



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

GEORG AUGUSTO SCHLEGEL

**PROTÓTIPO DE IOT PARA SMART AGRICULTURE COM FOCO EM
MONITORAMENTO METEOROLÓGICO**

**Assis/SP
2020**



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

GEORG AUGUSTO SCHLEGEL

**PROTÓTIPO DE IOT PARA SMART AGRICULTURE COM FOCO EM
MONITORAMENTO METEOROLÓGICO**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando(a): Georg Augusto Schlegel

Orientador(a): Prof. MSc. Guilherme de Cleve Farto

**Assis/SP
2020**

FICHA CATALOGRÁFICA

S339p SCHLEGEL, Georg Augusto.

Protótipo de iot para smart agriculture com foco em monitoramento meteorológico / Georg Augusto Schlegel. – Assis, 2020.

71p.

Trabalho de conclusão do curso (Análise e Desenvolvimento de Sistemas). – Fundação Educacional do Município de Assis - FEMA.

1.Smart agriculture 2.Monitoramento 3.Internet-coisas.

Orientador: Me. Guilherme de Cleva Farto.

CDD006.3

PROTÓTIPO DE IOT PARA SMART AGRICULTURE COM FOCO EM MONITORAMENTO METEOROLÓGICO

GEORG AUGUSTO SCHLEGEL

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____
Prof. MSc. Guilherme de Cleve Farto

Examinador: _____
Prof. Esp. Célio Desiró

RESUMO

A proposta desse trabalho é apresentar uma solução de baixo custo e simples utilização: Um protótipo de IoT para Smart Agriculture, com hardware desenvolvido na placa NodeMCU, onde todos os dados coletados são enviados ao banco de dados na nuvem e, através desta, podem ser acessados em tempo real. Tais dados serão consumidos pela aplicação WebApp, desenvolvida no NodeJS e ReactJS, e, após, apresentados ao usuário. O protótipo supramencionado atua com foco no monitoramento meteorológico e é capaz de realizar previsões em tempo real, com mobilidade, e contemplando, ainda, a possibilidade futura de que uma cultura desenvolva algum tipo de doença e a estimativa de produção e de crescimento por meio de deduções, a partir de dados, ações ou informações previamente apresentadas, auxiliando, assim, a tomada de decisão do usuário.

Palavras-chave: Smart Agriculture, Monitoramento Meteorológico, Internet das Coisas.

ABSTRACT

The proposal of this work is to present a solution of low cost and simple use: a prototype of IoT for Smart Agriculture, with hardware developed on the NodeMCU board, where all the data collected is sent to the database in the cloud and, through it, can be accessed in real time. Such data will be consumed by the WebApp, developed in NodeJS and ReactJS, and then presented to the user. The above-mentioned prototype focuses on meteorological monitoring and is capable of making real-time forecasts, with mobility, and also contemplating the future possibility of a culture developing some type of disease and the estimate of production and growth through deductions, from data, actions or information previously presented, thus helping the user's decision making.

Keywords: Smart Agriculture, Meteorological Monitoring, Internet of Things.

LISTA DE ILUSTRAÇÕES

Figura 1: Módulo ESP-12E.....	22
Figura 2: Requisitos de energia.....	23
Figura 3: Periféricos e E/S.....	24
Figura 4: Interruptores e indicadores.....	25
Figura 5: Comunicação serial.....	25
Figura 6: Pinagem ESP8266 NodeMCU.....	26
Figura 7: Sensor DHT22.....	28
Figura 8: Sensor BMP280.....	29
Figura 9: Sensor UVM-30A.....	30
Figura 10: Sensor BH1750.....	30
Figura 11: Sensor de Umidade do Solo Capacitivo.....	31
Figura 12: Sensor Hall US1881/U18.....	32
Figura 13: Sensor Reed Switch.....	33
Figura 14: Lipo-Po Rider.....	34
Figura 15: Estrutura Analítica do Projeto.....	36
Figura 16: Sequenciamento das atividades.....	37
Figura 17: Mapa Mental.....	40
Figura 18: Diagrama de Caso De Uso do Administrador e Usuário.....	42
Figura 19: Diagrama de atividade para cadastrar área.....	55
Figura 20: Diagrama de atividade para cadastrar doenças.....	56
Figura 21: Diagrama de sequência cadastrar área.....	57
Figura 22: Diagrama de sequência cadastrar doenças.....	58
Figura 23: Diagrama de classe.....	59
Figura 24: Diagrama de ER.....	60
Figura 25: Protótipo do WebApp.....	62
Figura 26: Modelagem NodeMcu ESP-12E e sensores do protótipo.....	63
Figura 27: Modelagem do modulo gerenciador de energia do protótipo.....	63
Figura 28: Protótipo NodeMcu ESP-12E e sensores do protótipo.....	64
Figura 29: modulo gerenciador de energia do protótipo.....	64
Figura 30: Pluviômetro de balança digital.....	65
Figura 31: Anemômetro.....	65
Figura 32: Versão final do protótipo.....	66

LISTA DE TABELAS

Tabela 1: Custo análise e custo programação.	37
Tabela 2: Custo linguagem e custo banco de dados.	38
Tabela 3: Custo de computadores, aluguel, água, luz e Internet.....	38
Tabela 4: Custo do protótipo IoT.....	39
Tabela 5: Custo total.	39
Tabela 6: UC 01 - Realizar login.	42
Tabela 7: UC 02 - Manter usuário.....	44
Tabela 8: UC 03 - Manter propriedade.....	46
Tabela 9: UC 04 - Manter área.....	48
Tabela 10: UC 05 - Manter estação.....	49
Tabela 11: UC 06 - Manter cultura.....	51

SUMÁRIO

1. INTRODUÇÃO	12
1.1.OBJETIVOS	14
1.2.JUSTIFICATIVAS	14
1.3.MOTIVAÇÃO	15
1.4.PERSPECTIVAS DE CONTRIBUIÇÃO	15
1.5.METODOLOGIA DE PESQUISA	16
1.6.RECURSOS NECESSÁRIOS	16
1.7.ESTRUTURA DO TRABALHO	17
2. TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO	19
2.1.JAVASCRIPT	19
2.2.REACT	19
2.3.REDUX	20
2.4.WEBPACK	20
2.5.BABEL	20
2.6.NODE	21
2.7.FIREBASE	21
2.8.PLACA DE DESENVOLVIMENTO E SENSORES	22
2.8.1. NodeMCU	22
2.8.2. DHT22.....	28
2.8.3. BMP280	28
2.8.4. UVM-30A.....	29
2.8.5. BH1750	30

2.8.6. Umidade do Solo Capacitivo.....	31
2.8.7. Hall US1881/U18	31
2.8.8. Reed Switch.....	32
2.8.9. Lipo-Po Rider.....	33
3. ESTRUTURA DO PROJETO.....	35
3.1. ESTRUTURA ANALÍTICA DE TRABALHO	35
3.2. SEQUENCIAMENTO DAS ATIVIDADES	36
3.3. ORÇAMENTO	37
4. ANÁLISE E ESPECIFICAÇÃO DO SISTEMA	40
4.1. MAPA MENTAL	40
4.2. LISTA DE REQUISITOS	41
4.3. DIAGRAMA E ESPECIFICAÇÃO DE CASOS DE USO	41
4.3.1. UC 01 - Realizar login	42
4.3.2. UC 02 - Manter usuário.....	43
4.3.3. UC 03 - Manter propriedade.....	44
4.3.4. UC 04 - Manter área	46
4.3.5. UC 05 - Manter estação	48
4.3.6. UC 06 - Manter cultura.....	49
4.3.7. UC 07 - Manter doenças.....	51
4.3.8. UC 08 - Emitir relatórios de dados meteorológicos	53
4.4. DIAGRAMA DE ATIVIDADES	54
4.5. DIAGRAMA DE SEQUENCIA	57
4.6. DIAGRAMA DE CLASSE.....	58
4.7. MODELO ENTIDADE-RELACIONAMENTO	59

5. DESENVOLVIMENTO DO PROJETO.....	61
5.1.PROTÓTIPO WEBAPP	61
5.2.PROTÓTIPO IOT.....	62
5.3.DIFICULDADES ENCONTRADAS	67
6. CONCLUSÕES.....	68
6.1.TRABALHOS FUTUROS	68
REFERÊNCIAS	69

1. INTRODUÇÃO

Na atual conjuntura, a agricultura atravessa um período de diversas mudanças em que se experimenta uma grande transição da agricultura de precisão para a agricultura de decisão. Ao passo em que muitos agricultores possuem à sua disposição tecnologias e ferramentas associadas ao acesso à informação, conhecimento e crescimento do nível de educação, há, ainda, a preocupação em combater as mais variadas consequências desse desenvolvimento, tais como as mudanças climáticas, a degradação do meio ambiente, a insegurança alimentar, a utilização de insumos de origem fóssil, o aumento de doenças e a necessidade de aumento da produtividade agrícola.

Há mais seres humanos vivos na Terra do que nunca - 7,7 bilhões - e esse número ainda está crescendo, uma população dessa magnitude traz muitos desafios. Projeções já conhecidas da Organização das Nações Unidas para Alimentação e Agricultura indicam que a população mundial alcançará cerca de 9,7 bilhões de pessoas em 2050. Para esse panorama, (FAO, 2019) estima que haverá necessidade de aumentar em 70% a produção de alimentos até 2050. Isso se justifica não somente pelo aumento do número de consumidores no planeta, mas, entre outros fatores, por já existirem atualmente no globo 870 milhões de pessoas vítimas da fome (GREBMER, HEADEY, *et al.*, 2013).

O mundo moderno e globalizado remete a todos a uma procura incessável pela economia ideal, no qual reúne aspectos sustentáveis e de justiça, *in scrito sensu*, razões pelas quais a bioeconomia ganhou força e notoriedade, uma vez que a sustentabilidade configura, indiscutivelmente, uma das maiores prioridades da sociedade quando se fala em desenvolvimento econômico (MASSRUHÁ, LEITE, *et al.*, 2014).

Nessa circunstância, os avanços em tecnologia de informação e comunicação (TIC), gradativamente, terão um caráter estratégico e político tanto no cenário nacional, quanto no internacional, tendo em vista que estes buscarão, através da implantação de políticas públicas, a adequação do desenvolvimento a nova realidade. (ROCO e BAINBRIDGE, 2003) mencionam que a sinergia entre nanotecnologia, TIC, biotecnologia e as ciências cognitivas, representarão a principal forma de se garantir o futuro da humanidade.

A TIC viabiliza, entre outras atividades, a aplicação de novas ferramentas tecnológicas no setor produtivo agrícola. Atualmente, com o auxílio dessa nova gama de tecnologias, tornou-se totalmente plausível e prático adotar o uso smartphones com o fito de planejar o

uso da área na implantação de lavouras e empregar sistemas computacionais web para a realização de gestão da propriedade com processamento de dados em tempo real, o que possibilita, inclusive, uma antecipação das necessidades do produto.

Aumentar a produção nesse nível não é fácil, mas os engenheiros e agricultores de hoje estão trabalhando juntos para criar uma solução tecnológica: a agricultura de precisão e a “fazenda inteligente”. A agricultura é a mais antiga indústria humana, mas certamente não é estranha à mudança tecnológica. As revoluções industriais dos séculos XIX e XX substituíram ferramentas portáteis e arados puxados por cavalos por motores a gasolina e fertilizantes químicos. Agora, estamos à beira de testemunhar outra mudança fundamental na agricultura graças a uma nova revolução industrial e às tecnologias da Indústria 4.0. Esta quarta grande revolução está assumindo o mundo agrícola com base na aplicação combinada de soluções de TIC, como agricultura de precisão, planejamento de safra, monitoramento de pragas e doenças, sensores e atuadores, sistemas de geolocalização, Internet das Coisas (IoT), Big Data, veículos aéreos não tripulados (UAVs, drones), robótica, etc. (BROWN, 2019).

As aplicações da Smart Agriculture não visam apenas explorações agrícolas convencionais e de grande porte, mas também podem ser novas alavancas para impulsionar outras tendências comuns ou crescentes em outros setores agrícolas, como a agricultura familiar (espaços pequenos ou complexos, culturas específicas, preservação de altas variedades de qualidade ou particulares) e agricultura biológica. A Smart Agriculture também pode oferecer grandes benefícios em termos de questões ambientais, por exemplo, através de um uso mais eficiente da água ou otimização de tratamentos e insumos. Dentro deste contexto, almeja-se fazer um estudo exploratório acerca do setor agrícola, mostrando como tecnologia da informação e comunicação, atrelado à agricultura, por meio da integração das aplicações digitais, podem auxiliar o setor agrícola em coleta de dados históricos, planejamento da safra, recomendação de produtos, dentre outras.

Devido às dificuldades existentes, o protótipo de IOT para Smart Agriculture com foco em monitoramento meteorológico e a aplicação WebApp para consumir os dados do protótipo, objetivam disponibilizar um produto capaz de realizar previsões acerca da possibilidade futura de que uma cultura desenvolva algum tipo de doença, estimativa de produção e de crescimento por meio de deduções, a partir de dados, ações ou

informações previamente apresentadas, contribuindo, assim, para a tomada de decisão do usuário.

1.1. OBJETIVOS

Este trabalho científico apresenta como objetivo desenvolver um protótipo de IOT para Smart Agriculture com foco em monitoramento meteorológico e, também, desenvolver uma aplicação WebApp para consumir os dados do protótipo, com o uso de sensores de temperatura, umidade do ar e do solo, pressão atmosférica, altitude, luz ultravioleta, luminosidade, e dois módulos, um pluviômetro de báscula digital e um anemômetro. Após a instalação da estação no local desejado, a aplicação poderá ajudar o usuário na tomada de decisão.

1.2. JUSTIFICATIVAS

Devido ao processo contínuo de evolução das áreas que envolvem ou interagem com a instrumentação eletrônica, automação e robótica, tais como, microeletrônica, sensores, computadores, telecomunicações e tecnologia da informação e da comunicação, o potencial de inovações ainda é pouco explorado, o que acaba gerando um interesse geral pela ampliação do campo de pesquisas na área.

Dentre as variadas formas de tecnologia existentes, as que mais sofrem pressão para se manterem atualizadas são as voltadas para a mecânica e automação agrícola, tendo em vista que o foco nesta área é, em sua maioria, pelo aumento da produtividade e, para que se chegue neste objetivo, de forma responsável, é necessária uma melhora ou viabilização de processos, tais como: medidas massivas de propriedades físicas e químicas do solo; acompanhamento da produção vegetal e volume de biomassa; planejamento de safra (o que plantar, onde plantar, quais produtos utilizar); medição e acompanhamento de disponibilidade de água; identificação e monitoramento de pragas (doenças, plantas invasoras, insetos), e identificação e monitoramento da saúde e nutrição animal (MASSRUHÁ, LEITE, *et al.*, 2014).

As tecnologias que tem sido adaptadas, desenvolvidas e exploradas na agricultura para atender a essas demandas, seja através do sensoriamento local ou remoto ou para a aplicação, carecem de modelos de referência e padronização, primeiramente, para viabilizar aquisição, comunicação, integração e manipulação de dados identificados ou georreferenciados, e, posteriormente, viabilizar a transformação desses dados em informações que relacionem variáveis ou parâmetros diversos, como, por exemplo, dados de solo, atributos biológicos dos cultivos e parâmetros climatológicos, para suporte à tomada de decisão (MASSRUHÁ, LEITE, *et al.*, 2014).

Espera-se que o protótipo e software proposto contribuam de forma efetiva para um melhor e mais detalhado monitoramento meteorológico, além de possibilitar a emissão de informações e relatórios em tempo real e com mobilidade, de forma que os dados estejam mais bem organizados e distribuídos e, assim, auxiliando a tomada de decisão do usuário.

1.3. MOTIVAÇÃO

Uma das motivações para este trabalho foi a dificuldade do setor Agropecuário em possuir o monitoramento meteorológico detalhado da propriedade com preço acessível, sendo um problema que atinge vários produtores, especificamente os de pequeno porte.

O tema IoT foi escolhido por ser uma área em grande desenvolvimento, com o propósito de contribuir nas pesquisas associadas a Smart Agriculture e monitoramento meteorológico.

1.4. PERSPECTIVAS DE CONTRIBUIÇÃO

Este trabalho poderá contribuir com futuros estudos envolvendo o IoT, relacionado a monitoramento meteorológico, tendo como principal característica seu custo-benefício, pois o mesmo busca apresentar o que se tem de mais atual, sem deixar de ser acessível a qualquer profissional da área. Cumpre ressaltar, ainda, que o referido protótipo possui uma solução simples ao problema apresentado na introdução, de forma que seja uma porta a novas ideias relacionadas ao cotidiano das pessoas e que as ajude na solução dos mais variados desafios enfrentados.

Por fim, insta salientar que o protótipo supramencionado contribui diretamente no dia a dia do usuário da aplicação, em virtude de possuir todas as informações do tempo local e os alertas desta.

1.5. METODOLOGIA DE PESQUISA

O presente trabalho tem como metodologia o estudo e a análise de projetos retirados de artigos científicos, TCC, teses, livros, artigos, revistas, empresas específicas da área, com visão ampla do objeto a ser estudado, tendo como conceito parte da pesquisa descritiva, onde serão coletadas informações e dados sobre o tema, visando uma análise no ambiente natural, para que haja maior concretização das informações identificadas.

Após a coleta dos dados e análises, o trabalho será apresentado como parte da pesquisa explicativa, onde haverá uma exposição dos motivos, causas e problemas relacionados ao tema principal do trabalho. Finalmente, a metodologia terminará com o desenvolvimento prático de software, com documentação, seguindo os padrões de engenharia de software, integrado aos sensores da estação meteorológica e seus módulos, concluindo a parte prática com a elaboração de uma aplicação WebApp que estará conectada à internet e integrada com o sistema principal do trabalho.

1.6. RECURSOS NECESSÁRIOS

Para desenvolver a pesquisa serão necessários os recursos de hardware e software mencionados neste tópico:

- **Hardware Protótipo**
 - Protobord 830 Pontos
 - NodeMcu ESP-12E (Amica)
 - Sensor DHT22 (Temperatura e umidade do ar)
 - Sensor BMP280 (Temperatura, pressão e altitude)
 - Sensor UVM-30A (Luz ultravioleta)
 - Sensor BH1750 (Luminosidade)
 - Sensor de Umidade do Solo Capacitivo

- Sensor Hall US1881/U18 (Pluviômetro de báscula digital)
- Imã de 4x2mm
- Sensor Reed Switch (Anemômetro)
- Lipo-Po Rider v1.3
- Mini painel solar 5v 200ma 1w (107x61x2mm)
- Bateria 3.7v 1200mah
- Resistores 10kΩ
- Jumpers Macho/Femea
- **Software**
 - **Arduino IDE** - Ambiente de desenvolvimento para facilitar o desenvolvimento e a gravação de códigos diretamente no microcontrolador.
 - **Visual Studio Code** - Ambiente de desenvolvimento para a tecnologia JavaScript.

1.7. ESTRUTURA DO TRABALHO

A estrutura deste trabalho é composta das seguintes partes:

- **Capítulo 1 - Introdução:** Neste capítulo é contextualizada a área de estudo. São apresentados, ainda, os objetivos, justificativas, motivação, perspectivas de contribuição e metodologia de pesquisa para o desenvolvimento deste trabalho.
- **Capítulo 2 - Tecnologias e Ferramentas de Desenvolvimento:** Neste capítulo há uma introdução sobre as tecnologias e recursos adotados para o desenvolvimento do WebApp do protótipo.
- **Capítulo 3 - Análise e Especificação do Sistema:** Este capítulo compreende a análise do sistema e a contextualização de suas especificações.
- **Capítulo 4 - Estrutura do Projeto:** Neste capítulo é apresentada a estrutura do projeto, e seu devido orçamento.
- **Capítulo 5 - Proposta de Trabalho:** Neste capítulo é apresentado a aplicação WebApp e o protótipo de IoT para Smart Agriculture com foco em monitoramento meteorológico.
- **Capítulo 6 - Conclusão Parcial:** Neste capítulo são apresentadas as vantagens da implementação da aplicação WebApp e do protótipo de IoT para Smart

Agriculture, com foco em monitoramento meteorológico, e os benefícios gerados à agricultura e às pessoas inseridas nesse meio.

- **Referências**

2. TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO

Para o desenvolvimento deste projeto foram consultados livros, artigos, monografias, *sites* e tutoriais que fornecem informações referentes ao desenvolvimento de sistemas *Web* com *JavaScript*, *React*, *React-Router*, *Redux*, *Webpack*, *Babel*, *node*, *Firebase* e o desenvolvimento do protótipo de IOT para Smart Agriculture com foco em monitoramento meteorológico com o Arduino.

2.1. JAVASCRIPT

A linguagem de programação utilizada é o JavaScript, que é uma linguagem de programação client-side da Web, podendo ser considerada a linguagem de programação mais onipresente da história, pois a ampla maioria dos sites modernos usam JavaScript e os navegadores modernos incluem seus interpretadores. (PRESCOTT, 2016).

Um dos motivos para essa escolha é a comunidade de *JavaScript* que se mostrou ativa o bastante para identificar as limitações que se apresentavam nas suas ferramentas e construir outras que corrigissem, amenizassem ou melhorassem tais aspectos.

Tomando como exemplo a tarefa de construção da interface de usuário, atualmente temos uma grande quantidade de opções de bibliotecas e frameworks para facilitar o desenvolvimento, sendo escolhidas para esse projeto a *React* e a *Redux*.

2.2. REACT

React é uma biblioteca de *JavaScript* voltada para a construção de *UIs*. Cada interface em *React* é composta por um grupo de peças, chamadas “componentes”, com um determinado propósito na página. Componentes podem receber parâmetros - ou, em *React*, “*props*” - que indicam como eles devem ser exibidos e/ou se comportarem, tornando-os altamente adaptáveis. A semelhança da *pattern* de componentes e *props* com a definição de uma “função” traz conceitos poderosos ao desenvolvimento *front-end*, tais como abstração, composição e modularidade (REACT, 2019).

2.3. REDUX

O *Redux* é uma implementação da arquitetura *Flux* que propõe uma solução ao problema de compartilhamento de estados em aplicações web e consiste em criar um fluxo unidirecional de dados que poderá ser consumido por qualquer parte da aplicação (REDUX, 2019).

2.4. WEBPACK

Webpack é uma ferramenta que analisa uma aplicação e, a partir do seu gráfico de dependências, gera uma ou mais *bundles* com os módulos necessários. Esta técnica traz vantagens como a realização de *tree shaking*, também conhecido como “eliminação de código morto”, e a redução do número de requisições que precisam ser feitas ao servidor - ou a otimização da forma como elas acontecem. Por padrão, o *webpack* só lida com arquivos *JavaScript* e *JSON*. É possível, porém, fazer com que ele consiga processar outros tipos de arquivo através de utilitários chamados “*loaders*”. Além de carregar os arquivos e adicioná-los ao gráfico de dependências, o *webpack* também aceita o uso de “*plugins*”, que modificam seus conteúdos, como o *babel* (WEBPACK, 2019).

2.5. BABEL

Babel é um compilador de *JavaScript* capaz de converter códigos com funcionalidades mais recentes de *ECMAScript* em códigos compatíveis com suas versões anteriores. novas versões de *ECMAScript* vêm sido lançadas anualmente, introduzindo características inéditas e, conseqüentemente, expandindo o vocabulário de seus programadores. No entanto, uma parcela considerável de usuários da Internet usa versões de navegadores web cujas *engines* não conseguem dar suporte às últimas *features* de *JavaScript*, justificando a necessidade de ferramentas como o Babel (BABEL, 2019).

2.6. NODE

Node.js, ou simplesmente “Node”, é um *runtime* de *JavaScript* orientado a eventos assíncronos. A programação orientada a eventos oferece uma alternativa mais eficiente e escalável a modelos que usam *multithreading*, e a possibilidade de realizar operações de I/O de forma assíncrona previne que a aplicação fique bloqueada durante elas. O Node permitiu que pessoas já familiarizadas com *JavaScript*, como muitos desenvolvedores *front-end*, possam construir aplicações *server-side* sem a necessidade de aprender uma nova linguagem. Além de programas web, conseguir executar código fora do browser também tornou *JavaScript* uma opção viável para *scripting* na linha de comando (TILKOV e VINOSKI, 2010).

2.7. FIREBASE

O Firebase é um Baas (Backend as a Service) para aplicações Web e Mobile do Google, foi lançado em 2004 e com o passar dos anos cresceu muito, se tornando uma ferramenta que hoje para alguns projetos é a melhor opção, devido a quantidade de serviços oferecidos por ele, além da facilidade de implementação (FIREBASE, 2019).

A criação de uma aplicação, seja ela Web ou Mobile, não é fácil, e é preciso se preocupar com diversos fatores, dentre eles:

- Funcionamento em variados tipos de dispositivos;
- Infraestrutura;
- Atualizações que não gerem mais erros que solucionam;
- Segurança na transferência de dados;
- Mecanismos de Autenticação.

Foi pensando em uma maneira de solucionar todos esses problemas que surgiu a motivação para a utilização do Firebase.

O Banco de Dados em Tempo Real permite o armazenamento e sincronismo dos dados entre usuários e dispositivos em tempo real com um banco de dados NoSQL hospedado na nuvem. Os dados atualizados são sincronizados em todos os dispositivos conectados

em segundos. Além disso, seus dados permanecem disponíveis caso seu aplicativo fique off-line, o que oferece uma ótima experiência do usuário, independentemente da conectividade de rede (FIREBASE, 2019).

2.8. PLACA DE DESENVOLVIMENTO E SENSORES

Foi utilizada, um NodeMcu ESP-12E, modelo Amica, para realizar a prototipagem do projeto, graças a sua experiência de usuário simples e acessível, sendo uma ferramenta fácil para prototipagem rápida que é amplamente utilizada em diferentes projetos e aplicativos. Ademais, o software utilizado é o IDE Arduino, que é descomplicado para iniciantes, mas flexível o suficiente para usuários avançados.

2.8.1. NodeMCU

A Internet das Coisas (IoT) tem sido um campo de tendências no mundo da tecnologia, o que contribuiu para modificar a maneira como trabalhamos. Objetos físicos e o mundo digital estão conectados, tendo isso em mente, a Espressif Systems, uma empresa de semicondutores de Xangai, lançou um microcontrolador habilitado para WiFi - ESP8266, de baixo custo, perfeito para praticamente qualquer projeto de IoT (ENGINEERS, 2020).

A placa de desenvolvimento equipa o módulo ESP-12E que contém o chip ESP8266, com o microprocessador Tensilica Xtensa® LX106 RISC de 32 bits, que opera com uma frequência de clock ajustável de 80 a 160 MHz e suporta RTOS como ilustra a figura 1.

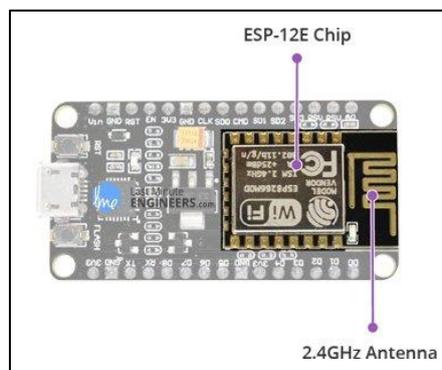


Figura 1: Módulo ESP-12E.
Fonte: Last Minute Engineers, 2020.

Há também 128 KB de RAM e 4 MB de memória Flash (para armazenamento de programas e dados), apenas o suficiente para lidar com as grandes cadeias de caracteres que compõem as páginas da Web, dados JSON/XML e tudo o que lançamos nos dispositivos IoT atualmente.

O ESP8266 integra o transceptor Wi-Fi 802.11b/g/n HT40, para que ele possa não apenas se conectar a uma rede WiFi e interagir com a Internet, mas também configurar uma rede própria, permitindo que outros dispositivos se conectem diretamente, o que torna o ESP8266 NodeMCU ainda mais versátil.

Como a faixa de tensão operacional do ESP8266 é de 3V a 3,6V, a placa vem com um regulador de tensão LDO para manter a tensão constante em 3,3V. Ele pode fornecer, de forma confiável, até 600mA, o que deve ser mais do que suficiente, tendo em vista que o ESP8266 puxa até 80mA durante as transmissões de RF e 20µA durante o modo de suspensão. A saída do regulador também é dividida em um dos lados da placa e rotulada como 3V3. Este pino pode ser utilizado para fornecer energia a componentes externos como ilustra a figura 2.

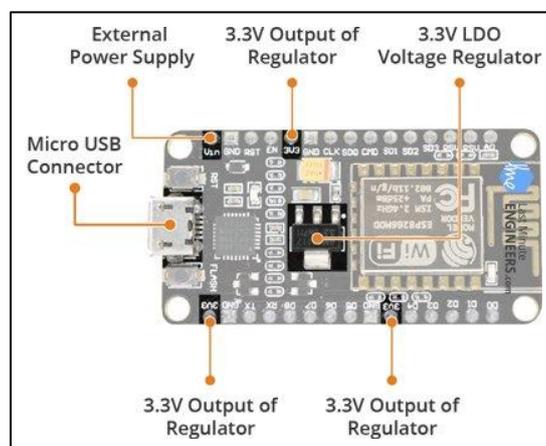


Figura 2: Requisitos de energia.
Fonte: Last Minute Engineers, 2020.

A energia do ESP8266 NodeMCU é fornecida pelo conector USB MicroB integrado. Como alternativa, pode-se usar uma fonte de tensão de 5V regulada, o pino VIN pode ser utilizado para fornecer energia, diretamente, ao ESP8266 e seus periféricos.

O ESP8266 NodeMCU possui um total de 17 pinos GPIO divididos nos cabeçalhos dos dois lados da placa de desenvolvimento, como ilustra a figura 3. Esses pinos podem ser atribuídos a todos os tipos de tarefas periféricas, incluindo:

- Canal ADC - um canal ADC de 10 bits.
- Interface UART - a interface UART é usada para carregar o código em série.
- Saídas PWM - pinos PWM para escurecer os LEDs ou controlar os motores.
- Interface SPI, I2C e I2S - Interface SPI e I2C para conectar todos os tipos de sensores e periféricos.
- Interface I2S - interface I2S caso seja necessário adicionar som ao seu projeto.

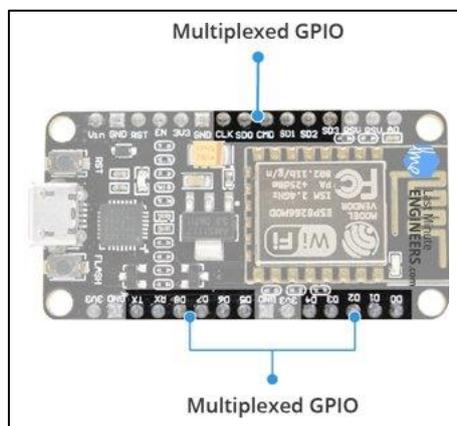


Figura 3: Periféricos e E/S.
Fonte: Last Minute Engineers, 2020.

Graças ao recurso de multiplexação de pinos do ESP8266 (vários periféricos multiplexados em um único pino GPIO) é possível que um único pino GPIO atue como PWM/UART/SPI.

O ESP8266 NodeMCU possui dois botões, como ilustra a figura 4. Um marcado como RST, localizado no canto superior esquerdo, é o botão utilizado para redefinir o chip ESP8266. Já o botão FLASH, localizado no canto inferior esquerdo, é responsável pelo download durante a atualização do firmware.

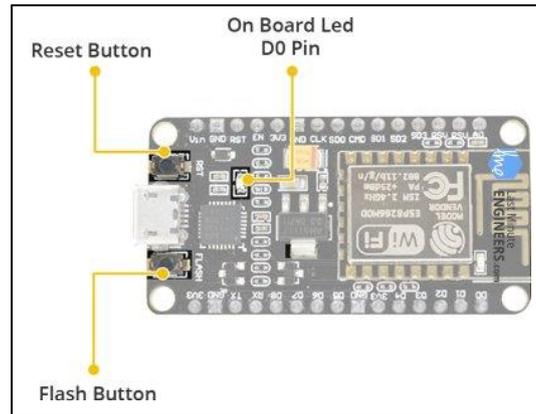


Figura 4: Interruptores e indicadores.
Fonte: Last Minute Engineers, 2020.

A placa também possui um indicador LED programável pelo usuário e conectado ao pino D0 da placa.

A placa inclui o controlador de ponte USB para UART CP2102 da Silicon Labs, como ilustra a figura 5, que converte o sinal USB em serial e permite programar pelo computador, além de possibilitar a comunicação com o chip ESP8266.

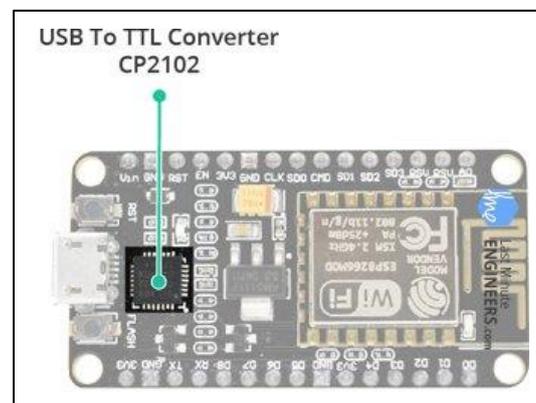


Figura 5: Comunicação serial.
Fonte: Last Minute Engineers, 2020.

O ESP8266 NodeMCU possui um total de 30 pinos que fazem a interface com o mundo exterior. Na figura 6 são apresentadas todas as conexões:

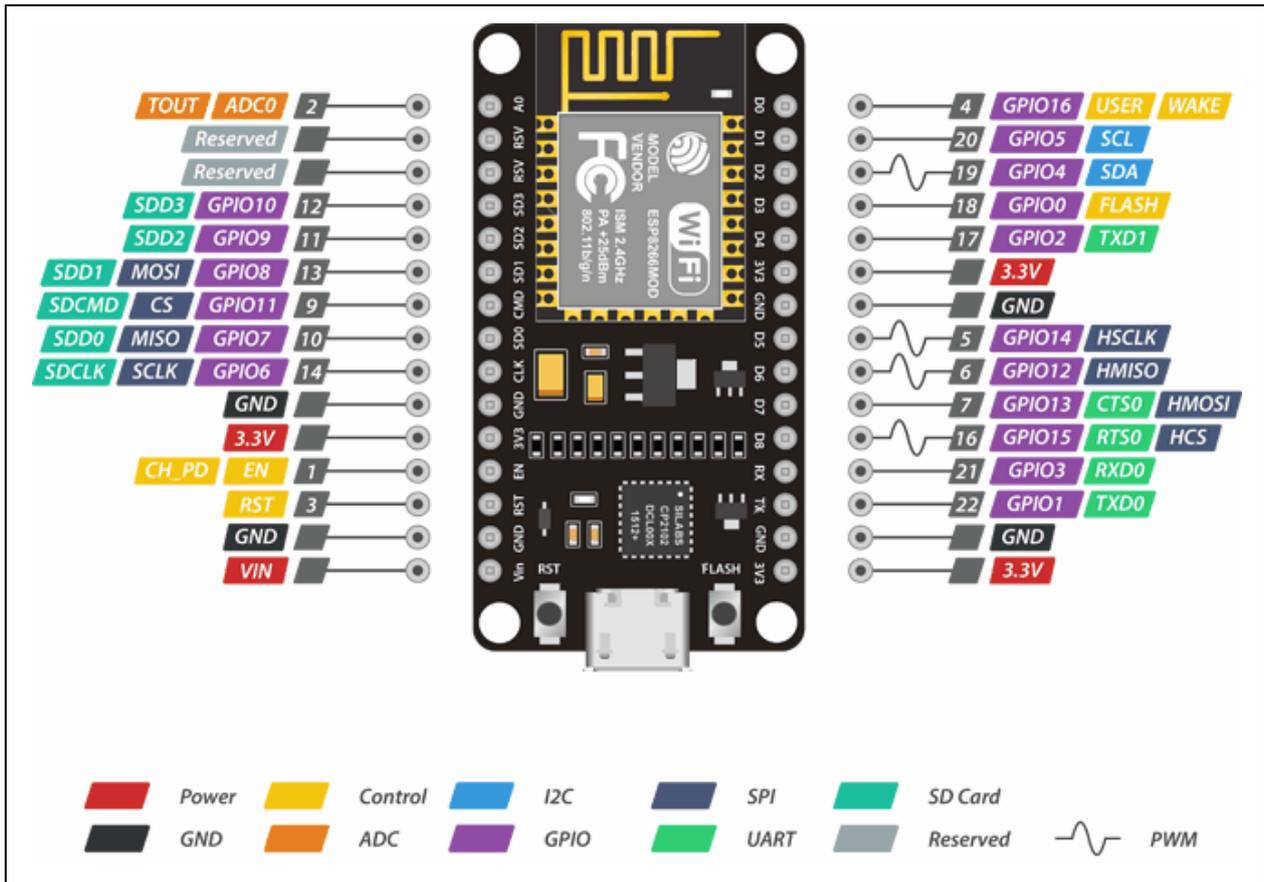


Figura 6: Pinagem ESP8266 NodeMCU.
Fonte: Last Minute Engineers, 2020.

Por uma questão de simplicidade, serão criados grupos de pinos com funcionalidades semelhantes.

Existem quatro pinos de força, quais sejam um pino VIN e três pinos de 3.3V. O pino VIN pode ser utilizado para fornecer energia, diretamente, ao ESP8266 e seus periféricos, caso exista uma fonte de tensão de 5V regulada. Os pinos de 3,3V são a saída de um regulador de tensão on-board. Esses pinos podem ser usados para fornecer energia a componentes externos.

O GND é um pino de aterramento da placa de desenvolvimento ESP8266 NodeMCU.

Os pinos I2C são usados para conectar todos os tipos de sensores e periféricos I2C em seu projeto. O I2C Master e o I2C Slave são suportados. A funcionalidade da interface I2C pode ser realizada de forma programática e a frequência do clock é de 100kHz, no máximo. É preciso se notar que a frequência do clock I2C deve ser maior que a frequência do clock mais lenta do dispositivo escravo.

Pinos GPIO ESP8266 - O NodeMCU possui 17 pinos GPIO que podem ser atribuídos a várias funções, como I2C, I2S, UART, PWM, controle remoto IR, luz LED e botão programaticamente. Cada GPIO habilitado para digital pode ser configurado para pull-up ou pull-down, interno ou definido para alta impedância. Quando configurado como uma entrada, também pode ser definido como trigger de borda ou trigger de nível para gerar interrupções na CPU.

Canal ADC - O NodeMCU é incorporado com um SAR ADC de precisão de 10 bits. As duas funções podem ser executadas utilizando o ADC viz, quais sejam o teste da tensão de alimentação do pino VDD3P3 e o teste da tensão de entrada do pino TOUT. No entanto, eles não podem ser implementados ao mesmo tempo.

Pinos UART ESP8266 - O NodeMCU possui 2 interfaces UART, ou seja, UART0 e UART1, que fornecem comunicação assíncrona (RS232 e RS485) e podem se comunicar em até 4,5Mbps. O UART0 (pinos TXD0, RXD0, RST0 e CTS0) pode ser utilizado para comunicação, já que suporta controle de fluídos. Já o UART1 (pino TXD1) apresenta apenas o sinal de transmissão de dados e, geralmente, tem como função a impressão de log.

Pinos SPI - O ESP8266 possui dois SPIs (SPI e HSPI) nos modos escravo e mestre. Essas SPIs também oferecem suporte aos seguintes recursos de SPI de uso geral:

- 4 modos de temporização da transferência do formato SPI
- Até 80MHz e os clock divididos de 80MHz
- FIFO de até 64 bytes

Os pinos SDIO ESP8266 possuem Interface de entrada/saída digital segura (SDIO), que é utilizada para interface direta com cartões SD. SDIO v1.1 de 25MHz e 4 bits e SDIO v2.0 de 50MHz e 4 bits são suportados.

Pinos PWM - A placa possui 4 canais de modulação por largura de pulso (PWM). A saída PWM pode ser implementada programaticamente e utilizada para acionar motores digitais e LEDs. A faixa de frequência PWM é ajustável de 1000µs a 10000µs, ou seja, entre 100Hz e 1kHz.

Os pinos de controle são usados para controlar o ESP8266. Esses pinos incluem o pino de ativação por chip (EN), o pino de redefinição (RST) e o pino WAKE.

- Pino EN - O chip ESP8266 é ativado quando o pino EN é puxado para ALTO. Quando puxado para BAIXO, o chip funciona com potência mínima.
- Pino RST - O pino RST é usado para redefinir o chip ESP8266.
- Pino WAKE - O pino Wake é usado para despertar o chip do sono profundo.

2.8.2. DHT22

O DHT22, ilustrado na figura 7, é um sensor de temperatura e umidade comumente utilizado por ser pequeno, barato e pelo baixo consumo. O sensor possui um Termistor NTC 10K dedicado para medir a temperatura e outro sensor dedicado para medir a umidade, além de um microcontrolador de 8 bits para transmitir os valores. O sensor também é calibrado de fábrica, portanto, fácil de fazer interface com outros microcontroladores.

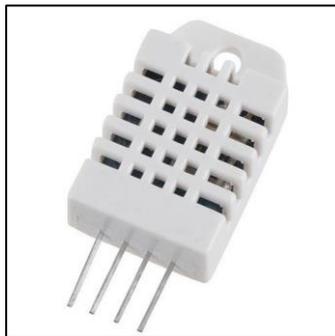


Figura 7: Sensor DHT22.
Fonte: Própria, 2020.

O sensor pode medir temperaturas de -40°C a 80°C e umidades de 0% a 100% com uma precisão de $\pm 1^{\circ}\text{C}$ e $\pm 1\%$ e tempo de resposta inferior a 5 segundos. A alimentação precisa ser de 3,3v a 5v, com consumo máximo de corrente de 0,5mA (LIU, 2020).

2.8.3. BMP280

O Sensor Barométrico BMP280 é um barômetro digital de alta precisão e baixa potência, como ilustrado na figura 8. Pode ser usado para medir a temperatura e a pressão atmosférica com precisão. Pode ser ligado a um microcontrolador com I2C ou SPI. O

sensor é compatível com Arduino, Pic, Raspberry Pi, Arm entre outros. É ideal para utilização em projetos de drones, estações meteorológicas entre outros (SENSORTEC, 2020).

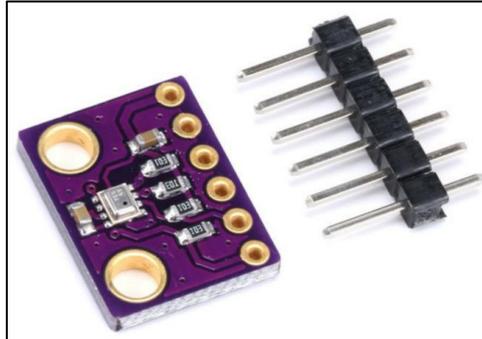


Figura 8: Sensor BMP280.
Fonte: Própria, 2020.

O sensor pode medir temperaturas de -40°C a 85°C e pressões de 300hPa a 1100hPa com uma precisão de $\pm 1^{\circ}\text{C}$ e $\pm 0.12\text{hPa}$ (equiv. $\pm 1\text{m}$); e tempo de resposta inferior a 5 segundos. A alimentação precisa ser de 1,7v a 3,6v com consumo máximo de corrente de $2.7\mu\text{A}$ (SENSORTEC, 2020).

2.8.4. UVM-30A

O Sensor de Raios Ultravioleta UVM-30^a, ilustrado na figura 9, é baseado no circuito integrado UVM-30A que permite a detecção de raios ultravioletas. Com este sensor se torna simples a função de monitoramento do índice UV, uma vez possui diversas aplicações. Seu uso se dá em sistemas de análise de crescimento das plantas, em estações meteorológicas, entre outros.

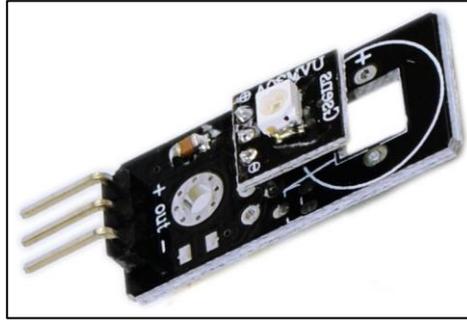


Figura 9: Sensor UVM-30A.
Fonte: Própria, 2020.

O Sensor é capaz de detectar raios ultravioleta de comprimentos de onda entre 200nm a 370nm. Funciona com uma precisão de $\pm 1\text{UV}$ e tempo de resposta inferior a 5 segundos. A alimentação precisa ser de 3v a 5v com consumo máximo de corrente de 0,1mA, sendo ideal para utilização com microcontroladores de uso geral (HEK, 2020).

2.8.5. BH1750

O sensor BH1750, ilustrado na figura 10, é um sensor de luminosidade com interface I2C, utilizado para detectar a incidência de luz. Permite realizar medições entre 1lux a 65.535lux. Possui um conversor AD de 16 bits e funciona com tensões entre 3v a 5v. É muito utilizado em sistemas que detectam a luminosidade para um ajuste de eficaz da iluminação, por exemplo. (COELNER, CLAWS, *et al.*, 2020)

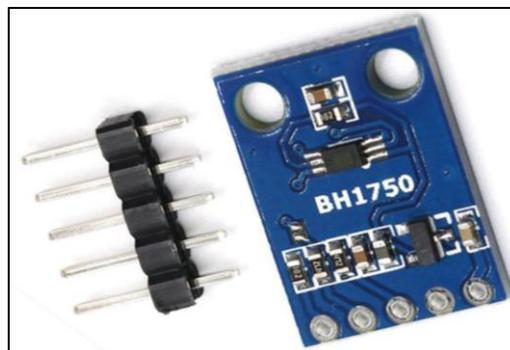


Figura 10: Sensor BH1750.
Fonte: Própria, 2020.

2.8.6. Umidade do Solo Capacitivo

O Sensor de Umidade em questão mede o nível de umidade do solo por meio de um sensor capacitivo, ao invés de fazer uso de um sensor resistivo como outros sensores no mercado, e, devido a isso, apresenta maior durabilidade e precisão. Em razão de suas vantagens, o Sensor de Umidade do Solo (Capacitivo), ilustrado na figura 11, é uma excelente escolha para quem quer realizar projetos como, estufa automatizada, irrigação automática de plantas e controle de seca (CIRCUITO, 2020).

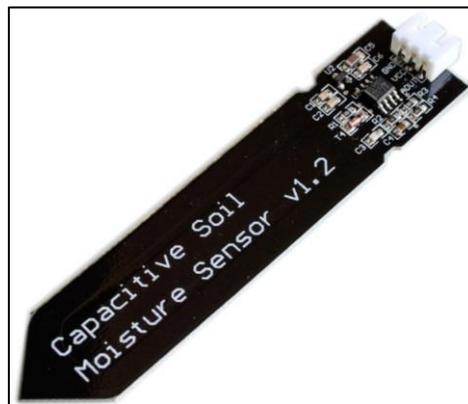


Figura 11: Sensor de Umidade do Solo Capacitivo.
Fonte: Própria, 2020.

O Sensor é fabricado com um material resistente à corrosão, o que lhe confere excelente vida útil. Este módulo inclui um regulador de tensão que fornece uma faixa de tensão operacional de 3,3V a 5.5V. É perfeito para plataformas de baixa tensão e possui consumo máximo de corrente de 0,1mA (CIRCUITO, 2020).

2.8.7. Hall US1881/U18

O sensor Hall US1881/U18 é um componente eletrônico de estado sólido que está se tornando cada vez mais popular, pois pode ser usado em vários tipos de aplicações, tais como detecção de posição, velocidade ou movimento direcional. Para reconhecer campos magnéticos, é amplamente utilizado para controlar, por exemplo, um certo número de voltas que uma roda executará ou as RPM de um motor enquanto estiver em

funcionamento. Além de reconhecer o campo magnético, O sensor Hall US1881, ilustrado na figura 12, pode identificar a frequência eletromagnética e medir sua intensidade. É comumente usado na indústria e na eletrônica para monitorar equipamentos sem contato físico, uma vez que não se desgasta por não entrar em contato com o equipamento monitorado, apresentando uma excelente relação custo/benefício (TECHNOLOGIES, 2020).



Figura 12: Sensor Hall US1881/U18.
Fonte: Própria, 2020.

Ao detectar um campo magnético, o detector do sensor Hall - U18 altera a saída digital para baixo nível lógico (LOW) e a mantém. Ao detectar o campo magnético novamente, a saída digital assume um alto nível lógico (HIGH). A alimentação precisa ser de 2.5v a 20v (TECHNOLOGIES, 2020).

2.8.8. Reed Switch

O sensor magnético Reed Switch, ilustrado na figura 13, é um componente amplamente utilizado em várias áreas. Sua operação é baseada na abertura ou fechamento de seus contatos, dependendo da ação que um campo magnético externo exerce sobre ele, ou seja, funciona como um interruptor (ELECTRONICS, 2020).



Figura 13: Sensor Reed Switch.
Fonte: Própria, 2020.

O sensor consiste em uma lâmpada de vidro interna e duas lâminas flexíveis sensíveis à ação de um campo magnético; em condições normais, as lâminas são separadas e não conduzem corrente elétrica, operando desta maneira, como um interruptor aberto, já se aproximando de um gerador de campo magnético como um ímã, o campo magnético magnetiza as pás e gera a atração entre elas, o que causa o fechamento dos contatos e a consequente condução da corrente. Exemplos de aplicações incluem: alarme, sensor de presença ou passagem de um determinado objeto, entre outros (ELECTRONICS, 2020).

2.8.9. Lipo-Po Rider

O Lipo Rider, ilustrado na figura 14, é um módulo de gerenciamento de carga para baterias Li-Ion e Li-Po que permite o gerenciamento de energia produzida por meio de painéis solares compatíveis, transportando essa tensão de maneira controlada para a bateria. Produzido para auxiliar na recarga de baterias por meio de uma fonte de energia renovável e ecológica, o carregador de baterias Li-Ion e Li-Po pode ser usado para recarregar equipamentos como telefones celulares e servir como fonte de energia para o Arduino, por exemplo.

Possui um duplo tipo de carregamento, pois, além de permitir a alimentação por meio de um painel solar com tensão entre 4,8V e 6,5V DC, possibilita alimentar o sistema por meio de uma fonte com um cabo micro USB. A conexão do painel solar e a bateria é feita por meio de um conector jst 2.0, onde a tensão liberada para carregamento é regulada em 4,2V com uma corrente de 200mA, permitindo a conexão de baterias de tamanhos diferentes (incluindo baterias com tensão de 3,7V). É importante lembrar que quanto

maior a bateria, maior o tempo de carregamento, e, ainda, que é possível usá-lo durante este período.

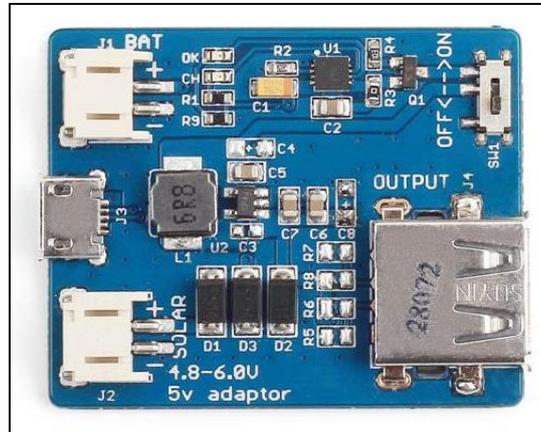


Figura 14: Lipo-Po Rider.
Fonte: Própria, 2020.

O Lipo Rider também possui um conector USB fêmea para alimentar dispositivos externos com tensão de 5V e corrente de 100mA, com um interruptor ON/OFF para permitir esta função. Ele também possui um LED indicador de carga ("Carga"), que permanece aceso sempre que a bateria está sendo carregada, desligando quando a carga está no nível máximo.

3. ESTRUTURA DO PROJETO

Neste capítulo é apresentada a estrutura do projeto adotada para o desenvolvimento do trabalho de conclusão de curso. A implantação será dividida em etapas e para defini-las será utilizado o diagrama de Estrutura Analítica de Projetos (EAP), do inglês Work Breakdown Structure (WBS), que é definido como o processo de subdivisão hierárquica do trabalho em componentes, o que proporciona um gerenciamento mais descomplicado, tendo como objetivo principal identificar elementos terminais, tais como produtos, serviços e resultados a serem obtidos em um projeto. (RUGGIERI, 2019).

3.1. ESTRUTURA ANALÍTICA DE TRABALHO

O principal objetivo da EAP é organizar a estrutura analítica do projeto, dividindo o trabalho em pequenas partes com um alto nível de detalhes deste, a fim de facilitar o entendimento.

As etapas do desenvolvimento da aplicação são ilustradas na Figura 23, por meio do diagrama EAP, que será utilizado para organizar e orientar o trabalho a ser desenvolvido. As etapas supramencionadas serão três, quais sejam os requisitos, que consiste no levantamento, análise, validação e especificação destes, a análise, que contempla a definição dos objetivos e a elaboração dos diagramas e, por fim, o desenvolvimento, aonde será realizada a montagem dos componentes, a programação e o teste dos mesmos.

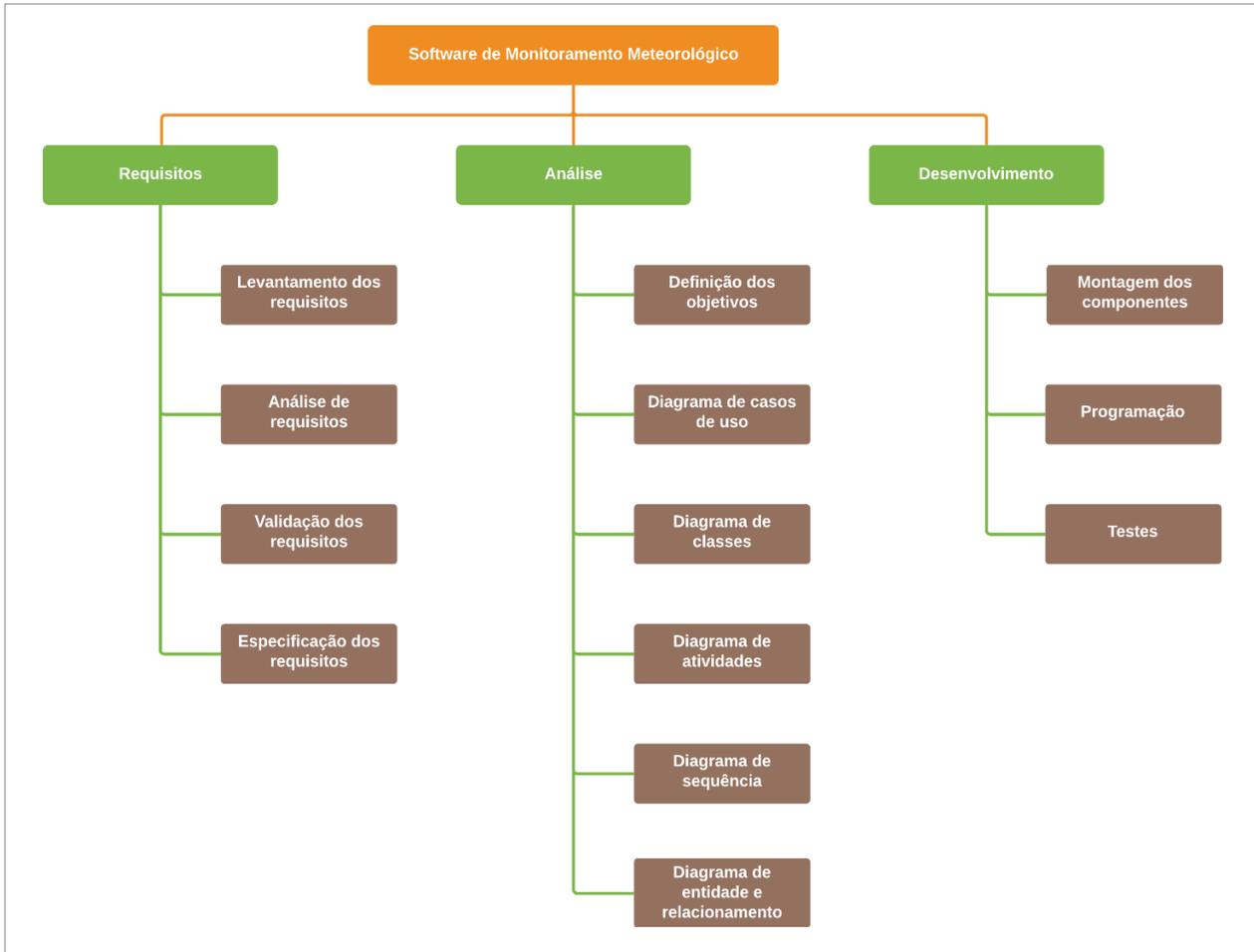


Figura 15: Estrutura Analítica do Projeto.
Fonte: Própria, 2020.

3.2. SEQUENCIAMENTO DAS ATIVIDADES

O diagrama de sequenciamento de atividades ilustra o tempo de duração para a realização de cada atividade do projeto em desenvolvimento, tendo como objetivo definir de forma lógica a execução das tarefas, como ilustra a figura 24.

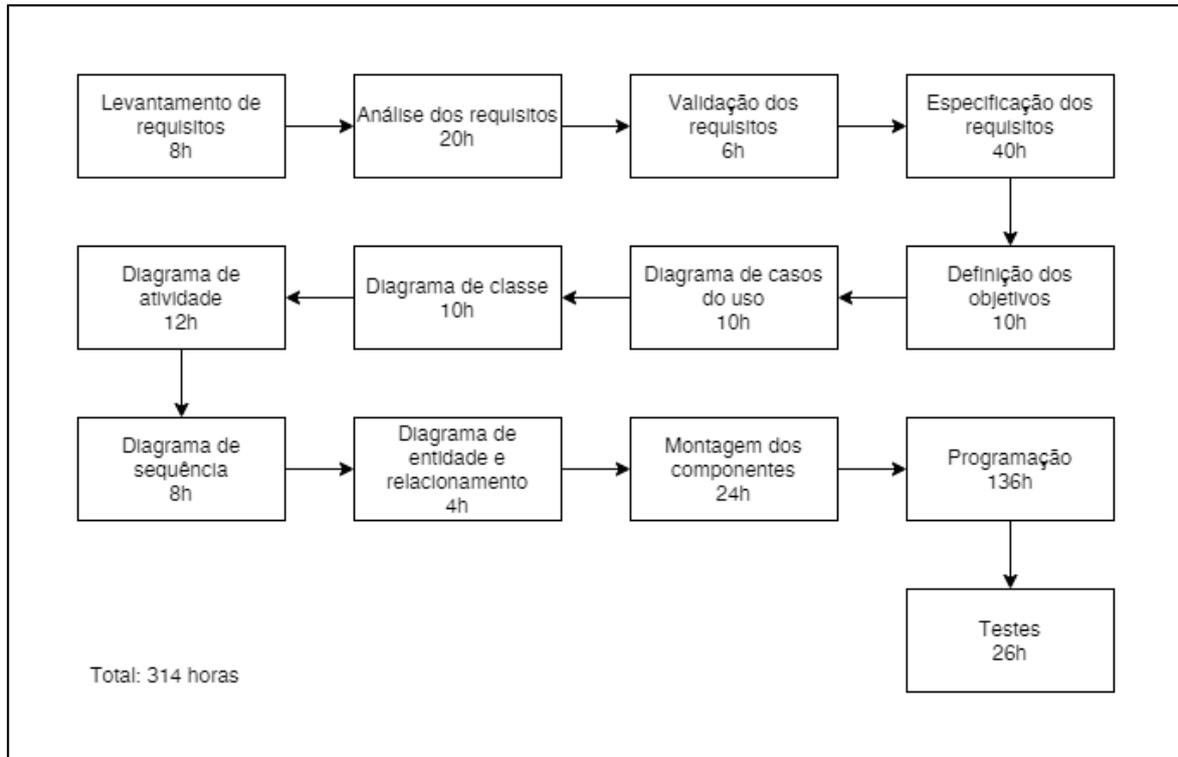


Figura 16: Sequenciamento das atividades.

Fonte: Própria, 2020.

O diagrama apresentado ilustra o tempo necessário para conclusão de todas as atividades do trabalho de conclusão de curso.

3.3. ORÇAMENTO

Custo Análise e Custo Programação: Primeiro multiplicamos a quantidade de horas trabalhadas pelo valor da hora, gerando assim o subtotal, como ilustra a tabela 15.

	Horas de trabalho	Valor/hora	Subtotal
Custo Análise:	128	R\$ 20,00	R\$ 2.560,00
Custo Programação:	186	R\$ 15,00	R\$ 2.790,00

Tabela 1: Custo análise e custo programação.

Custo Linguagem e Custo Banco de dados (B.D): A linguagem utilizada para a programação será o React. O seu valor de licença é gratuita. O B.D utilizado será o

Firestore da Google. Para esse projeto, também utilizaremos a licença gratuita, como ilustra a tabela 16.

	Valor da licença	Valor por dia	Subtotal
Custo Linguagem:	R\$ 0,00	R\$ 0,00	R\$ 0,00
Custo B.D:	R\$ 0,00	R\$ 0,00	R\$ 0,00

Tabela 2: Custo linguagem e custo banco de dados.

Custo de Computadores: Primeiramente, para calcular o custo do computador divide-se o valor do computador pelo tempo de projeto, neste caso 1 ano (12 meses), depois divide-se novamente pela quantidade de horas trabalhadas no projeto durante o mês, neste caso 160 horas, após isso multiplica-se pela quantidade de computadores que forem necessários para o projeto, neste caso 1 (um), chegando, assim, ao valor por dia. Por fim, multiplica-se pela quantidade de horas de projeto, neste caso 314 horas, para se obter o total.

Aluguel, Água, luz e Internet: Primeiramente, para calcular o custo de Aluguel, Água, luz e Internet, divide-se o valor pela quantidade de horas trabalhadas no projeto no decorrer do mês, neste caso 160 horas, chegando assim ao valor por dia. Por fim multiplica-se pela quantidade de horas de projeto, neste caso 314 horas, para gerar o total, como ilustra a tabela 17.

	Valor	Valor por dia	Subtotal
Custo Computador:	R\$ 3.500,00	R\$ 1,83	R\$ 572,39
Aluguel:	R\$ 600,00	R\$ 3,75	R\$ 1.177,50
Água, luz, Internet:	R\$ 350,00	R\$ 2,19	R\$ 686,87

Tabela 3: Custo de computadores, aluguel, água, luz e Internet.

Custo do protótipo IoT: Primeiramente, para calcular o custo do protótipo IoT, divide-se o valor do protótipo IoT pelo tempo de projeto, neste caso 1 ano (12 meses), depois divide-se novamente pela quantidade de horas trabalhadas no projeto no mês, neste caso 160 horas, após isso multiplica-se pela quantidade de protótipos IoTs que se fizerem necessários para o projeto, neste caso 1, chegando assim ao valor por dia. Por fim,

multiplica-se pela quantidade de horas de projeto, neste caso 314 horas, para gerar o total, como ilustra a tabela 18.

	Valor	Valor por dia	Subtotal
Custo NodeMcu ESP-12E (Amica):	R\$ 19,90	R\$ 0,01	R\$ 3,25
Custo sensor DHT22:	R\$ 19,90	R\$ 0,01	R\$ 3,25
Custo sensor BMP280:	R\$ 9,90	R\$ 0,01	R\$ 1,62
Custo sensor UVM-30A:	R\$ 62,78	R\$ 0,03	R\$ 10,27
Custo sensor BH1750:	R\$ 8,30	R\$ 0,00	R\$ 1,36
Custo sensor Umidade do Solo Capacitivo:	R\$ 14,58	R\$ 0,01	R\$ 2,38
Custo sensor Hall US1881/U18:	R\$ 3,99	R\$ 0,00	R\$ 0,65
Custo pluviômetro em acrílico:	R\$ 45,00	R\$ 0,02	R\$ 7,36
Custo balança impressa em ABS:	R\$ 20,00	R\$ 0,01	R\$ 3,27
Custo ímã de 4x2mm:	R\$ 2,99	R\$ 0,00	R\$ 0,49
Custo sensor Reed Switch:	R\$ 1,34	R\$ 0,00	R\$ 0,22
Custo estrutura anemômetro	R\$ 97,00	R\$ 0,05	R\$ 15,86
Custo Lipo-Po Rider:	R\$ 199,03	R\$ 0,10	R\$ 32,55
Custo Protoboard 830 Pontos:	R\$ 12,86	R\$ 0,01	R\$ 2,10
Custo Mini painel solar 5v 200ma 1w:	R\$ 21,86	R\$ 0,01	R\$ 3,58
Custo Bateria 3.7v 1200mah:	R\$ 44,99	R\$ 0,02	R\$ 7,36
Resistores 10kΩ:	R\$ 1,00	R\$ 0,00	R\$ 0,16
Jumpers Macho/Femea:	R\$ 10,00	R\$ 0,01	R\$ 1,64

Tabela 4: Custo do protótipo IoT.

Total do custo: Após realizar todos os cálculos do projeto, foram somados os totais, resultando no custo do software, conforme ilustrado na tabela 5.

Total do custo:	R\$ 7.884,14
-----------------	--------------

Tabela 5: Custo total.

4. ANÁLISE E ESPECIFICAÇÃO DO SISTEMA

A aplicação é modelada utilizando-se a *Unified Modeling Language* (UML), com o uso da ferramenta *draw.io* para a elaboração de diversos diagramas tais como o diagrama de classes, de componentes, de atividades, de sequências, de casos de uso e de entidade e relacionamento.

4.1. MAPA MENTAL

Para melhor abstração da aplicação proposta, fora desenvolvido um Mapa Mental ou *Mind Map*, que consiste em um diagrama responsável por facilitar o entendimento de uma ideia lógica, assim como organizar tópicos relacionados a partir de um determinado foco inicial (SOUZA, 2004). O conceito de mapa mental foi desenvolvido por Tony Buzan na década de 70 e seu uso, neste projeto de conclusão de curso, contribui para a definição das funcionalidades da aplicação a serem desenvolvidas.

Na Figura 15 são apresentadas as atividades da aplicação proposta:

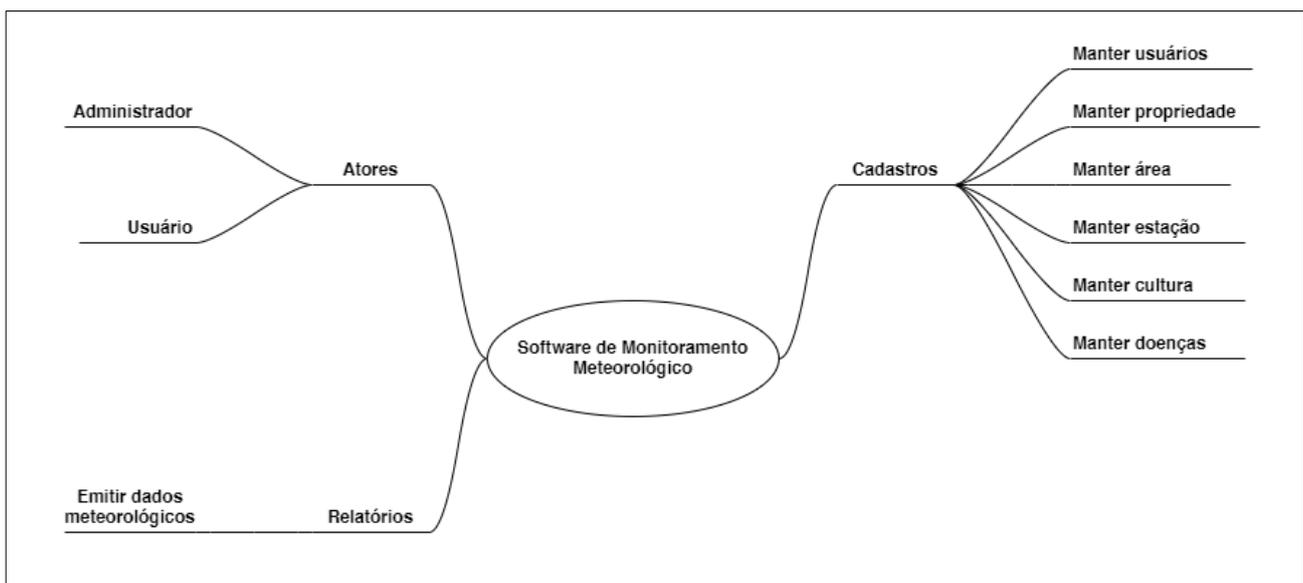


Figura 17: Mapa Mental.
Fonte: Própria, 2020.

4.2. LISTA DE REQUISITOS

O *software* proposto neste trabalho possuirá diversas funcionalidades, tais como:

- Realizar login;
- Manter usuário;
- Manter propriedade;
- Manter área;
- Manter estação;
- Manter cultura;
- Manter doenças;
- Emitir relatório dados meteorológicos;

4.3. DIAGRAMA E ESPECIFICAÇÃO DE CASOS DE USO

Para melhor visualização dos requisitos da aplicação foram elaborados alguns diagramas. Como elemento primário da UML, utiliza-se o Diagrama de Casos de Uso para especificar o comportamento de um sistema ou de parte de um sistema e é uma descrição de um conjunto de sequências de ações (BOOCH, RUMBAUGH e JACOBSON, 2006).

Na Figura 16 é apresentado o Diagrama de Caso de Uso para as entidades “Administrador” e “Usuário”:

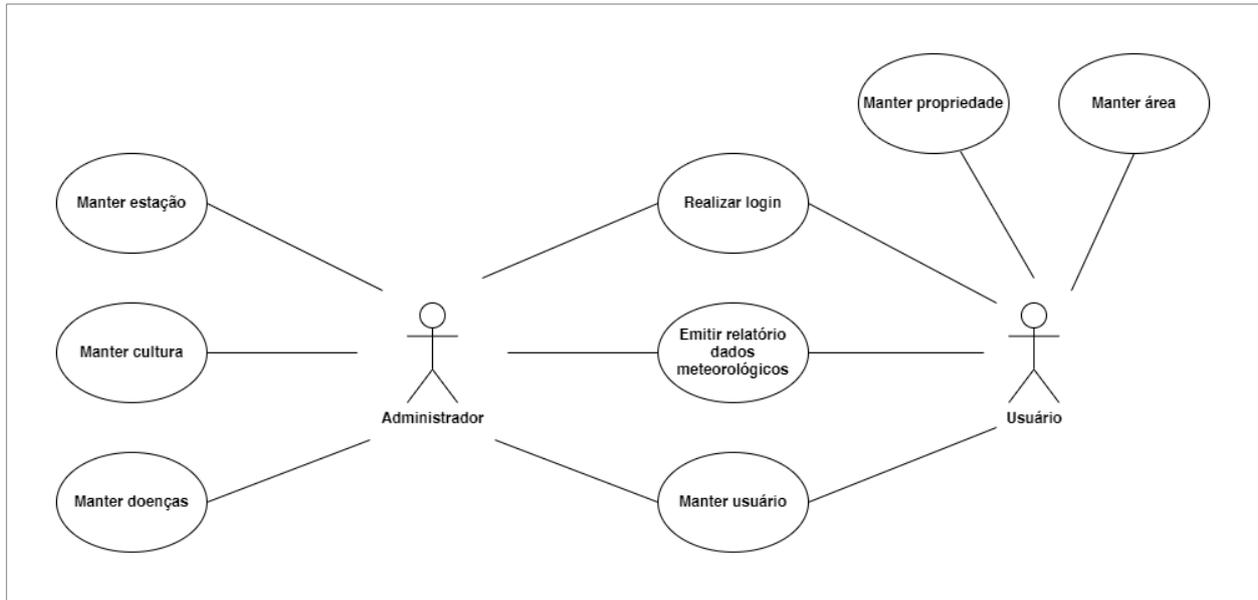


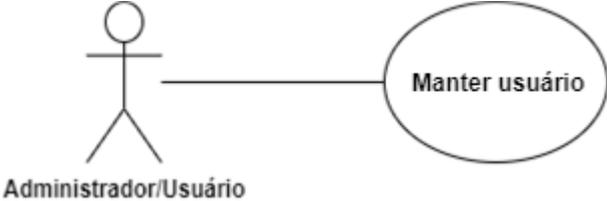
Figura 18: Diagrama de Caso De Uso do Administrador e Usuário.
Fonte: Própria, 2020.

4.3.1. UC 01 - Realizar login

1 - Finalidade/Objetivo	Realizar acesso a plataforma.
2 - Atores	Administrador/Usuário.
3 - Evento inicial	Informar login e senha de acesso.
4 - Fluxo Principal	a) O sistema apresenta a interface; b) O sistema apresenta os tipos de usuário; c) O Ator seleciona o tipo; d) O Ator realiza o login; (A1)
5 - Fluxo alternativo	A1 - O Ator ainda não possui o login; a) O ator seleciona a opção criar conta; b) O Sistema apresenta o formulário; c) O ator preenche o formulário; d) O Ator confirma a operação;

Tabela 6: UC 01 - Realizar login.

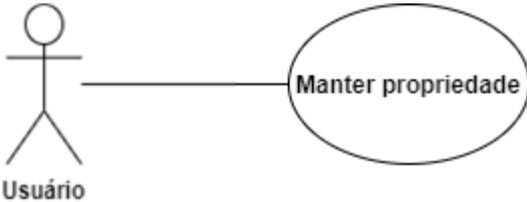
4.3.2. UC 02 - Manter usuário

 <p style="text-align: center;">Administrador/Usuário</p>	
1 - Finalidade/Objetivo	Realizar o cadastro, alteração, inativação e consulta dos usuários.
2 - Atores	Administrador/Usuário.
3 - Pré-Condições	Estar logado.
4 - Evento inicial	O ator começa o caso de uso selecionando a opção de manter usuário.
5 - Fluxo Principal	<p>a) O sistema apresenta a interface;</p> <p>b) O sistema mostra as opções de manter usuário;</p> <p>(A1) (A2) (A3) (A4)</p> <p>c) O caso de uso é finalizado.</p>
6 - Fluxo Alternativo	<p>A1 - Pesquisar usuário</p> <p>a) O ator clica em pesquisar usuário;</p> <p>b) O ator preenche os campos desejados para pesquisa;</p> <p>c) O ator seleciona a opção de pesquisar;</p> <p>d) O sistema exibe o resultado da pesquisa;</p> <p>e) O caso de uso volta para o passo “c” do fluxo principal.</p> <p>A2 - Cadastrar usuário</p> <p>a) O ator clica em cadastrar usuário;</p> <p>b) Todos os campos da tela são habilitados;</p> <p>c) O ator preenche os campos;</p> <p>d) O sistema valida os campos preenchidos;</p> <p>e) O sistema habilita o botão de confirmar;</p> <p>f) O ator confirma o cadastro;</p> <p>g) O caso de uso volta para o passo “c” do fluxo</p>

	<p>principal.</p> <p>A3 - Alterar usuário</p> <p>a) O ator clica em alterar usuário;</p> <p>b) Todos os campos da tela são habilitados;</p> <p>c) O ator altera os campos desejados;</p> <p>d) O sistema valida os campos preenchidos;</p> <p>e) O sistema habilita o botão de confirmar;</p> <p>g) O ator confirma as alterações;</p> <p>h) O caso de uso volta para o passo “c” do fluxo principal</p> <p>A4 - Inativar usuário</p> <p>a) O ator clica em inativar usuário;</p> <p>b) O ator preenche os campos desejados para inativação;</p> <p>c) O ator seleciona o usuário desejado para inativação;</p> <p>d) O sistema habilita o botão de confirmar;</p> <p>e) O ator confirma a inativação;</p> <p>f) O caso de uso volta para o passo “c” do fluxo principal.</p>
--	---

Tabela 7: UC 02 - Manter usuário.

4.3.3. UC 03 - Manter propriedade

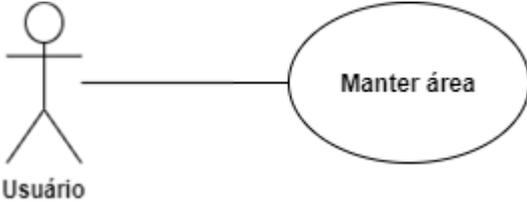
	
<p>1 - Finalidade/Objetivo</p>	<p>Realizar o cadastro, alteração, inativação e consulta das propriedades.</p>

2 - Atores	Usuário.
3 - Pré-Condições	Estar logado.
4 - Evento inicial	O ator começa o caso de uso selecionando a opção de manter propriedade.
5 - Fluxo Principal	<p>a) O sistema apresenta a interface;</p> <p>b) O sistema mostra as opções de manter propriedade; (A1) (A2) (A3) (A4)</p> <p>c) O caso de uso é finalizado.</p>
6 - Fluxo Alternativo	<p>A1 - Pesquisar propriedade</p> <p>a) O ator clica em pesquisar propriedade;</p> <p>b) O ator preenche os campos desejados para pesquisa;</p> <p>c) O ator seleciona a opção de pesquisar;</p> <p>d) O sistema exibe o resultado da pesquisa;</p> <p>e) O caso de uso volta para o passo “c” do fluxo principal.</p> <p>A2 - Cadastrar propriedade</p> <p>a) O ator clica em cadastrar propriedade;</p> <p>b) Todos os campos da tela são habilitados;</p> <p>c) O ator preenche os campos;</p> <p>d) O sistema valida os campos preenchidos;</p> <p>e) O sistema habilita o botão de confirmar;</p> <p>f) O ator confirma o cadastro;</p> <p>g) O caso de uso volta para o passo “c” do fluxo principal.</p> <p>A3 - Alterar propriedade</p> <p>a) O ator clica em alterar propriedade;</p> <p>b) Todos os campos da tela são habilitados;</p> <p>c) O ator altera os campos desejados;</p> <p>d) O sistema valida os campos preenchidos;</p> <p>e) O sistema habilita o botão de confirmar;</p>

	<p>g) O ator confirma as alterações;</p> <p>h) O caso de uso volta para o passo “c” do fluxo principal</p> <p>A4 - Inativar propriedade</p> <p>a) O ator clica em inativar propriedade;</p> <p>b) O ator preenche os campos desejados para inativação;</p> <p>c) O ator seleciona a propriedade desejado para inativação;</p> <p>d) O sistema habilita o botão de confirmar;</p> <p>e) O ator confirma a inativação;</p> <p>f) O caso de uso volta para o passo “c” do fluxo principal.</p>
--	--

Tabela 8: UC 03 - Manter propriedade.

4.3.4. UC 04 - Manter área

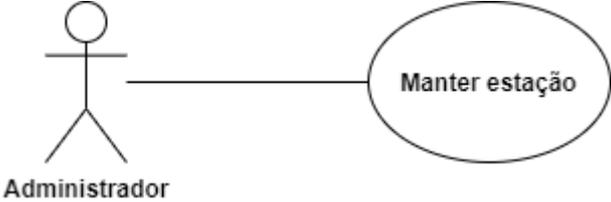
	
1 - Finalidade/Objetivo	Realizar o cadastro, alteração, inativação e consulta das áreas.
2 - Atores	Usuário.
3 - Pré-Condições	Estar logado.
4 - Evento inicial	O ator começa o caso de uso selecionando a opção de manter área.
5 - Fluxo Principal	<p>a) O sistema apresenta a interface;</p> <p>b) O sistema mostra as opções de manter área;</p> <p>(A1) (A2) (A3) (A4)</p> <p>c) O caso de uso é finalizado.</p>

6 - Fluxo Alternativo	<p>A1 - Pesquisar área</p> <ul style="list-style-type: none">a) O ator clica em pesquisar área;b) O ator preenche os campos desejados para pesquisa;c) O ator seleciona a opção de pesquisar;d) O sistema exibe o resultado da pesquisa;e) O caso de uso volta para o passo “c” do fluxo principal. <p>A2 - Cadastrar área</p> <ul style="list-style-type: none">a) O ator clica em cadastrar área;b) Todos os campos da tela são habilitados;c) O ator preenche os campos;d) O sistema valida os campos preenchidos;e) O sistema habilita o botão de confirmar;f) O ator confirma o cadastro;g) O caso de uso volta para o passo “c” do fluxo principal. <p>A3 - Alterar área</p> <ul style="list-style-type: none">a) O ator clica em alterar área;b) Todos os campos da tela são habilitados;c) O ator altera os campos desejados;d) O sistema valida os campos preenchidos;e) O sistema habilita o botão de confirmar;g) O ator confirma as alterações;h) O caso de uso volta para o passo “c” do fluxo principal <p>A4 - Inativar área</p> <ul style="list-style-type: none">a) O ator clica em inativar área;b) O ator preenche os campos desejados para inativação;
-----------------------	---

	<p>c) O ator seleciona a área desejado para inativação;</p> <p>d) O sistema habilita o botão de confirmar;</p> <p>e) O ator confirma a inativação;</p> <p>f) O caso de uso volta para o passo “c” do fluxo principal.</p>
--	---

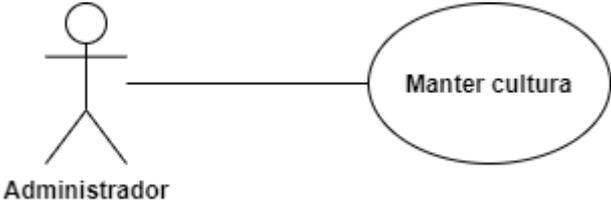
Tabela 9: UC 04 - Manter área.

4.3.5. UC 05 - Manter estação

 <p>Administrador</p> <p>Manter estação</p>	
1 - Finalidade/Objetivo	Realizar o cadastro, alteração, inativação e consulta das estações.
2 - Atores	Administrador.
3 - Pré-Condições	Estar logado.
4 - Evento inicial	O ator começa o caso de uso selecionando a opção de manter estação.
5 - Fluxo Principal	<p>a) O sistema apresenta a interface;</p> <p>b) O sistema mostra as opções de manter estação;</p> <p>(A1) (A2) (A3) (A4)</p> <p>c) O caso de uso é finalizado.</p>
6 - Fluxo Alternativo	<p>A1 - Pesquisar estação</p> <p>a) O ator clica em pesquisar estação;</p> <p>b) O ator preenche os campos desejado para pesquisa;</p> <p>c) O ator seleciona a opção de pesquisar;</p> <p>d) O sistema exibe o resultado da pesquisa;</p> <p>e) O caso de uso volta para o passo “c” do fluxo principal.</p>

	<p>A2 - Cadastrar estação</p> <p>a) O ator clica em cadastrar estação;</p> <p>b) Todos os campos da tela são habilitados;</p> <p>c) O ator preenche os campos;</p> <p>d) O sistema valida os campos preenchidos;</p> <p>e) O sistema habilita o botão de confirmar;</p> <p>f) O ator confirma o cadastro;</p> <p>g) O caso de uso volta para o passo “c” do fluxo principal.</p> <p>A3 - Alterar estação</p> <p>a) O ator clica em alterar estação;</p> <p>b) Todos os campos da tela são habilitados;</p> <p>c) O ator altera os campos desejados;</p> <p>d) O sistema valida os campos preenchidos;</p> <p>e) O sistema habilita o botão de confirmar;</p> <p>g) O ator confirma as alterações;</p> <p>h) O caso de uso volta para o passo “c” do fluxo principal</p> <p>A4 - Inativar estação</p> <p>a) O ator clica em inativar estação;</p> <p>b) O ator preenche os campos desejados para inativação;</p> <p>c) O ator seleciona a estação desejado para inativação;</p> <p>d) O sistema habilita o botão de confirmar;</p> <p>e) O ator confirma a inativação;</p> <p>f) O caso de uso volta para o passo “c” do fluxo principal.</p>
--	---

Tabela 10: UC 05 - Manter estação.

	
1 - Finalidade/Objetivo	Realizar o cadastro, alteração, inativação e consulta das culturas.
2 - Atores	Administrador.
3 - Pré-Condições	Estar logado.
4 - Evento inicial	O ator começa o caso de uso selecionando a opção de manter cultura.
5 - Fluxo Principal	<ul style="list-style-type: none"> a) O sistema apresenta a interface; b) O sistema mostra as opções de manter cultura; (A1) (A2) (A3) (A4) c) O caso de uso é finalizado.
6 - Fluxo Alternativo	<p>A1 - Pesquisar cultura</p> <ul style="list-style-type: none"> a) O ator clica em pesquisar cultura; b) O ator preenche os campos desejados para pesquisa; c) O ator seleciona a opção de pesquisar; d) O sistema exibe o resultado da pesquisa; e) O caso de uso volta para o passo “c” do fluxo principal. <p>A2 - Cadastrar cultura</p> <ul style="list-style-type: none"> a) O ator clica em cadastrar cultura; b) Todos os campos da tela são habilitados; c) O ator preenche os campos; d) O sistema valida os campos preenchidos; e) O sistema habilita o botão de confirmar; f) O ator confirma o cadastro; g) O caso de uso volta para o passo “c” do fluxo principal.

	<p>A3 - Alterar cultura</p> <p>a) O ator clica em alterar cultura;</p> <p>b) Todos os campos da tela são habilitados;</p> <p>c) O ator altera os campos desejados;</p> <p>d) O sistema valida os campos preenchidos;</p> <p>e) O sistema habilita o botão de confirmar;</p> <p>g) O ator confirma as alterações;</p> <p>h) O caso de uso volta para o passo “c” do fluxo principal</p> <p>A4 - Inativar cultura</p> <p>a) O ator clica em inativar cultura;</p> <p>b) O ator preenche os campos desejados para inativação;</p> <p>c) O ator seleciona a cultura desejado para inativação;</p> <p>d) O sistema habilita o botão de confirmar;</p> <p>e) O ator confirma a inativação;</p> <p>f) O caso de uso volta para o passo “c” do fluxo principal.</p>
--	---

Tabela 11: UC 06 - Manter cultura.

4.3.7. UC 07 - Manter doenças

 <pre> graph LR A((Administrador)) --- UC((Manter doenças)) </pre>	
1 - Finalidade/Objetivo	Realizar o cadastro, alteração, inativação e consulta das doenças.
2 - Atores	Administrador.
3 - Pré-Condições	Estar logado.

4 - Evento inicial	O ator começa o caso de uso selecionando a opção de manter doença.
5 - Fluxo Principal	a) O sistema apresenta a interface; b) O sistema mostra as opções de manter doença; (A1) (A2) (A3) (A4) c) O caso de uso é finalizado.
6 - Fluxo Alternativo	<p>A1 - Pesquisar doença</p> a) O ator clica em pesquisar doença; b) O ator preenche os campos desejados para pesquisa; c) O ator seleciona a opção de pesquisar; d) O sistema exibe o resultado da pesquisa; e) O caso de uso volta para o passo “c” do fluxo principal. <p>A2 - Cadastrar doença</p> a) O ator clica em cadastrar doença; b) Todos os campos da tela são habilitados; c) O ator preenche os campos; d) O sistema valida os campos preenchidos; e) O sistema habilita o botão de confirmar; f) O ator confirma o cadastro; g) O caso de uso volta para o passo “c” do fluxo principal. <p>A3 - Alterar doença</p> a) O ator clica em alterar doença; b) Todos os campos da tela são habilitados; c) O ator altera os campos desejados; d) O sistema valida os campos preenchidos; e) O sistema habilita o botão de confirmar; g) O ator confirma as alterações; h) O caso de uso volta para o passo “c” do fluxo principal.

	<p>principal</p> <p>A4 - Inativar doença</p> <p>a) O ator clica em inativar doença;</p> <p>b) O ator preenche os campos desejados para inativação;</p> <p>c) O ator seleciona a doença desejado para inativação;</p> <p>d) O sistema habilita o botão de confirmar;</p> <p>e) O ator confirma a inativação;</p> <p>f) O caso de uso volta para o passo “c” do fluxo principal.</p>
--	---

Tabela 12: UC 07 - Manter doenças.

4.3.8. UC 08 - Emitir relatórios de dados meteorológicos

1 - Finalidade/Objetivo	Realizar a emissão do relatório de dados meteorológicos.
2 - Atores	Administrador e usuário
3 - Pré-Condições	Estar logado.
4 - Evento inicial	O ator começa o caso de uso selecionando a opção de emitir relatório de dados meteorológicos.
5 - Fluxo Principal	<p>a) O sistema apresenta a interface;</p> <p>b) O ator clica em relatório de dados meteorológicos;</p> <p>c) O ator seleciona quais dados devem estar no relatório;</p> <p>d) O ator seleciona a opção de emitir relatório de</p>

	<p>dados meteorológicos (A1);</p> <p>e) O sistema exibe o relatório de dados meteorológicos (A2);</p> <p>f) O caso de uso é finalizado.</p>
6 - Fluxo Alternativo	<p>A1 - O ator cancela a emissão do relatório</p> <p>a) O ator seleciona a opção cancelar a emissão do relatório de dados meteorológicos;</p> <p>b) O caso de uso volta para o passo “a” do fluxo principal;</p> <p>A2 - O ator envia para a impressão</p> <p>a) O ator seleciona a opção de imprimir o relatório de dados meteorológicos;</p> <p>b) O caso de uso volta para o passo “a” do fluxo principal;</p>

Tabela 13: UC 08 - Emitir relatórios de dados meteorológicos.

Os diagramas de casos de uso apresentados representam as funcionalidades e suas respectivas narrativas do sistema para as entidades "Administrador e Usuário".

4.4. DIAGRAMA DE ATIVIDADES

O diagrama de atividades é definido pela UML para representação do fluxo de controle de uma determinada atividade para a outra de forma que possa ser ilustrada uma modelagem dos aspectos dinâmicos do sistema, logo, fornecendo um entendimento transparente para uma determinada ação (BOOCH, RUMBAUGH e JACOBSON, 2006).

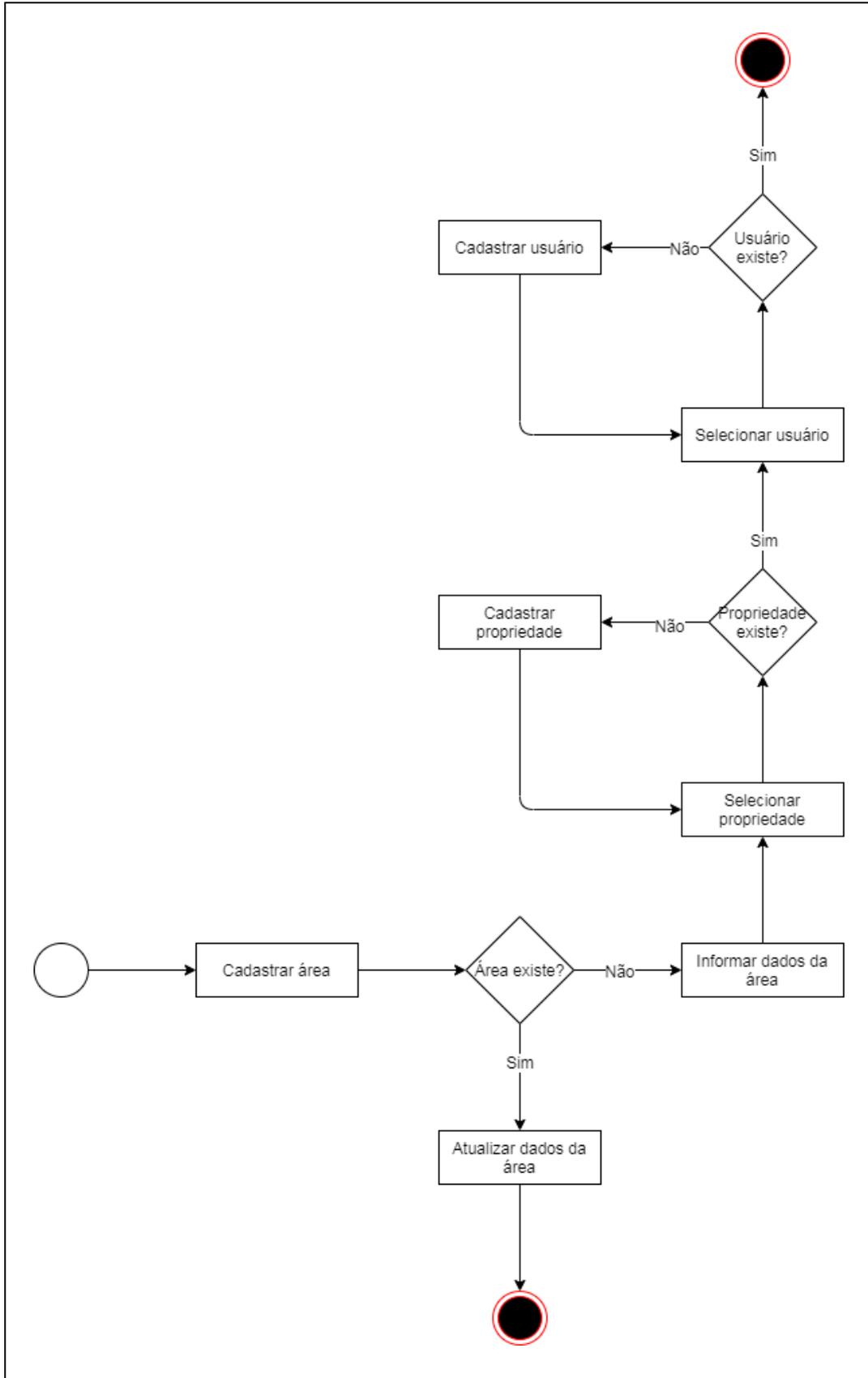


Figura 19: Diagrama de atividade para cadastrar área.
Fonte: Própria, 2020.

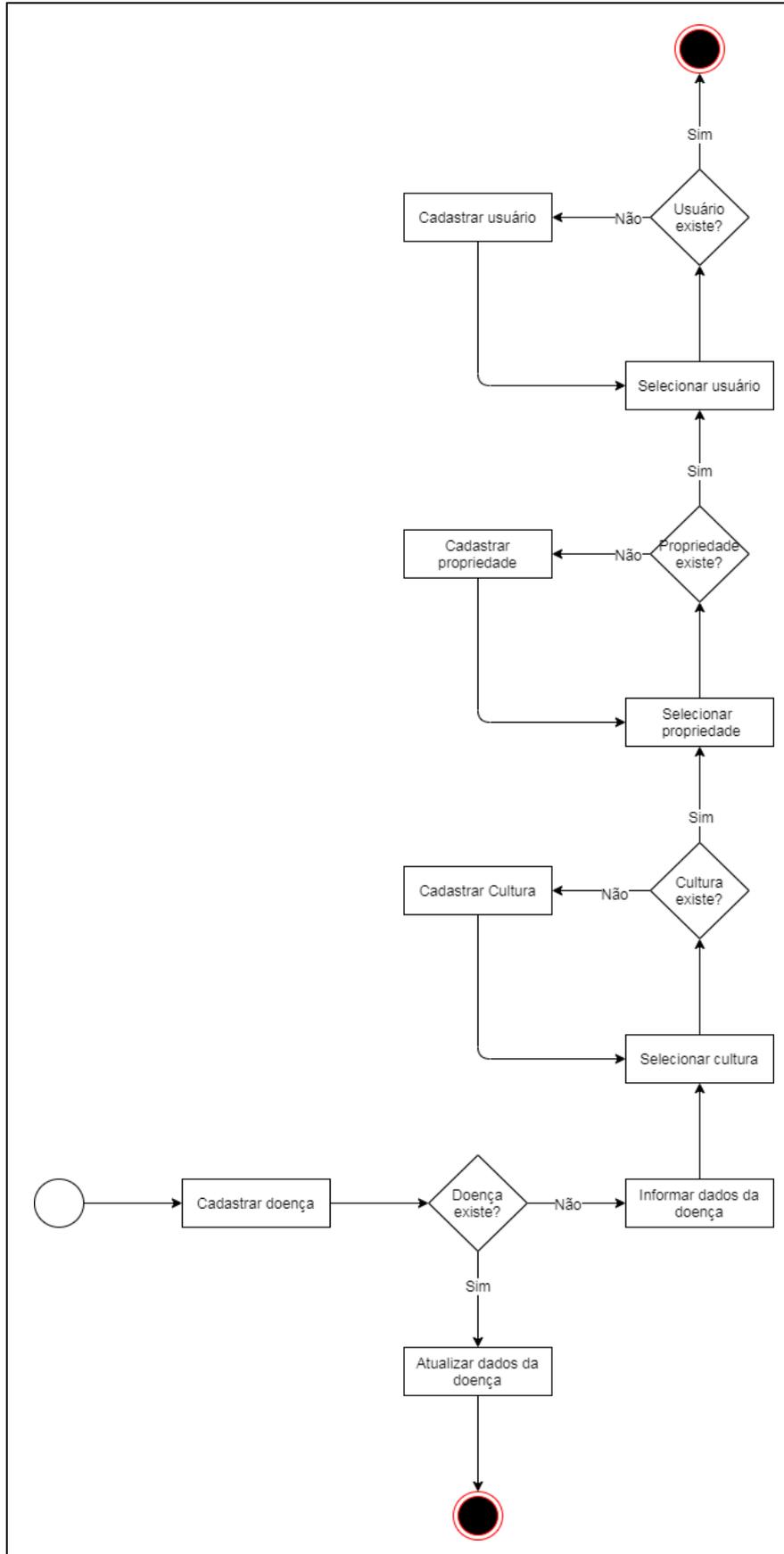


Figura 20: Diagrama de atividade para cadastrar doenças.
Fonte: Própria, 2020.

Os diagramas de atividade apresentados ilustram o processo de uma determinada atividade.

4.5. DIAGRAMA DE SEQUÊNCIA

Um diagrama de sequência representa a troca de mensagens entre objetos de um determinado sistema, o que possibilita a visualização da troca de informação por meio de mensagens enviadas entre objetos contendo informações relacionadas ao resultado de uma determinada atividade (BOOCH, RUMBAUGH e JACOBSON, 2006).

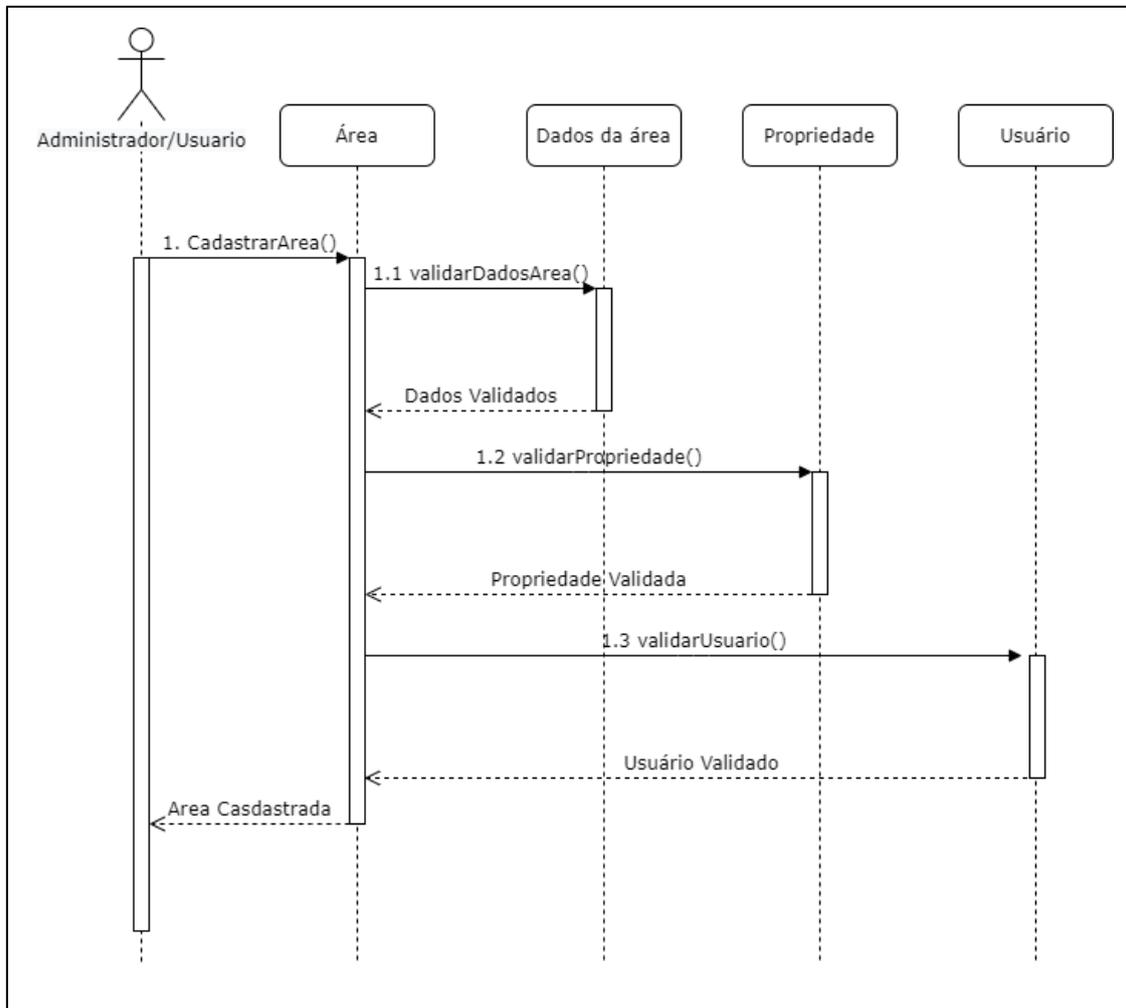


Figura 21: Diagrama de sequência cadastrar área.
Fonte: Própria, 2020.

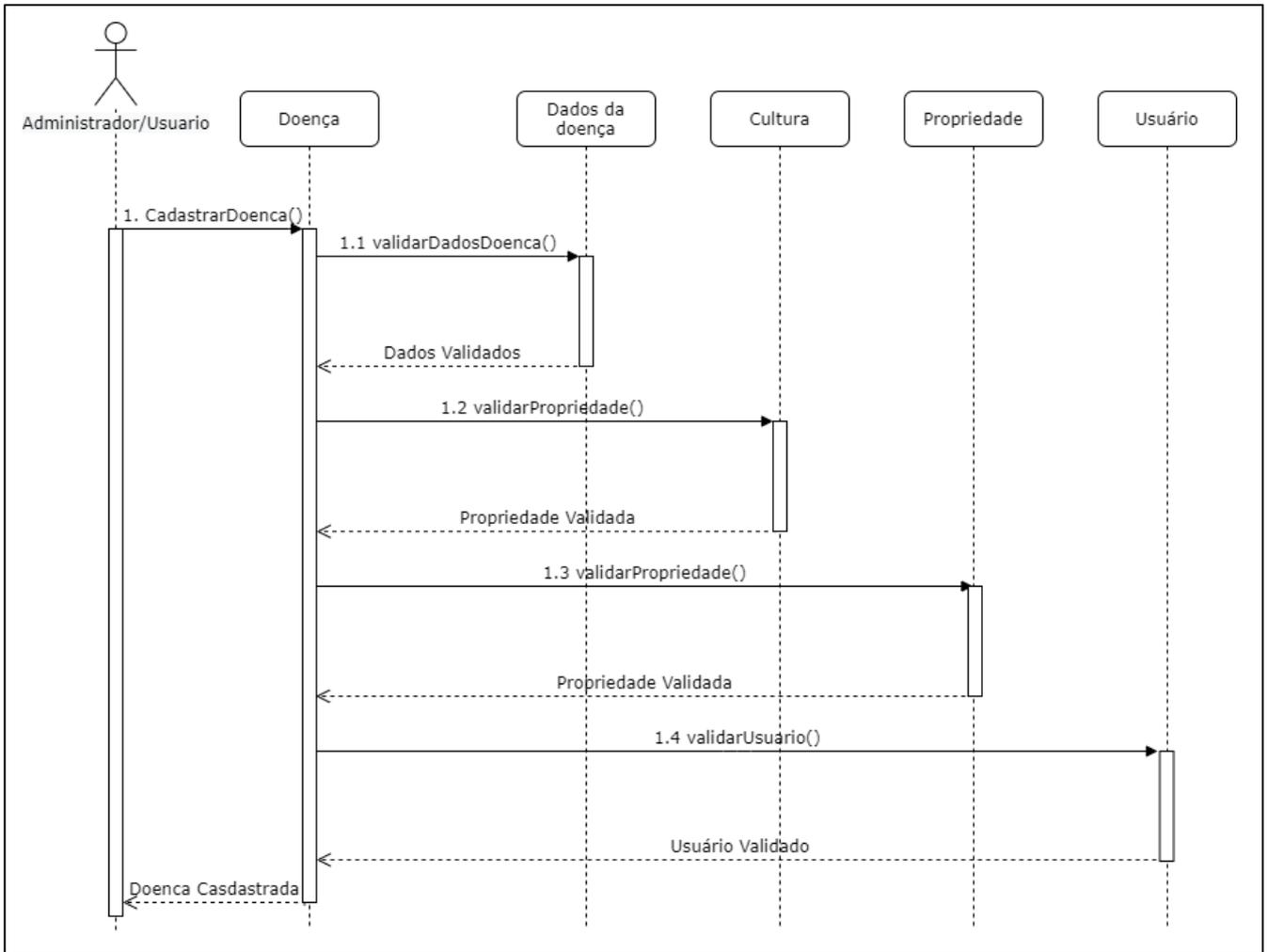


Figura 22: Diagrama de sequência cadastrar doenças.

Fonte: Própria, 2020.

Os diagramas de sequência apresentados ilustram a troca de mensagens para a execução de uma determinada atividade solicitada pela entidade.

4.6. DIAGRAMA DE CLASSE

O diagrama de classes ilustra um conjunto de classes e seus relacionamentos na forma estática de um sistema; eles são importantes não apenas para documentação, mas também para o desenvolvimento do sistema, contribuindo de maneira impactante para a consolidação do escopo das funcionalidades do projeto, fornecendo um alto nível de modelagem de esquemas, como ilustra a figura 21 (BOOCH, RUMBAUGH e JACOBSON, 2006).

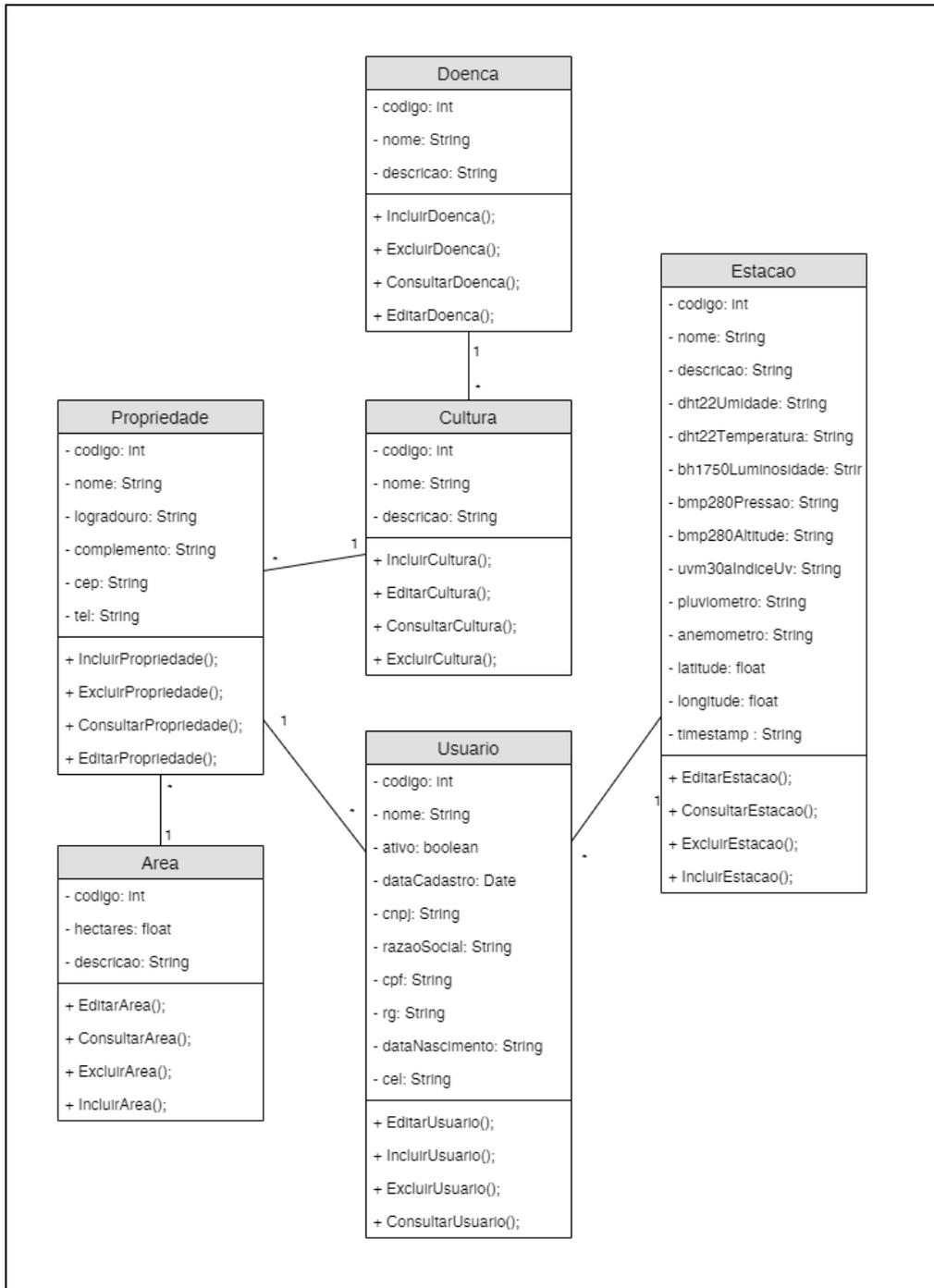


Figura 23: Diagrama de classe.

Fonte: Própria, 2020.

O diagrama apresentado define todas as classes, atributos e relacionamentos existentes no sistema.

4.7. DIAGRAMA ENTIDADE-RELACIONAMENTO

O Diagrama entidade-relacionamento ilustra a estrutura lógica do banco de dados descrito como um conjunto de objetos básicos chamados entidade e seus relacionamentos entre estes objetos, de forma a facilitar a especificação e desenvolvimento do esquema de banco de dados, como ilustra a figura 22 (REZENDE, 2006).

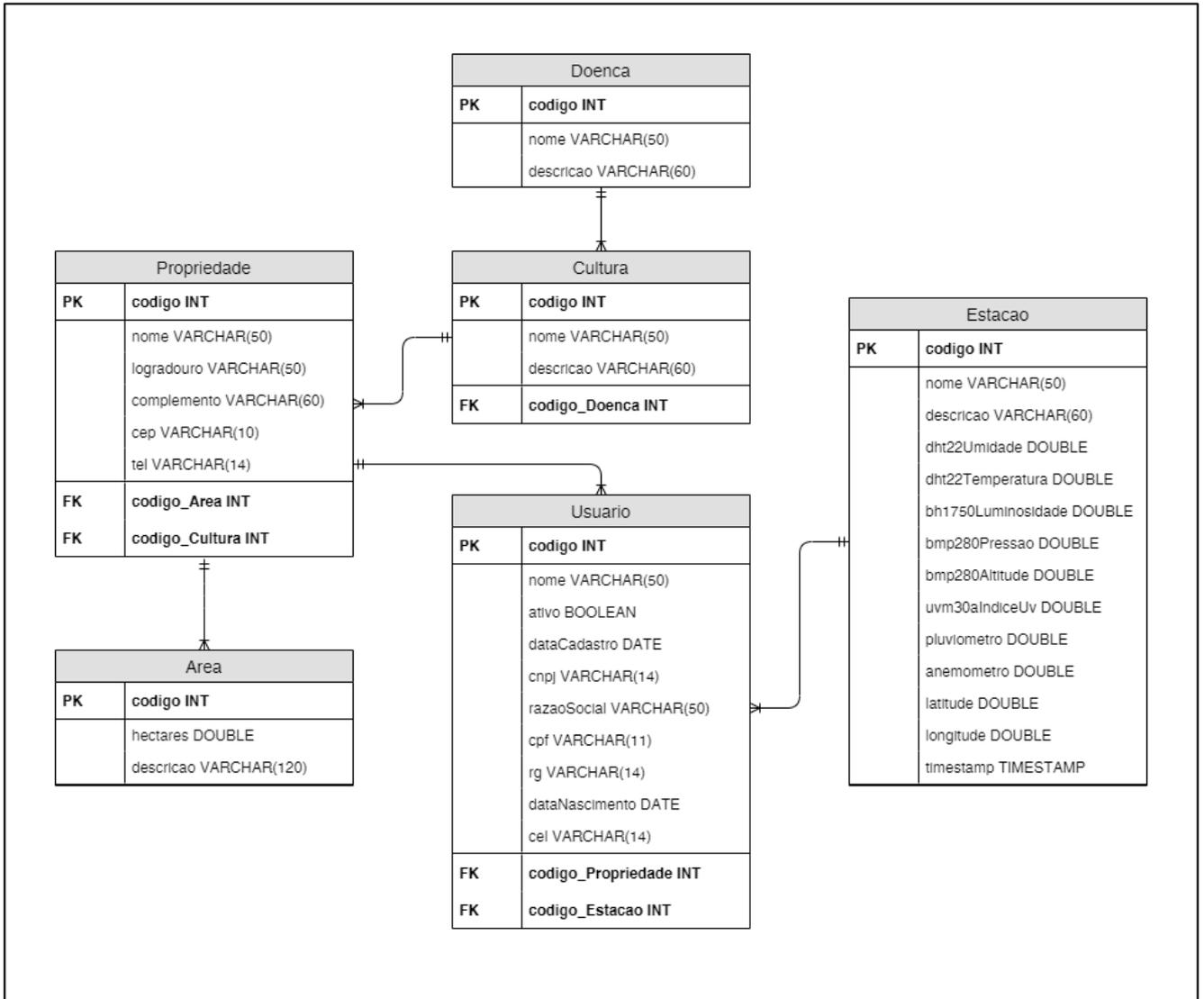


Figura 24: Diagrama de ER.
Fonte: Própria, 2020.

O diagrama apresentado ilustra as tabelas e relacionamentos existentes no banco de dados do sistema.

5. DESENVOLVIMENTO DO PROJETO

Este trabalho tem por objetivo abordar a proposta de desenvolver um protótipo de IOT para Smart Agriculture com foco em monitoramento meteorológico, bem como desenvolver uma aplicação WebApp para consumir os dados do protótipo. A leitura desses dados do protótipo se dará por meio dos sensores que captarão temperatura, umidade do ar, pressão atmosférica, altitude, luz ultravioleta, luminosidade, volume de chuva e velocidade do vento.

O aplicativo desenvolvido será usado para exibir os dados coletados pelo protótipo, podendo emitir alertas, fazer previsões sobre a possibilidade futura de uma cultura desenvolver algum tipo de doença, entre outros. Logo, ajudará o usuário a tomar uma decisão.

5.1. PROTÓTIPO WEBAPP

Os protótipos de sites geralmente são demonstrações de um site no início do ciclo de vida do projeto. Eles são usados para obter um acordo e coletar opiniões dos vários participantes do projeto, incluindo os usuários finais. Normalmente, um protótipo de site passa por algumas rodadas de testes de usabilidade para garantir que tudo funcione antes do projeto se desenvolver. A questão de se preocupar em criar um protótipo deve ser atendida com uma pergunta diferente: está disposto a se arriscar diretamente ao desenvolvimento sem testar suas suposições previamente? Embora possa levar algum tempo para escolher a ferramenta certa e aprendê-la, esse tempo é superado pelos custos envolvidos no lançamento de um site não testado que não funciona (CARMICHAEL e REES, 2020).

Dessa forma, a figura 25, ilustra a dashboard do protótipo do WebApp, demonstrando algumas informações meteorológicas captadas do protótipo de IoT para Smart Agriculture com foco em monitoramento meteorológico em tempo real.

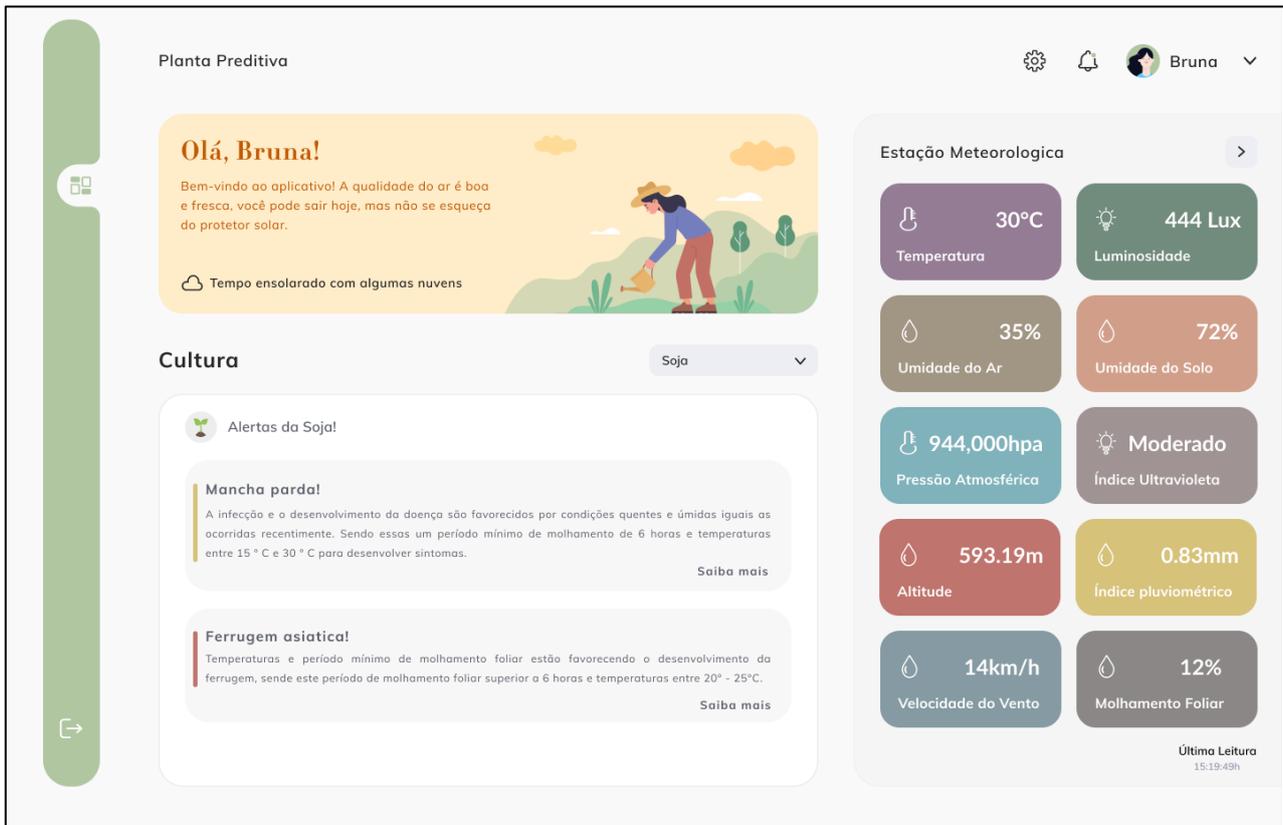


Figura 25: Protótipo do WebApp.
Fonte: *Própria*, 2020.

O WebApp também irá emitir alertas e avisos, auxiliando assim a tomada de decisão do usuário.

5.2. PROTÓTIPO IOT

O hardware da IoT vem em uma enorme variedade de formatos. O Arduino e o Raspberry Pi, duas das plataformas mais comuns, são plataformas completamente diferentes, e não se pode arquitetar seu software e hardware da mesma maneira para eles. O impacto de decisões de nível superior, tal como escolher qual sistema operacional adotar, será menor do que a escolha da sua plataforma. O primeiro passo crítico é escolher o caminho certo para o seu projeto. Para esse projeto a placa escolhida foi a NodeMcu ESP-12E (Amica), que é apenas uma variação do chipset ESP8266

Nas figuras 26 e 27 temos a modelagem do protótipo de IoT para Smart Agriculture com foco em monitoramento meteorológico, onde podemos visualizar todos os componentes eletrônicos e suas conexões, necessários para a montagem física deste.

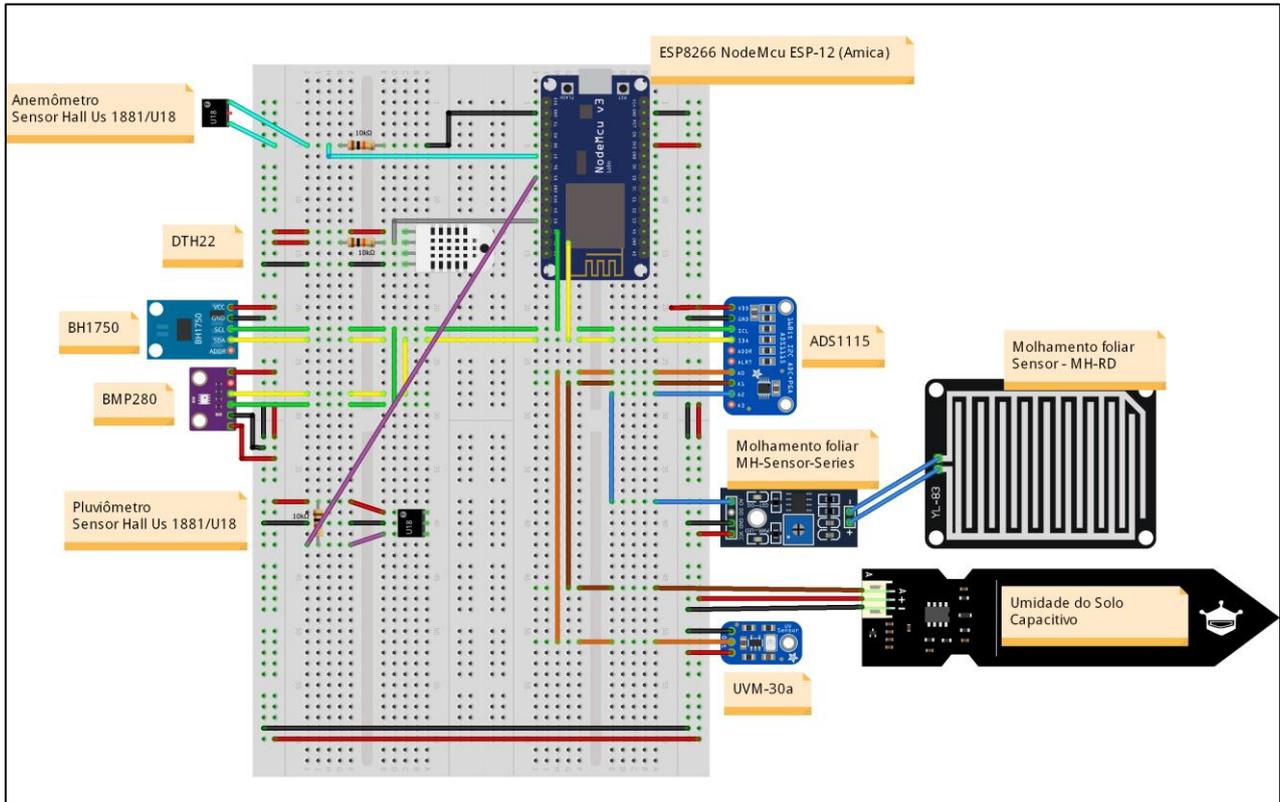


Figura 26: Modelagem NodeMcu ESP-12E e sensores do protótipo.
Fonte: Própria, 2020.

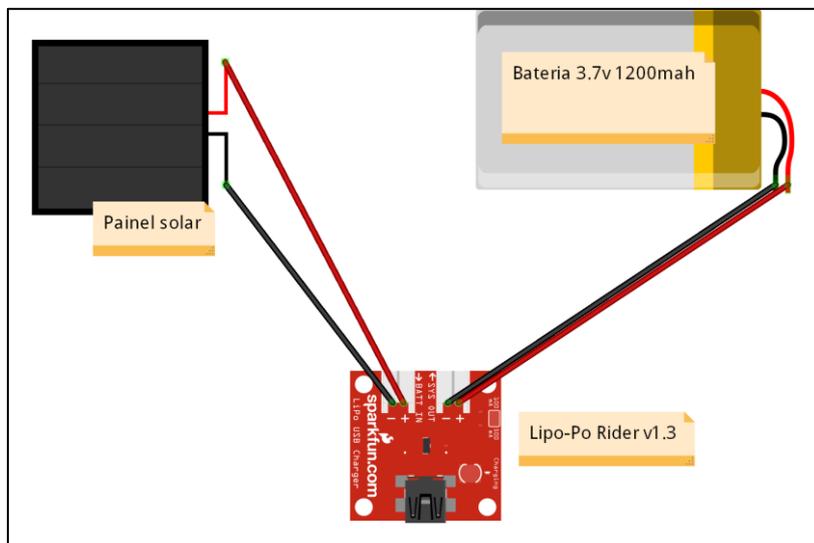


Figura 27: Modelagem do módulo gerenciador de energia do protótipo.
Fonte: Própria, 2020.

Após a modelagem, foi criado o protótipo físico, como ilustrado na figura 28, tendo o mesmo a capacidade de registrar temperatura, umidade, pressão atmosférica, altitude, luz ultravioleta, luminosidade, umidade do solo, quantidade de chuva e velocidade do vento.

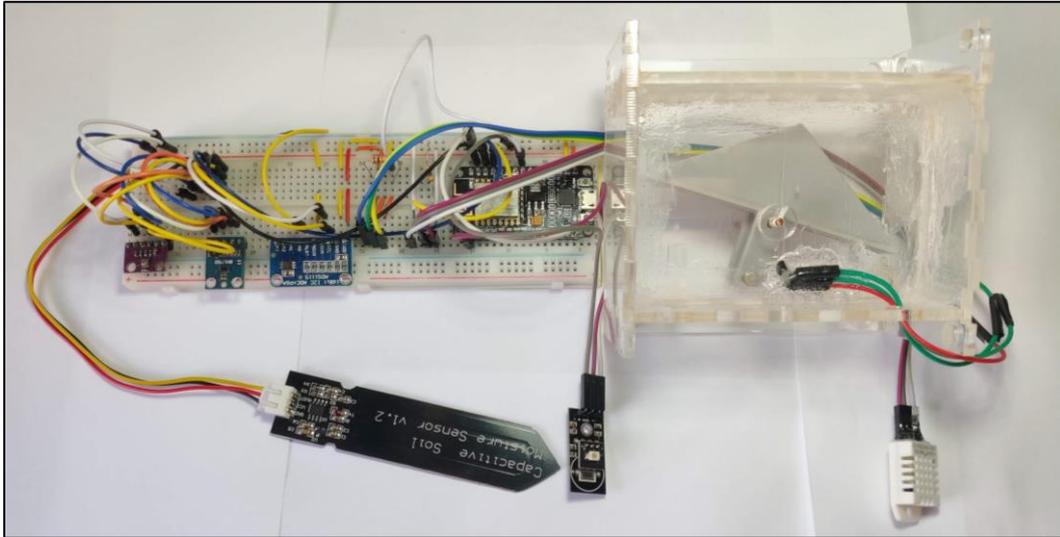


Figura 28: Protótipo NodeMcu ESP-12E e sensores do protótipo.

Fonte: Própria, 2020.

Já a figura 29 ilustra o gerenciador de energia do protótipo, que permite o gerenciamento de energia produzida por meio de painéis solares, conduzindo essa tensão de forma controlada até a bateria ou dispositivos externos, neste caso o NodeMcu ESP-12E.

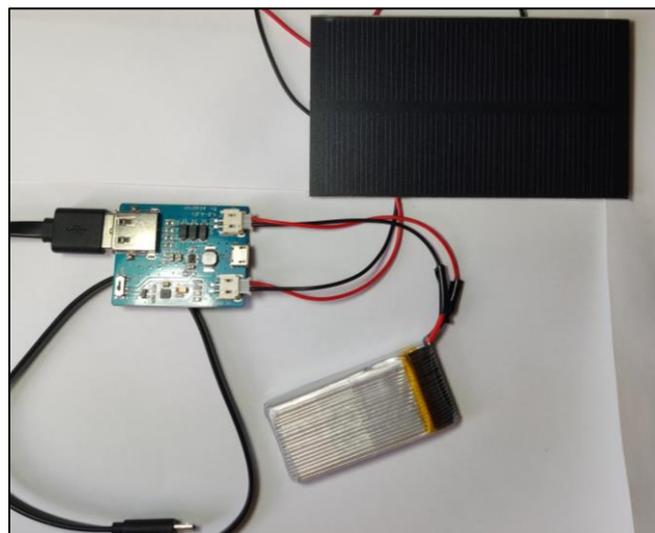


Figura 29: modulo gerenciador de energia do protótipo.

Fonte: Própria, 2020.

A figura 30 representa o pluviômetro de balança digital, que utiliza um sensor Hall com um ímã anexado em uma das pernas da balança. Mediante a ocorrência de chuva, a água é canalizada para a balança e, à medida que a água é coletada, a balança enche até o ponto em que tomba, causando a passagem do ímã pelo sensor Hall, ocasionando um incremento no acúmulo de chuva no protótipo. Essa ação esvazia a balança para as próximas medições.



Figura 30: Pluviômetro de balança digital.
Fonte: Própria, 2020.

A figura 31 representa o anemômetro. Este equipamento utiliza-se eletronicamente de um sistema de interruptor chamado Reed Switch, contendo duas placas de ferro separadas que, quando encostadas, mandam um sinal para o NodeMcu ESP-12E. Toda vez que o vento bate nas pás, a parte superior começa a girar, acionando o sensor Reed Switch e proporcionando a medição da velocidade do vento.

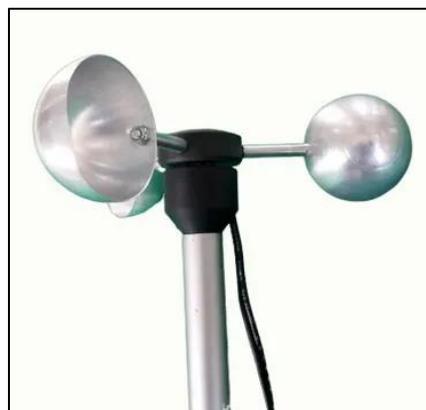


Figura 31: Anemômetro.
Fonte: Própria, 2020.

A figura 32 configura a versão final do protótipo.

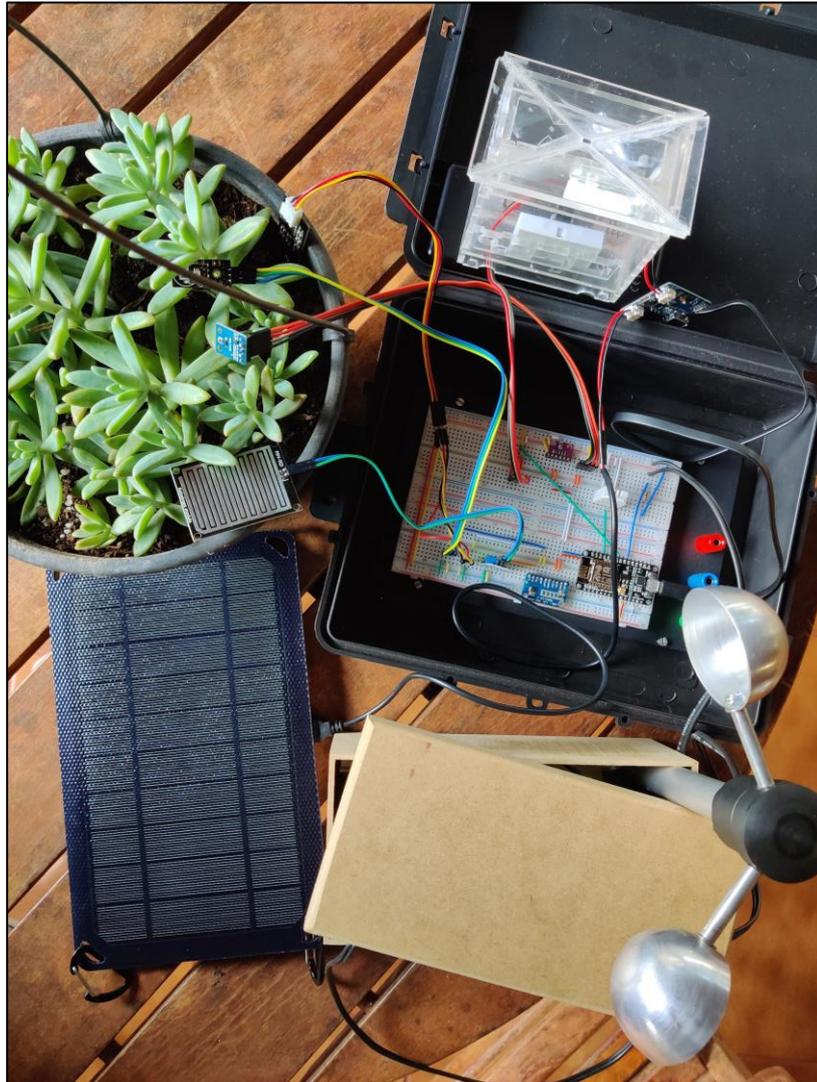


Figura 32: Versão final do protótipo.
Fonte: Própria, 2020.

Mesmo diante de todos os percalços e com todos os itens descritos durante a exposição deste trabalho, foi possível montar um protótipo final, que, mediante alguns ajustes pode trazer uma contribuição significativa à Smart Agriculture, uma vez que é capaz de monitorar a temperatura, umidade do ar, índice pluviométrico, umidade do solo, pressão atmosférica, incidência de raios UV e a luminosidade oferecida pela luz solar, facilitando, assim, a tomada de decisão do usuário.

5.3. DIFICULDADES ENCONTRADAS

Com base em algumas falhas de funcionamento detectadas e a descalibragem de alguns sensores durante o desenvolvimento, podemos citar que o projeto pode ter sua qualidade comprometida em algum momento devido à aquisição de sensores de baixo custo.

6. CONCLUSÕES

O protótipo de IOT para Smart Agriculture com foco em monitoramento meteorológico apresenta diversas funcionalidades implementadas com o intuito de facilitar o usuário a tomar uma decisão, podendo emitir alertas, fazer previsões sobre a possibilidade futura de uma cultura desenvolver algum tipo de doença, entre outros. Sendo assim, é possível concluir que o mesmo seria um grande atenuador das dificuldades encontradas pelo agricultor no seu dia a dia, tal como a falta de estações meteorológicas e dados meteorológicos, por conta do alto custo de sua implantação, além da escassez de informações meteorológicas locais.

Com o desenvolvimento deste trabalho foi possível adquirir maior experiência com relação ao planejamento, modelagem e desenvolvimento de software. Na modelagem de dados, a UML foi essencial no desenvolvimento do sistema e contribuiu de forma impactante no aprendizado. Na pesquisa, houve contato com conceitos que geraram conhecimento extra em *JavaScript*, *React*, *React-Router*, *Redux*, *Webpack*, *Babel*, *node*, *Firebase* que são tecnologias atuais e muito utilizadas no mercado de trabalho.

6.1. TRABALHOS FUTUROS

Dentro do objetivo proposto, que é um Protótipo de IoT para Smart Agriculture, com foco em monitoramento meteorológico de baixo custo, este se mostrou eficiente e acessível, facilitando a tomada de decisão do usuário.

A sugestão para trabalhos futuros é melhorar a aparência do circuito, elaborando uma placa de circuito impresso com os componentes e sensores necessários para o funcionamento, dando um aspecto mais limpo e profissional. Uma boa melhoria, também, seria a construção de um abrigo melhor desenvolvido.

Por fim, é de se ressaltar que, para a obtenção de resultados mais fidedignos, seria necessário realizar um estudo maior em relação à calibração de sensores ou a aquisição e instalação de sensores mais profissionais.

REFERÊNCIAS

- BABEL. **What is Babel?** Disponível em: <<https://babeljs.io/docs/en>>. Acesso em: 28 set. 2019.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **Uml - Guia do Usuário**. 2ª. ed. Rio de Janeiro: Elsevier Brasil, 2006.
- BROWN, M. **Smart Farming - Automated and Connected Agriculture**. Disponível em: <<https://www.engineering.com/DesignerEdge/DesignerEdgeArticles/ArticleID/16653/Smart-FarmingAutomated-and-Connected-Agriculture>>. Acesso em: 28 set. 2019.
- CARMICHAEL, A.; REES, D. **Experience UX**. Disponível em: <<https://www.experienceux.co.uk/faqs/what-is-a-website-prototype>>. Acesso em: 05 mar. 2020.
- CIRCUITO, C. **Sensor de Umidade do Solo Capacitivo**. Disponível em: <<https://www.curtocircuito.com.br/sensor-de-umidade-do-solo-capacitivo.html>>. Acesso em: 26 fev. 2020.
- COELNER et al. **BH1750**. Disponível em: <<https://github.com/claws/BH1750>>. Acesso em: 26 fev. 2020.
- ELECTRONICS, S. **REED SWITCH**. Disponível em: <https://standexelectronics.com/wp-content/uploads/OKI_Reed_Switch_ORD213.pdf>. Acesso em: 26 fev. 2020.
- ENGINEERS, L. M. **Insight Into ESP8266 NodeMCU Features & Using It With Arduino IDE**, 2020. Disponível em: <<https://lastminuteengineers.com/esp8266-nodemcu-arduino-tutorial>>. Acesso em: 25 fev. 2020.
- FAO. Food and Agriculture Organization. **World Food Situation**. Disponível em: <<http://www.fao.org/worldfoodsituation/csdb/en>>. Acesso em: 16 out. 2019.
- FIREBASE. **Firestore por plataforma**. Disponível em: <<https://firebase.google.com/docs>>. Acesso em: 28 out. 2019.
- GREBMER, K. V. et al. **2013 Global hunger index : The challenge of hunger : Building resilience to achieve food and nutrition security**. International Food Policy Research Institute. Washington. 2013.

HEK. **UV Sensor**. Disponível em: <<https://www.mysensors.org/build/uv>>. Acesso em: 26 fev. 2020.

LIU, T. **DHT22 Sensor Datasheet**. Disponível em: <https://components101.com/sites/default/files/component_datasheet/DHT22%20Sensor%20Datasheet.pdf>. Acesso em: 25 fev. 2020.

MASSRUHÁ, S. M. F. S. et al. **Tecnologias da informação e comunicação e suas relações com a agricultura**. 1ª. ed. Brasília: Embrapa, 2014.

PRESCOTT, P. **Programação em JavaScript**. 1ª. ed. [S.l.]: Babelcube, 2016.

REACT. **Getting Started**. Disponível em: <<https://reactjs.org/docs/getting-started.html>>. Acesso em: 28 out. 2019.

REDUX. **Getting Started with Redux**. Disponível em: <<https://redux.js.org/introduction/getting-started>>. Acesso em: 28 out. 2019.

REZENDE, D. A. **Engenharia de Software e Sistemas de Informação**. 3ª. ed. Rio de Janeiro: Brasport, 2006.

ROCO, M. C.; BAINBRIDGE, W. S. **Converging Technologies for Improving Human Performance**. 1ª. ed. Dordrecht: Springer Netherlands, 2003.

RUGGIERI, R. **WBS - Uma ferramenta importante para o Gerente de Projetos**. Disponível em: <http://www.tiespecialistas.com.br/2010/11/wbs-%E2%80%93-uma-ferramenta-importante-para-o-gerente-de-projetos/#.UTA0B6J_CSo>. Acesso em: 24 out. 2019.

SENSORTEC, B. **BMP280 Digital Pressure Sensor**. Disponível em: <<https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf>>. Acesso em: 26 fev. 2020.

SOUZA, L. I. G. D. **O Redesign da Informação no Processamento de Imagem**. Tese (doutorado) - Universidade Federal de Santa Catarina. Florianópolis. 2004.

TECHNOLOGIES, U. **Bipolar Latch Type Hall-effect For High-temperature Operation**. Disponível em: <<http://www.utc-ic.com/uploadfile/2012/0331/20120331125747992.pdf>>. Acesso em: 26 fev. 2020.

TILKOV, S.; VINOSKI, S. **Node.js: Using JavaScript to Build High-Performance Network Programs**, IEEE Internet Computing, v. 14, p. 80-83, nov. 2010.

WEBPACK. **Concepts**. Disponível em: <<https://webpack.js.org/concepts>>. Acesso em: 28 set. 2019.