5 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Conforme apresentado no presente trabalho, os resultados obtidos até o momento puderam ser estimados somente em relação ao Sistema Web e ao Backend, além do protótipo Arduino funcional, porém sem realizar a integração e comunicação com o aplicativo em React Native. O protótipo do Arduino foi testado a partir de testes exploratórios e os resultados das leituras do sensor, identificadas a partir de LEDs associados e do monitor serial do Arduino IDE. Também foi possível verificar a funcionalidade na conexão com WiFi, assim como o seu comportamento perante problemas na rede WiFi (desligar o roteador e causar instabilidades na rede).

A partir da execução dos cenários de testes elaborados para o Sistema Web, foram executados os dez cenários propostos, obtendo como retorno o resultado esperado de cada cenário. Entende-se que, a partir destes, sua funcionalidade está adequada à necessidade descrita neste projeto e, em relação aos requisitos funcionais propostos para o Sistema Web, quatro foram atendidos com sucesso (todos relacionados aos cadastros validados nos cenários), os requisitos relacionados à permissão de acesso foram aceitos, porém não validados no aplicativo.

Com base nos resultados obtidos até o momento, detalhados anteriormente, compreende-se que o sistema backend atendeu as necessidades e objetivos esperados, uma vez que todos os registros básicos foram cadastrados e manipulados sem problemas. Novos testes se farão necessários a partir da finalização do aplicativo *mobile*, bem como a integração de todos os sistemas e então a implantação do protótipo para ser validado em cenários reais, associando ao dia-a-dia do usuário.

Realizando uma análise em alto nível, comparando os resultados obtidos até o momento em relação aos Trabalhos Relacionados, citados na seção 2.4, o autor pode identificar maior segurança e integridade na coleta de dados, por parte do sensor utilizado, conforme previsto na justificativa pela escolha do sensor Magnético em relação ao sensor de movimento PIR. Quando comparado à aplicabilidade, facilidade de uso e solução para a problemática encontrada, não se

pode chegar a uma conclusão, visto que o mecanismo principal de comando do protótipo é o Aplicativo *Mobile*.

## 6 CONSIDERAÇÕES FINAIS

O trabalho apresentado tem como proposta a criação e implantação de um sistema de alarme residencial com ferramentas *open-source*, a partir da plataforma Arduino, assim como realizar seus cadastros básicos através de um sistema web e seu monitoramento a partir de um aplicativo *mobile*. O desenvolvimento deste projeto surgiu a partir de uma necessidade do autor, que posteriormente foi identificada como uma necessidade de demais moradores que residem na localidade de Recanto Maestro, campus universitário com fluxo diário muito grande de visitantes e alunos.

Durante o período de desenvolvimento algumas limitações foram encontradas, dentre essas limitações, as que mais tiveram impacto no resultado obtido foi o baixo conhecimento do Arduino escolhido para prototipação, neste caso a Wemos D1 R2, que por sua vez geraram alguns erros para que pudesse dar início ao desenvolvimento, além disso, o baixo conhecimento do autor em desenvolvimento de software também prejudicou, fazendo com que a curva de aprendizagem fosse mais demorada, assim, não sendo possível a finalização do projeto em tempo hábil.

Para a implementação de trabalhos futuros, espera-se finalizar o desenvolvimento e testes do projeto, conforme proposto. Além disso, observou-se a necessidade de remodelar o Sistema Web de modo que facilite o gerenciamento das informações cadastradas, uma vez que a estrutura atual foi pensada apenas como um sistema que realiza cadastros, porém encontrou-se a necessidade de gerir esses dados, a partir da criação de cópias desse protótipo.

Outra necessidade observada, foi a utilização de armazenamento em nuvem. Atualmente o sistema backend está armazenado em um servidor local de acesso restrito, fazendo com que o protótipo só funcione caso esteja conectado na mesma rede WiFi do servidor.

Por fim, criar uma identidade visual para o projeto, transformando o mesmo em um possível produto a ser comercializado localmente para suprir essa necessidade. A construção do projeto foi guiada pelo uso de templates, tanto para o sistema Web quanto para o aplicativo, assim faz-se necessária a definição de uma identidade visual, bem como a adaptação do mesmo a

essa nova identidade. Além do visual da aplicação, entende-se necessária a criação de uma estrutura física que comporte o protótipo, ofuscando os fios e as placas utilizadas no projeto.

## 7 REFERÊNCIAS

AGNOL, Cleiton Dal. Comparação entre Microcontroladores e Aplicação do FPGA no Controle do Conversor BOOST. Centro Universitário UNIFACVEST. Curso de Engenharia Elétrica. Lages, Santa Catarina, 2018.

ALECRIM, Emerson. Banco de dados MySQL e PostgreSQL. [S.l.], 2008. Disponível em: <<http://www.infowester.com/postgremysql.php>>. Acesso em 21 julho 2019.

ALMEIDA, Wellington Márcio. Gestão da Qualidade – Curso Técnico de Manutenção. Ouro Preto, SENAI-MG, 2012. 68 p. EDUCAÇÃO PROFISSIONAL.

ANTONIO, C. Pro React: Build Complex Front-End Applications in a Composable Way With React. Apress, 2015.

AURESIDE, 2016. Disponível em: <http://www.aureside.org.br/>. Acesso em 07 julho 2019.

BACKENDLESS. WHAT IS BACKEND AS A SERVICE? 2017. Disponível em: <<https://backendless.com/what-is-backend-as-a-service/>>. Acesso em 20 outubro 2019.

Banzi, M. (2010). Primeiros passos com o arduino (Primeira ed.). São Paulo, SP, Brasil: Novatec Editora Ltda.

BEGHINI, Lucas Bragazza. Automação Residencial de baixo custo por meio de dispositivos móveis com sistema operacional Android. Escola de Engenharia de São Carlos, Universidade de São Paulo, 2013.

BEGHINI, L. B. Automação Residencial de baixo custo por meio de dispositivos móveis com sistema operacional Android, nov. 2013. Disponível em: <<http://goo.gl/K39oLz>>. Acesso em 07 outubro. 2019.

BOLZANI, C.A.M. RESIDÊNCIAS INTELIGENTES. São Paulo: Livraria da Física, 2004.

BOLZANI, C.A.M. Desmistificando a Domótica. Sinergia, São Paulo, v. 8, n. 1, p. 17-20, jan./jun. 2007 Disponível em: <<http://goo.gl/PEHUUo>>. Acesso em 16 de setembro 2019.

BORGES, L. P.; DORES, R. C., Automação predial sem fio utilizando bacnet/zigbee com foco em economia de energia. 2010, 76f. Trabalho de conclusão de curso - Curso de Graduação em Engenharia de Controle e Automação – UNB, Brasília, 2010.

BRANKS A.; PORCELLO E. Learning React: Functional Web Development with React and Redux. Sebastopol, United States of America: O'Reilly, 2017.

BRUGNARI, A.; MAESTRELLI, L. H. M., AUTOMAÇÃO RESIDENCIAL via WEB. 2010, 36f. Trabalho de conclusão de curso - Curso de Graduação em Engenharia de Computação - PUC-PR, Curitiba, 2010.

BUKMAN, Guilherme. Desenvolvimento de um protótipo de baixo custo para segurança residencial. Colegiado do Curso de Engenharia de Controle de Automação, Universidade Federal de Ouro Preto, 2016.

CABRAL, Carlos. React Native: Construa aplicações móveis nativas com JavaScript. 2016. Disponível em: <<https://tableless.com.br/react-native-construa-aplicacoes-moveis-nativas-com-javascript/>>. Acesso em 13 agosto 2019.

CASADOMO. Domótica - Introducción. Disponível em: <<http://www.casadomo.com/>>. Acesso em 13 agosto 2019.

CANTELON, M. et al. Node. js in Action. [S.l.]: Manning Publications, 2017.

CHAN, Iana. O que é front-end e back-end?. 2016. Disponível em: <<https://www.programaria.org/o-que-e-front-end-e-back-end/>>. Acesso em 11 agosto 2019.

CURY, T. E.; HIRSCHMANN D. R. .Ensino de matemática através do Arduino. Disponível em: <<http://goo.gl/wLOubO>>. Acesso em 22 agosto 2019.

Dayley, Brad (2014) “Node.js, MongoDB and AngularJS Web Development”, Michigan, Pearson Education.

DIAS, C. L. A.; PIZZOLATO, N. D. DOMÓTICA. Aplicabilidade e Sistemas de Automação Residencial. VÉRTICES, v. 6, n. 3, set./dez. 2004. Disponível em: < <http://goo.gl/tCIqQw>>. Acesso em 24 setembro 2019.

ELLIOTT E. Programming JavaScript applications: Robust Web Architecture with Node, HTML5, and a Modern JS Library. Sebastopol, United States of America: O'Reilly, 2014.

ELMASRI, Ramez, Sistemas de banco de dados. 4. ed. São Paulo: Pearson Addison Wesley, 2005.

EVANS, M.; NOBLE, J.; HOCHENBAUM, J. Arduino em ação. Editora Novatec, São Paulo, p. 25-26, ago. 2013.

FELIX, Waldyr. 5 motivos para começar a usar Node.js em seus projetos HOJE. 2016. Disponível em: <<https://waldyrfelix.com.br/5-motivos-para-come%C3%A7ar-a-usar-node-js-emseus-projetos-hoje-678f4db7b93>>. Acesso em 20 outubro 2019.

GODOY, A. S. Introdução à pesquisa qualitativa e suas possibilidades. Revista de Administração de Empresas São Paulo, v. 35, n. 2, p. 57-63, mar./abr. 1995. Disponível em: <<http://goo.gl/uJs6Hu>>. Acesso em 15 novembro 2019.

GONZAGA, Flávio S; BIRCKAN, Guilherme. Curso de PHP e MySQL, 2000. Florianópolis, SC.

HAYERBEK, Marijn. Eloquent Javascript - 2 edição. Disponível em: <<https://goo.gl/b9ymuS>>, Acesso em 14 de outubro de 2019.

KOSTRZEWA, D. Is React.js the Best JavaScript Framework in 2018? 2018. Disponível em: <[hackernoon.com/is-react-js-the-best-javascript-framework-in-2018-264a0eb373c8](http://hackernoon.com/is-react-js-the-best-javascript-framework-in-2018-264a0eb373c8)>. Acesso em 20 outubro 2019.

KUPKA, Fernando. O que é React Native?. 2017. Disponível em: <<https://www.organicadigital.com/seeds/o-que-e-react-native/>>.

LÜDKE, Menga; ANDRÉ, Marli Elisa Dalmazo Afonso de. Pesquisa em Educação: abordagens qualitativas. São Paulo: Editora Pedagógica e Universitária (EPU), 1986.

MAIA, G. M. F. Acionamento remoto de portões elétricos via celular através de microcontrolador. Centro Universitário de Brasília -UniCEUB, jun. 2012. Disponível em: <<http://goo.gl/OhapMa>>. Acesso em 22 outubro 2019.

MARCHESAN, Marcelo. Sistema de Monitoramento Residencial Utilizando a Plataforma Arduino, 2012. Trabalho de Conclusão de Curso - Curso Superior de Tecnologia em Redes e Computadores - UFSM (Universidade Federal de Santa Maria), Santa Maria, RS.

MARIANO, C. L. Benchmarking JavaScript frameworks. Tese (M.Sc. in Computing - Advanced Software Development). Dublin Institute of Technology. Dublin, Irlanda: 2017.

MARTE, Claudio Luiz. Automação Predial: A Inteligência Distribuída Nas Edificações. São Paulo: Carthago & Forte, 1995.

MARTINS, C. S, 2013. Os desafios para a mobilização de aplicações baseadas em plataforma Web. X Encontro Anual de Computação - EnAComp 2013, Fundação CPqD - Centro de Pesquisa e Desenvolvimento em Telecomunicações Campinas - SP - Brasil.

MARTINS, N. A. Sistemas Microcontrolados: Uma Abordagem com o Microcontrolador PIC 16F84. São Paulo: Novatec, 2005.

MATUCHAKI, G. Projeto de um sistema de climatização de aviário, 2011. Disponível em: < <http://goo.gl/IR5WJP> >. Acesso em 30 outubro 2019.

MINAYO, M. C. S. O desafio do conhecimento. Pesquisa qualitativa em saúde. São Paulo: HUCITEC, 2007.

MONK, Simon. Programação com Arduino: começando com sketches. Porto Alegre: Bookman, 2013.

MOZILLA. About JavaScript. 2017. Disponível em: <[https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About\\_JavaScript](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About_JavaScript)>. Acesso em 20 outubro 2019.

MURATORI, J. R.; HENRIQUE, P. H. D. Automação residencial: histórico, definições e conceitos. Disponível em <<http://goo.gl/dMBh4o>>. Acesso em 22 outubro 2019.

NODEJS. 2017. Disponível em: <<https://nodejs.org/>>. Acesso em 20 outubro 2019.

NOGUEIRA, C. L. M.; ALARCÃO, L. M.; BRAZ, A. R. B.; MOREIRA, J. R. AutoControl: uma proposta para acessibilidade e segurança residencial com o apoio da plataforma Arduino, jun. 2013. Disponível em: <<http://goo.gl/ULNii2>>. Acesso em 26 outubro 2019.

OCCHINO, Tom. React Native: Bringing modern web techniques to mobile, 2015. Disponível em: <<https://code.fb.com/android/react-native-bringing-modern-webtechniques-to-mobile/>>. Acesso em 20 outubro 2019.

OLIVEIRA, Igor. React Native é mesmo nativo?. 2017. Disponível em: <<http://programadorbr.com/blog/react-native-e-mesmo-nativo/>>. Acesso em 20 outubro 2019.

PANDIT, N. What Is ReactJS and Why Should We Use It? 2018. Disponível em: <[www.c-sharpcorner.com/article/what-and-why-reactjs](http://www.c-sharpcorner.com/article/what-and-why-reactjs)>. Acesso em 12 agosto 2019.

PEDRASSANI, C. E. Uma solução em NodeJS e React Native para busca de oferta de emprego. Trabalho de Conclusão do Curso de Sistemas de Informação, AMF, 90p., 2018.

PENIDO É. C. C.; TRINDADE R. S., Microcontroladores, Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, 2013.

PEREIRA, Fernanda. HTML, CSS e Javascript – Entendendo melhor a base da programação Front-End. 2018. Disponível em: <<http://apexensino.com.br/html-css-e-javascript-entendendo-melhor-base-da-programacao-front-end/>> Acesso em 11 agosto 2019.

PEREIRA, Luiz Antônio de Moraes, 2007. Automação Residencial: rumo a um futuro pleno de novas soluções. Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio).

PINHEIRO, José Maurício Santos, 2006. ZigBee em Home Area Network. Projeto de Redes. Disponível em: <http://www.projetoderedes.com.br>. Acesso em 20/09/2019.

PLURALSIGHT. What's the Difference Between the Front-End and BackEnd? 2015. Disponível em: <<https://www.pluralsight.com/blog/film-games/whats-difference-front-end-back-end>>. Acesso em 20 outubro 2019.

QUINDERÉ, Patrick Romero Frota, 2009. Casa Inteligente – Um Protótipo de Sistema de Automação Residencial de Baixo Custo. Ciência da Computação, Faculdade Farias Brito, Fortaleza.

RIBEIRO, M. ISABEL. Sensores em robótica, Enciclopédia Nova Activa Multimédia, Volume de Tecnologias. Portugal, 2004.

SANTOS, Guilherme. Node.js — O que é, por que usar e primeiros passos. 2016. Disponível em: <<https://medium.com/thdesenvolvedores/node-js-o-que-%C3%A9-por-que-usar-e-primeiros-passos-1118f771b889>>. Acesso em 9 de maio de 2018.

SILVA, Bruna Roberta Seewald. Sistema de automação residencial de baixo custo para redes sem fio. Instituto de Informática, Curso de Engenharia de Computação, Universidade Federal do Rio Grande do Sul, 2014.

SILVA, Denys A.; SOUSA, Caio F.. CONSTRUÇÃO DE APP COM REACT NATIVE. Revista Tecnologias em Projeção, v10, n°1, ano 2019.

STACKOVERFLOW. Most Popular Technologies: Web Frameworks. Developer Survey Results, StackOverflow, 2019. Disponível em: <[insights.stackoverflow.com/survey/2019#technology](https://insights.stackoverflow.com/survey/2019#technology)>. Acesso em 13 abril 2019.

TECHINBRAZIL. Integradores da Automação Residencial. Brasil, 2015. Disponível em: <<https://techinbrazil.com.br/integradores-de-automacao-residencial-no-brasil>>. Acesso em 11 agosto 2019.

TERUEL, E. C. Uma proposta de framework para sistemas de automação residencial com interface para WEB. 2008. 158 f. Dissertação (Mestrado em Tecnologia: Gestão, Desenvolvimento e Formação) – Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2008.

VENTURINI et al. Projeto de um sistema de medição e controle de temperatura De baixo custo para ser utilizado na etapa de mosturação do Processo de fabricação de cervejas artesanais. CIPPUS, v. 7, n. 1, p. 1-12, 2019.