

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E  
TECNOLOGIA DE SÃO PAULO  
CÂMPUS GUARULHOS

PEDRO IGOR BORÇATTI DA SILVA

DESENVOLVIMENTO DE UM SISTEMA DE MONITORAMENTO  
DE PLANTAÇÕES

GUARULHOS

2021

PEDRO IGOR BORÇATTI DA SILVA

**DESENVOLVIMENTO DE UM SISTEMA DE MONITORAMENTO DE  
PLANTAÇÕES**

Trabalho de Conclusão de Curso apresentado como exigência parcial para obtenção do diploma do Curso Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Câmpus Guarulhos.

Orientador: Prof. Me. Cristiano Alves Pessoa

PEDRO IGOR BORÇATTI DA SILVA

**DESENVOLVIMENTO DE UM SISTEMA DE MONITORAMENTO DE  
PLANTAÇÕES**

Trabalho de Conclusão de Curso apresentado como exigência parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Câmpus Guarulhos.

Orientador: Prof. Me. Cristiano Alves Pessoa

Aprovado pela banca examinadora em 25 de novembro de 2021.

**BANCA EXAMINADORA**

---

Prof. Me. Cristiano Alves Pessoa  
IFSP Câmpus Guarulhos

---

Prof. Dr. Flávio Aparecido Pontes  
IFSP Câmpus Boituva

---

Prof. Me. Rogério Daniel Dantas  
IFSP Câmpus Guarulhos

## **AGRADECIMENTOS**

Gostaria de agradecer aos cientistas e exploradores que desbravaram nosso mundo em busca de conhecimento.

Agradeço, especialmente, a minha família, pais e irmãos pelo amparo e apoio para com meus estudos.

Gostaria de agradecer também a todo corpo docente e administrativo do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Câmpus Guarulhos, por me proporcionar uma formação de qualidade, por fazer parte de um momento importante da minha vida.

Agradeço especialmente ao professor orientador Cristiano Pessoa, pelas oportunidades e pela orientação deste trabalho, e aos professores Giovani Fonseca Ravagnani Disperati, Joel Saade e Rogério Daniel Dantas pela inestimável ajuda no decorrer da elaboração deste trabalho.

*“[...] Foi dito que a astronomia é uma experiência que traz humildade e constrói o caráter. Talvez, não haja melhor demonstração das tolices e vaidades humanas que essa imagem distante do nosso pequeno mundo. Ela enfatiza nossa responsabilidade de tratarmos melhor uns aos outros, e de preservar e estimar o único lar que nós conhecemos... o pálido ponto azul”.*

O Pálido Ponto Azul, Carl Sagan

## LISTA DE ILUSTRAÇÕES

Figura 1 - Arquitetura do FeerRTOS .....	22
Figura 2 - Diagrama do microcontrolador ES32 .....	23
Figura 3 - Fonte DCDC Step Down MP1584 .....	26
Figura 4 - Fonte de alimentação e medidores de tensão .....	27
Figura 5 - Sonda para o sensor SHT31 .....	28
Figura 6 - Estrutura de firmware de Agro Sensor .....	29
Figura 7 - Lógica de funcionamento da <i>firmware</i> .....	30
<i>Figura 8 - Diagrama de deployment</i> .....	31
<i>Figura 9 - Sequência de comunicação entre dispositivo e servidor m2m</i> .....	32
<i>Figura 10 - Diagrama de casos de uso</i> .....	34
<i>Figura 11 - Caixa do conjunto Agro Sensor e bateria</i> .....	35
Figura 12 - Teste de campo do Agro Sensor .....	35
Figura 13 - Dispositivo Agro Sensor .....	36
Figura 14 - Gráfico montado no <i>overview</i> do dia 1 ao dia 10 de novembro .....	37
Figura 15 - Esquema elétrico do Agro Sensor .....	47
Figura 17 - Tela de visão geral ( <i>overview</i> ) .....	48
Figura 16 - Diagrama de classe .....	49
Figura 18 - Tela de gerenciamento de dispositivos .....	50
Figura 19 - Tela de gerenciamento de usuários .....	51
Figura 20 - Formulário de cadastro ou atualização de dispositivos .....	52
Figura 21 - Tela de visão detalhada dos dispositivos .....	53

## **LISTA DE TABELAS**

Tabela 1 - Especificações do SHT31 .....	28
Tabela 2 - Sprints para o desenvolvimento do projeto .....	42
Tabela 3 - Especificações técnicas do painel solar, bateria e modulo .....	41

## LISTA DE SIGLAS

IoT	Internet Of Things
M2M	Machine to Machine
MQTT	Message Queuing Telemetry Transpor
RTOS	Real Time Operating System
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
SQL	Structured Query Language



## RESUMO

Este trabalho apresenta o desenvolvimento de um sistema de monitoramento remoto de plantações, composto por dispositivos que medem temperatura e umidade dos solos por meio de sondas. O sistema possui uma *dashboard* que apresenta os dados coletados e possibilita o gerenciamento dos dispositivos conectados ao mesmo. O projeto divide-se em duas partes: desenvolvimento do *software* e *hardware* de aquisição de dados através de microcontrolador ESP32 e de um sistema supervisor de serviços web que compõem a *dashboard*. O hardware desenvolvido utiliza o microcontrolador ESP32 WROOM 32E, programado com o sistema operacional embarcado FreeRTOS, o qual se comunica por meio do protocolo MQTT com um servidor do tipo *Machine to Machine* (M2M) composto por um *Broker* implementado em Node-RED, cuja finalidade é gerenciar as conexões e transmissões dos dados recebidos. Estes dados também são enviados para um serviço de banco de dados. Como resultado foi elaborada uma plataforma que permite adicionar ou remover dispositivos e monitorar o estado do funcionamento destes, bem como permitir o cadastro de usuários que podem visualizar e baixar os dados medidos pelo dispositivo.

**Palavras-chave:** Monitoramento. IoT. Agricultura 4.0. MQTT. Node-red.

## **ABSTRACT**

This work shows the development of a remote acquisition system of plantation, which is composed of devices that measure soil temperature and moisture using probes, and the development of a dashboard system that shows the data which presents the collected data and makes it possible to manage the devices connected to it. This work is divided in two parts: development of data acquisition software and hardware through ESP32 microcontroller and the development of supervisory systems. The hardware developed use the microcontroller ESP32 WROOM 32E that use the embedded system FreeRTOS, this device communicates through the MQTT protocol with the server Mach Machine to Machine (M2M) using a MQTT broker implemented with node-red whose purpose is manager the connections and the data received. This data is also being sent to the database service. The work's result was the development of a platform that can add or remove devices as well as monitor their status. It is also capable of registering users that can both see and download measurement data from devices.

**Keywords:** Monitoring. IoT. Agriculture 4.0. MQTT. Node-red.

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>13</b>
<b>2 OBJETIVOS.....</b>	<b>14</b>
2.1 Objetivo Geral.....	14
2.2 Objetivo Específicos.....	14
<b>3 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>15</b>
3.1 Internet das Coisas .....	15
3.2 Agricultura 4.0.....	16
3.3 Sistemas Operacionais Embarcados de Tempo Real.....	17
<b>4 MATERIAIS E MÉTODOS.....</b>	<b>19</b>
<b>4.1 Tecnologias Utilizadas.....</b>	<b>19</b>
4.1.1 HTML - Linguagem de marcação de texto.....	19
4.1.2 CSS - Folhas de estilo em cascata .....	19
4.1.3 JavaScript.....	19
4.1.4 PHP - Pré-processador de hipertexto .....	19
4.1.5 Banco de dados relacional MySql .....	20
4.1.6 Linguagem C.....	20
4.1.7 Transporte de telemetria de enfileiramento de mensagens.....	20
4.1.8 Codeigniter .....	21
4.1.9 Node-RED .....	21
4.1.10 Sistema operacional FreeRTOS .....	22
4.1.11 Modulo ESP-WROOM-32.....	23
4.1.12 Framework ESP-IDF .....	23
4.1.13 Docker.....	24
<b>4.2 Processo de Desenvolvimento .....</b>	<b>24</b>
<b>4.3 Verificação e Validação do Sistema.....</b>	<b>24</b>
<b>5 DESENVOLVIMENTO .....</b>	<b>26</b>
<b>5.1 Projeto Agro Sensor.....</b>	<b>26</b>
5.1.1 Alimentação.....	26
5.1.2 Sensor de temperatura e umidade.....	27
5.1.3 Outras interfaces.....	28
5.1.4 <i>Firmware</i> .....	29
<b>5.2 Serviços Web.....</b>	<b>30</b>
5.2.1 Servidor machine to machine.....	31

5.2.1 Serviço web .....	33
5.3 Teste do Sistema.....	34
6 RESULTADOS OBTIDOS .....	36
6.1 Placa Agro Sensor.....	36
6.2 Serviços <i>web</i> .....	36
7 CONSIDERAÇÕES FINAIS.....	39
REFERÊNCIAS.....	40
APÊNDICE A - Tabelas e Figuras.....	44

## 1 INTRODUÇÃO

Com o crescimento do agronegócio no Brasil, surge a necessidade de investimento em tecnologias que buscam o aprimoramento da cadeia produtiva. Essas tecnologias possibilitam o monitoramento e análise de uma variada gama de dados em tempo real, de forma a auxiliar na gestão e na tomada de decisões de forma rápida e precisa (ROSSONI et al., 2019), (COSTA et al., 2018), (BRENES et al., 2020).

Tais tecnologias empregadas no meio produtivo do agronegócio também apresentam um impacto positivo no uso de recursos naturais, diminuindo desperdício e, conseqüentemente, o custo, promovendo um ganho de eficiência no consumo de insumos. Isto é decisivo para amenizar impactos ambientais negativos e auxiliar no desenvolvimento sustentável destas produções (BRENES et al., 2020) (SOARES, 2020). (COSTA et al., 2018a).

Em 2017, a agricultura foi um dos maiores responsáveis pelo consumo de água no país e, assim como em outros lugares do mundo, a otimização e redução do consumo de recursos hídricos se mostra um desafio. Nesse âmbito, a Internet das coisas pode ser utilizada de forma a mitigar este problema. Isto foi verificado no trabalho SWAMP, um projeto piloto colaborativo entre Itália, Espanha e Brasil que buscou desenvolver um sistema de gerenciamento inteligente de irrigação utilizando métodos e conceitos da tecnologia IoT (RIBEIRO et al., 2018), (KAMIENSKI et al., 2018), (COSTA et al., 2018b) (IBGE, 2020).

A Internet das coisas, IoT, atualmente vem sendo aplicada em diversos segmentos. No setor de agronegócios isto vem sendo denominado de agricultura 4.0, denotando a forma como a tecnologia empregada tente ao controle e acesso remoto dos dispositivos e dados. Além disso, certos aspectos como conectividade, armazenamento em nuvem, autonomia energética e custo são importantes características nesse contexto (COSTA et al., 2018a), (BRENES et al., 2020), (COSTA et al., 2018b). Assim, este trabalho tem o objetivo de desenvolver um sistema de monitoramento do solo de plantações buscando explorar as dificuldades tecnológicas na implantação destes métodos de IoT aplicados a agricultura, realizando um estudo no desenvolvimento de novas alternativas para otimização do uso de recursos e melhorias de controle sobre a produção.

## **2 OBJETIVOS**

### **2.1 Objetivo Geral**

O objetivo deste trabalho é o desenvolvimento de um sistema de monitoramento remoto de plantações, composto por dispositivos que medem temperatura e umidade dos solos por meio de sondas. Também será desenvolvido um sistema de supervisão que apresente os dados coletados e possibilite o gerenciamento dos dispositivos conectados e acesso as informações.

### **2.2 Objetivo Específicos**

- Realizar levantamento bibliográfico;
- Desenvolver um dispositivo para realizar a aquisição periódica de temperatura e umidade do solo;
- Desenvolver um serviço *web* que possibilite o gerenciamento dos dispositivos e acesso as informações coletadas;
- Verificar o sistema desenvolvido através de testes de campo.

### 3 FUNDAMENTAÇÃO TEÓRICA

#### 3.1 Internet das Coisas

O termo Internet das Coisas foi usado pela primeira vez em 1999 por Kevin Ashton que dez anos depois escreveu o artigo “That ‘Internet of Things’ Thing” para o RFID Journal em 2009. Em que ele diz sobre a importância da capacidade dos computadores de coletar, armazenar, interpretar e usar dados a fim de permitir que eles “observem o mundo” sem a ajuda humana, na visão do autor isso impactaria positivamente na economia de recursos naturais e energéticos (MAGRANI, 2018), (ASHTON, 2009).

A Internet das coisas tornou-se possível após o surgimento da *web* 3.0, mas antes disso outras revoluções ocorreram até que chegássemos neste ponto. A Internet se origina no fim da década de 1930 pelo projeto militar Advanced Research Projects Agency Network (Arpanet), mas só após o surgimento de novas tecnologias como o world wide web (*www* ou *web*), criado no final da década de 1980 pelo Conseil Européen pour la Recherche Nucléaire (CERN), surge a primeira versão da *web* (MAGRANI, 2018).

A *web* 1.0 é caracterizada principalmente pela conexão entre pessoas de forma estática. Em 2004 é popularizado o termo *web* 2.0. Apesar de não ser claro quando esta surgiu, a *web* 2.0 é definida por ser colaborativa e dinâmica, onde os usuários contribuem gerando conteúdo graças às redes sociais, *wikis* e outras plataformas. A *web* 3.0, termo criado a partir do anterior, é marcado pela conectividade, redes integradas e descentralizadas, *web* inteligente, internet pessoal entre outras características (MAGRANI, 2018).

É nesse contexto e graças às tecnologias de rádio que permitem a conexão entre objetos como Wi-Fi e Bluetooth que surge a Internet das coisas, expressão que retrata dispositivos que fazem parte de um sistema e estão conectados à internet, interagindo no dia a dia com pessoas gerando dados sendo e sendo usados de forma inteligente (MAGRANI, 2018), (ASHTON, 2009).

### 3.2 Agricultura 4.0

A agricultura no início século vinte era caracterizada pela sua baixa produção, sendo realizada de forma manual utilizando tração animal e ferramentas artesanais. Além disso, era encarada como uma atividade de subsistência. Esses fatores definem a agricultura 1.0 que, apesar de ainda ser empregada, perdeu espaço ao decorrer do tempo (MASSRUHÁ *et al.*, 2020), (SILVA *et al.*, 2021).

A agricultura 2.0 é estabelecida após diversos avanços científicos e inovações tecnológicas, como motores a combustão que substituíram o uso de animais na produção, e uso de produtos industrializados como defensivo agrícola e técnicas de fertilização que contribuíram para o aumento da produção. A agricultura 2.0 também é marcada pela comercialização e produção em escala (MASSRUHÁ *et al.*, 2020), (SILVA *et al.*, 2021).

Já, a agricultura 3.0 destaca-se pela união dos avanços tecnológicos com o emprego de estudos adicionais focados nas melhoras produtivas, permitidos por tais avanços. Há, assim, o objetivo de se atingir maior precisão nos resultados. Isso se dá, por exemplo, com a utilização de técnicas de georreferenciamento por satélite, características atribuídas à agricultura de precisão. Estes fatores permitiram maior desempenho e precisão da produção, além da otimização do uso de recursos utilizados nos processos de adubação e irrigação (MASSRUHÁ *et al.*, 2020), (SILVA *et al.*, 2021).

Agricultura 4.0, por sua vez, é uma terminologia que remete a aplicação de métodos e tecnologias na produção agrícola onde há união do ambiente virtual e físico, fazendo o uso de redes que conectam máquinas, atuadores e sensores que fornecem dados para sistemas e ferramentas na Internet. Há assim o uso da computação, armazenamento em nuvem e métodos analíticos para auxílio na tomada de decisões. Em outras palavras, a agricultura 4.0 faz uso dos mesmos conceitos aplicados na Internet das coisas, o que permite um uso ainda mais otimizado e inteligente dos recursos necessários para os processos produtivos (MASSRUHÁ *et al.*, 2020), (MASSRUHÁ *et al.*, 2017), (SILVA *et al.*, 2021), (FRANÇA, 2021).



### 3.3 Sistemas Operacionais Embarcados de Tempo Real

Sistema embarcado é um conjunto de *software* e *hardware* que desempenha uma função ou grupo de funções com uma finalidade específica, sendo projetado especialmente para isso. Isto resulta, portanto, em uma grande eficiência nessas atividades (WANG, 2017). (PUHAN, 2015).

Geralmente o *hardware* desses dispositivos usa componentes eletrônicos escolhidos de acordo com papel que será executado. Um exemplo disto seriam microcontroladores ou microprocessadores que possuem recursos limitados em comparação a um computador comum, porém apresentam vantagens tais como baixo consumo energético, ocupam pouco espaço, dispensam grande quantidade de periféricos adicionais para seu funcionamento e, em alguns casos, contam com recursos específicos embarcados no próprio *hardware* que permitem que esses componentes tenham um desempenho maior para atividades específicas. Bons exemplos de sistemas embarcados incluem relógios digitais, sensores e até mesmo *smartphones* (WANG, 2017), (PUHAN, 2015), (LIMA *et al.*, 2012).

Estes equipamentos sempre contam com um *software* que, assim como o *hardware*, é otimizado e ou especializado para realizar uma tarefa específica. Tais *softwares* podem ser desenvolvidos de duas maneiras: 1) *bare-metal*, quando o programa lida diretamente com o *hardware*, método que costuma ser otimizado e consumir poucos recursos, porém sua complexidade é elevada; 2) utilizando sistemas operacionais voltados para ambientes embarcados, que apresentam uma série de vantagens, tais como desenvolvimento facilitado em comparação a alternativa *bare-metal*, portabilidade e abstração do *hardware* (WANG, 2017), (PUHAN, 2015), (LIMA *et al.*, 2012).

Um sistema operacional é um grupo de *softwares* que compõem um ambiente multitarefa, o qual abstrai o *hardware* e o gerenciamento de seus recursos de outras aplicações. Para o RTOS o tempo tem grande importância, de forma que a quantidade de tarefas sendo executadas paralelamente é menos relevante que o tempo de resposta e a capacidade de determinar a duração destas tarefas. Esses sistemas são compostos basicamente por três partes: o Kernel, o Scheduler e Tasks (WANG, 2017), (PUHAN, 2015).

- **Kernel:** é a camada de abstração entre *hardware* e *software*, que gerencia os recursos do dispositivo como memória RAM, interfaces de entrada e saída, entre outros periféricos (WANG, 2017), (PUHAN, 2015).
- **Scheduler:** *software* responsável pelo gerenciamento do uso da CPU, decidindo quais *tasks* serão executadas, geralmente considerando as prioridades de cada uma. O *scheduler* pode operar de forma preemptiva, quando é permitido que a execução de uma *task* seja interrompida para atender outras *tasks* ou interrupções; ou de forma cooperativa, quando não é permitido a interrupção de uma *task* até o final de sua execução (WANG, 2017), (PUHAN, 2015).
- **Tasks:** são pequenos programas que executam funções específicas, sejam elas do próprio sistema ou de uma aplicação. Eles podem assumir quatro estados: 1) execução, quando uma *task* está alocada na CPU; 2) suspensão, quando a *task* está desativada, e neste caso ela não é escalonada para ser executada; 3) bloqueada, quando a *task* está aguardando um evento ou que recurso seja liberado; 4) pronta, quando a *task* está pronta para ser executada e não está suspensa e nem bloqueada (WANG, 2017), (PUHAN, 2015).

## 4 MATERIAIS E MÉTODOS

Neste capítulo será discutido o método de desenvolvimento e validação do projeto, assim como algumas das principais tecnologias utilizadas nos desenvolvimentos e a forma como foram usadas.

### 4.1 Tecnologias Utilizadas

Neste tópico serão apresentadas as principais tecnologias como linguagens e *frameworks* utilizados no desenvolvimento deste projeto.

#### 4.1.1 HTML - Linguagem de marcação de texto

O HyperText Markup Language (HTML) é uma linguagem de marcação de texto estática criada pelo físico Tim Berners-Lee na década de 90 no CERN, essa linguagem tem o objetivo de facilitar o compartilhamento de informações pela internet, permitindo organizar os textos e criar links e cabeçalhos (BALLERINI, 2021), (TOTVS, 2021).

#### 4.1.2 CSS - Folhas de estilo em cascata

Criada como uma linguagem complementar ao HTML, o Cascading Style Sheets (CSS), foi criado para melhorar a forma que o *design* das páginas *web* são feitos, tornando o processo de desenvolvimento e o entendimento dos códigos de marcação mais simples. O CSS permite mudar características como fontes, layout, cores e outros aspectos de uma página *web* (BALLERINI, 2021), (PEREIRA, 2009).

#### 4.1.3 JavaScript

Originalmente criado por Brendan Eich para rodar no Netscape Navigator, o JavaScript tem como proposta tornar as páginas *webs* mais dinâmicas. Atualmente, além de trazer mais dinamismo para *websites*, o JavaScript pode ser utilizado para criar aplicativos para diversos ambientes como *desktop* e *mobile* (BALLERINI, 2021), (SILVA, 2015).

#### 4.1.4 PHP - Pré-processador de hipertexto

Hypertext Preprocessor (PHP) é uma linguagem que foi desenvolvida em 1994 por Rasmus Lerdorf. O PHP é uma linguagem *open source* de tipagem dinâmica e interpretada. Apesar de ser de uso geral, ela é muito popular no desenvolvimento *web*,

possibilitando maior dinamismo nas páginas *web* e podendo ser embutida junto ao HTML. Diferentemente do JavaScript, o PHP roda do lado do servidor, de forma que o cliente tem acesso apenas o HTML renderizado da página (PHP, 2019).

#### 4.1.5 Banco de dados relacional MySql

O MySql é um banco de dados de dados relacional de código aberto muito popular no mercado, sendo utilizado por grandes empresas como YouTube e Facebook. Assim como outros bancos relacionais ele utiliza a linguagem *Standard Query Language* (SQL), que é uma linguagem declarativa usada para realizar consultas. (SILVEIRA, 2019).

#### 4.1.6 Linguagem C

A linguagem C foi criada por Dennis Ritchie em 1972 para o sistema operacional Unix e se trata de uma programação multiplataforma, compilada e fortemente tipada. Com ela podemos ter acesso de baixo nível ao *hardware* além de poder trabalhar junto do Assembler, dessa forma sendo muito popular em microcontroladores (SCHILDT, 1995).

#### 4.1.7 Transporte de telemetria de enfileiramento de mensagens

Message Queue Telemetry Transport (MQTT) é um protocolo de comunicação M2M desenvolvido pelos engenheiros Andy Stanford-Clark e Arlen Nipper em 1999, com o objetivo de transmitir a telemetria de oleodutos por meio de satélites. Este permite envio de mensagens e comandos por meio do protocolo TCP/IP com baixo consumo de recursos de *hardware* e uma estrutura simples, o que o tornou um dos protocolos mais utilizados em aplicações IoT atualmente (HIVEMQ, 2020).

O MQTT utiliza o conceito de *publish subscribe*, onde um dispositivo publica (*publish*) um tópico em um *broker* e o cliente assina esse tópico (*subscribe*), assim toda vez que uma publicação de um tópico é feita o *broker* envia essa informação para o cliente. Desta forma, o *broker* é o intermediário entre quem está publicando e quem está assinando. Para o cliente enviar um comando para um dispositivo o mesmo processo é adotado (HIVEMQ, 2020).

#### 4.1.8 Codeigniter

*Codeigniter* é um *framework* multiplataforma de código aberto para desenvolvimento web, com uma estrutura simples que fornece ferramentas que facilitam e agilizam o desenvolvimento de software. Ele utiliza a arquitetura *Model*, *View* e *Controller* (MVC), o qual é constituído por pelo menos 3 camadas, responsáveis por dividir o código entre responsabilidades específicas: 1) *Model*, encarregado das regras de negócio e manipulação e acesso dos dados; 2) *View*, camada que mostra as informações do *Model* para o usuário, portanto responsável pela interface; 3) *Controller*, liga o *model* e *view* com a finalidade passar as informações do *model* para o *view* (ANDRADE, 2021), (GUEDES, 2020).

Além da afinidade os atributos desse *framework* foram decisivos para sua escolha no projeto, sendo empregado no desenvolvimento da aplicação web encarregada entre a comunicação humano computador, e gerenciamento de dispositivos (PHP).

#### 4.1.9 Node-RED

O Node-RED teve início em 2013 na IBM por Nick O'Leary e Dave Conway-Jones no grupo de serviços de tecnologia emergentes, que desde outubro de 2016 faz parte da Fundação JS. Criado em JavaScript o Node-RED foi desenvolvido inicialmente como prova de conceito para manipular e visualizar o mapeamento entre tópicos do protocolo de comunicação Message Queuing Telemetry Transport (MQTT), porém, mostrou ser uma ferramenta muito mais geral podendo ser utilizado em diversos outros contextos, sobretudo em aplicações de automação e Internet das coisas (OPENJS FOUNDATION, 2021), (ENVISIA, 2019).

Por ser uma ferramenta de programação baseada em fluxo seu uso é simplificado, permitindo um desenvolvimento fácil e rápido. Além disso o Node-RED permite integração de inúmeras API's (OPENJS FOUNDATION, 2021), (ENVISIA, 2019).

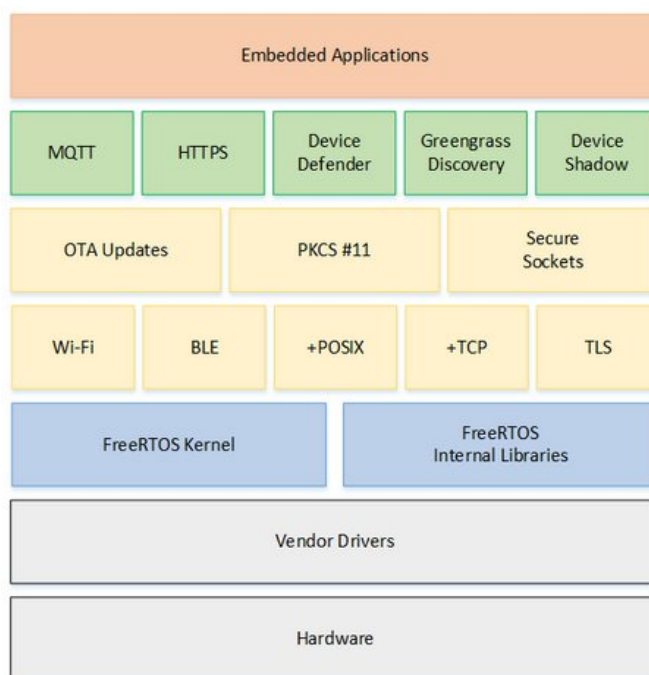
Por essas características essa tecnologia foi selecionada para esse projeto e aplicada com foco na comunicação *machine to machine* (M2M), especialmente entre os dispositivos e os servidores (NODE-RED).

#### 4.1.10 Sistema operacional FreeRTOS

O FreeRTOS é um sistema operacional de tempo real de código aberto atualmente mantido pela Amazon Web Services e parceiros, desenvolvido para sistemas embarcados, sendo utilizado em diversos microcontroladores e microprocessadores. Por este motivo, os principais apoiadores do projeto são empresas que desenvolvem e produzem microchips (BARRY, 2016), (AMAZON WEB SERVICES, 2021).

O FreeRTOS contém inúmeras bibliotecas de conectividade e segurança, além de recursos de gerenciamento do próprio sistema e periféricos. Também permite portabilidade entre diversos microcontroladores. Outra característica importante deste sistema é a arquitetura modularizada como mostrado na Figura 1, possibilitando que o desenvolvedor utilize apenas o essencial para a sua aplicação, assim otimizando o uso dos recursos dos microchips (BARRY, 2016), (AMAZON WEB SERVICES, 2021).

Figura 1 - Arquitetura do FreeRTOS

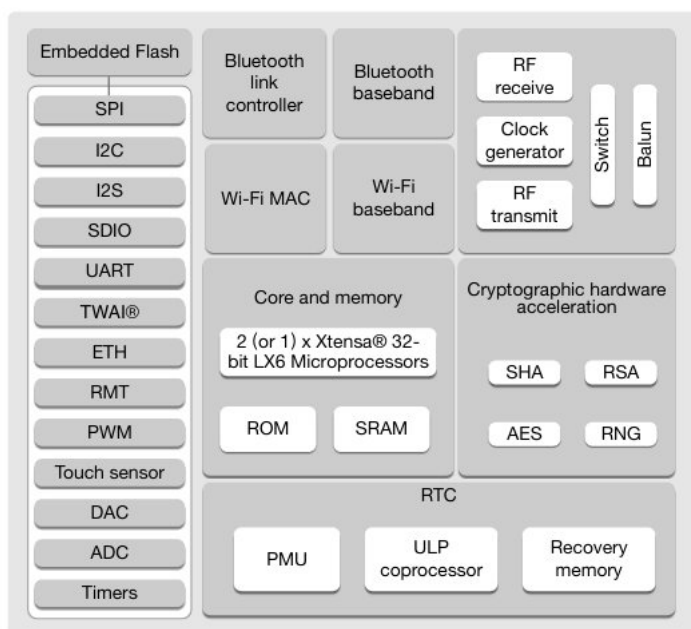


Fonte: Amazon Web Services (2021).

#### 4.1.11 Módulo ESP-WROOM-32

O ESP-32 é um módulo produzido pela Espressif especialmente projetado para aplicações que envolvam internet das coisas, pois além dos periféricos mais clássicos encontrados em outros controladores, como comunicação serial RS232, ele conta com periféricos de rádio e hardware dedicado para trabalhar com criptografia e comunicação *Wi-Fi* ou *bluetooth*, como mostra a Figura 2. O módulo ainda tem uma unidade lógica de processamento dual core que pode operar até 240 MHz, bem como um coprocessador de baixo consumo. Com estas especificações é possível utilizar esse chip em projetos que demandam alto desempenho e baixo consumo energético. (ESPRESSIF, 2021).

Figura 2 - Diagrama do microcontrolador ES32



Fonte: ESPRESSIFE (2021).

#### 4.1.12 Framework ESP-IDF

Para programar o módulo ESP32 a Espressif disponibiliza o *framework* ESP-IDF que utiliza o sistema operacional de tempo real FreeRTOS, em uma versão portada para o módulo da série ESP-32 que utiliza um *fork* do *kernel* original do sistema, preparado para trabalhar em ambiente multi core. Também o *framework* conta com uma série de *drivers* que dão acesso aos periféricos do microcontrolador, bem como diversos módulos já desenvolvidos tais como sistema de arquivos, comunicação MQTT, entre outros.

#### 4.1.13 Docker

O Docker é uma ferramenta de containerização do Linux utilizada no desenvolvimento de diversas aplicações, que as executa de forma independente e isolada compartilhando a mesma infraestrutura. Diferentemente de uma máquina virtual, o Docker usa os recursos de *hardware* e *kernel* do hospedeiro diretamente, o que o torna mais leve (DOCKER, 2020), (RED HAT, 2020).

Assim como outras ferramentas de contêineres, o Docker permite criar imagens dos contêineres. Imagens são arquivos que contém os softwares e configuração da infraestrutura destes ambientes, sejam eles de desenvolvimento ou de produção. Isto facilita o compartilhamento destas imagens entre os desenvolvedores e garante que não ocorrerão problemas de compatibilidade entre os distintos ambientes em que a aplicação está rodando (DOCKER, 2020) (RED HAT, 2020).

### 4.2 Processo de Desenvolvimento

Como método de desenvolvimento deste projeto optou-se pelo modo iterativo e incremental. Nele o ciclo de vida consiste em várias interações que incorporam uma sequência de tarefas. A cada interação o sistema é criado adicionando funções ao decorrer de todo processo, de forma que o projeto possa ser apresentado de forma parcial antes do término, permitindo assim correções e melhorias.

O sistema está dividido em duas partes sendo elas divididas em sub partes, a primeira parte denominada de “Agro Sensor” que se refere ao desenvolvimento do dispositivo que coleta e transmite os dados; e a segunda parte são os serviços web responsares por receber, armazenar e mostrar os dados coletados pelos Agro Sensores. A tabela 2 do apêndice apresenta as *sprints* de todo o projeto, relacionando as duas partes às respectivas subpartes.

### 4.3 Verificação e Validação do Sistema

A validação do sistema busca medir de forma qualitativa o funcionamento em um longo período de tempo do projeto desenvolvido neste trabalho.

Para verificação e validação do sistema, foi realizada uma bateria de testes funcionais conduzidos manualmente, de forma a atestar o funcionamento do hardware e dos requisitos desenvolvidos.



Foi verificado, dessa forma, que o sistema encontra-se funcionando sem erros que comprometam a aquisição e armazenamento dos dados, e em um ambiente que se aproxima do real.

## 5 DESENVOLVIMENTO

Nesse capítulo serão descritas algumas das principais partes do desenvolvimento do projeto de maneira mais detalhada.

### 5.1 Projeto Agro Sensor

#### 5.1.1 Alimentação

O objetivo primário do Agro Sensor é realizar medidas de temperatura e umidade de forma remota, dessa forma o dispositivo muitas vezes não estará próximo de uma fonte de alimentação. Portanto, é necessário utilizar uma fonte que comporte um painel solar e uma bateria recarregável.

A fonte escolhida para isso foi a MP1584, mostrada na Figura 3, que se trata de um conversor de tensão contínua do tipo *Step Down*, com capacidade de entrada de 4.4V a 28V e saída ajustável 0.8V a 20V, além de corrente de 3A (MPS, 2011) possibilitando operar com painéis de solares de baixa potência. A saída regulável também permite que o equipamento possa operar com diversas baterias.

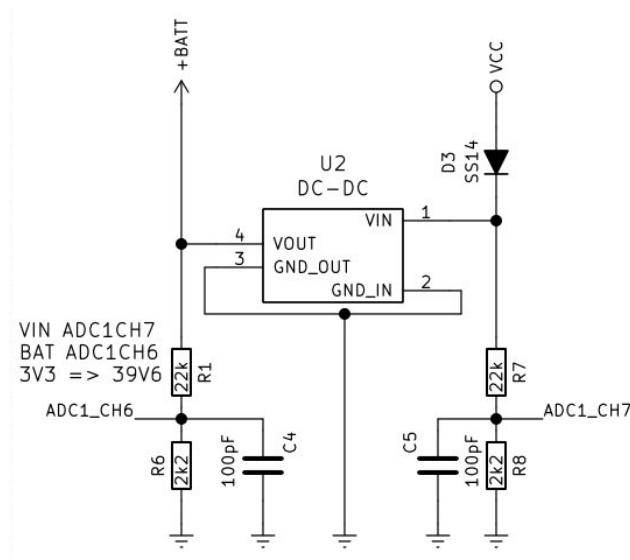
Figura 3 - Fonte DCDC Step Down MP1584



Fonte: Elaborado pelo autor.

O circuito da fonte conta com duas entradas, sendo elas o VCC, que recebe a alimentação do painel solar, e o VBAT, dedicado à alimentação da bateria. Ambos VCC e VBAT são valores que precisam ser monitorados, uma vez que este se trata de um passo fundamental para manutenção do dispositivo e para a troca de bateria. Para isto o circuito conta com dois divisores de tensão, conectados a conversores analógicos digitais do módulo ESP-32, como mostra a Figura 4. Estes conversores permitem que uma grandeza física seja convertida para um valor digital, a fim de que este possa ser processado pelo *firmware* posteriormente.

Figura 4 - Fonte de alimentação e medidores de tensão



Fonte: Elaborado pelo autor.

Na configuração montada as entradas podem medir no mínimo 0V até, no máximo, 39.6V. Tal capacidade de medida é definida em função da tolerância do conversor analógico digital do módulo, sendo ela de 3.3V (ESPRESSIF, 2021), bem como pela tensão máxima de operação da fonte.

### 5.1.2 Sensor de temperatura e umidade.

O tipo de sensor utilizado é um fator essencial para o objetivo proposto. Neste sentido, a possibilidade de atualização e de troca dos sensores proporciona maior flexibilidade, uma vez que facilita a manutenção e permite que sejam realizadas medidas de outras grandezas e em condições diferentes.

Para que o Agro Sensor seja compatível com sensores diversos, de forma que a única alteração seja a nível de software, é necessário utilizar comunicação serial para transmitir dados ao módulo ESP-32. Por isso optou-se pela comunicação serial de barramento I2C, que utiliza apenas duas vias, sendo uma para *clock* e outra para dados, além de possibilitar a conexão de diversos tipos de sensores no mesmo barramento (MENDONÇA, 2018).

O sensor utilizado no Agro Sensor é o SHT31 que mede temperatura e umidade relativa do ar, como mostrado na Tabela 1. Suas especificações técnicas atendem à demanda do projeto, uma vez que não há necessidade de medir temperaturas demasiadamente elevadas ou baixas.

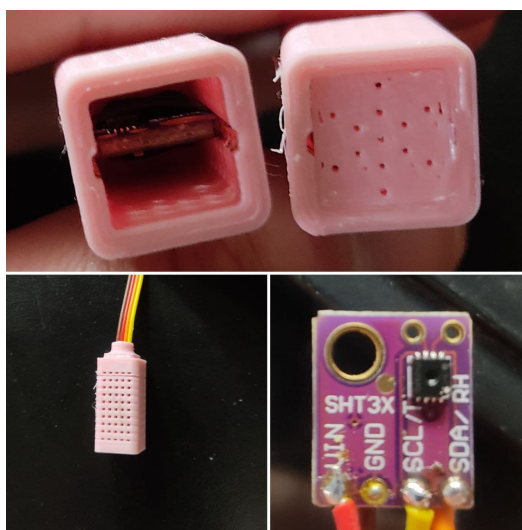
Tabela 1 - Especificações do SHT31

	Temperatura	Umidade	Alimentação
Acurácia:	0.2° C	1.5% RH	X
Mínimo:	-40° C	0% RH	2.4V
Máximo:	125° C	100% RH	5.5V

Fonte: ( SENSIRION, 2016).

O Sensor SHT 31 não pode ser enterrado diretamente no solo, uma vez que os elementos químicos presentes na terra junto à umidade podem danificar o sensor e até mesmo ocasionar curto circuito, causando danos catastróficos ao dispositivo. Por este motivo foi necessário a criar um encapsulamento, transformando o sensor em uma sonda garantindo proteção ao sensor ao ser enterrado. A Figura 5 mostra a sonda construída para essa finalidade.

Figura 5 - Sonda para o sensor SHT31



Fonte: Elaborado pelo autor.

### 5.1.3 Outras interfaces

Além dos sensores o dispositivo conta com outras interfaces de entrada e saída: 1) Relé para controle de elementos de maior potência; 2) Uma entrada digital para trabalhar com botões; 3) Um barramento de expansão modular para agregar funções futuras que necessitem de hardware adicional. Estes periféricos possibilitam maior flexibilidade para o Agro Sensor. Com eles há inúmeras configurações e

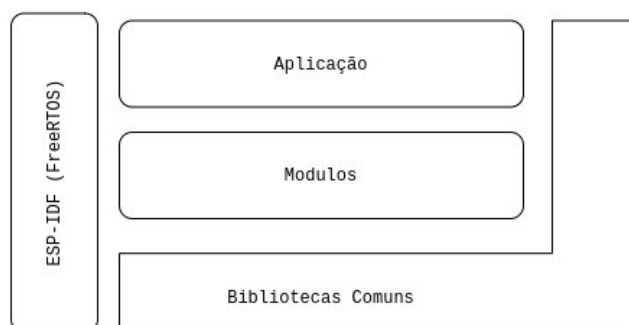
aplicações em que o projeto possa ser implementado. A Figura 15, localizada em apêndice, apresenta em maior detalhe as demais interfaces.

#### 5.1.4 Firmware

O *firmware* do dispositivo está construído com linguagem C empregando o *framework* ESP-IDF que usa o sistema operacional de tempo real FreeRTOS como base. Aproveitando os *drivers* disponibilizados pela ferramenta de desenvolvimento, outra camada de abstração voltada para os recursos utilizados pela placa foi construída, para permitir uso em ambientes *multi core* e *multi tasks*. Essa abstração é denominada de *commum*, composta por um grupo de bibliotecas comum a todo o sistema.

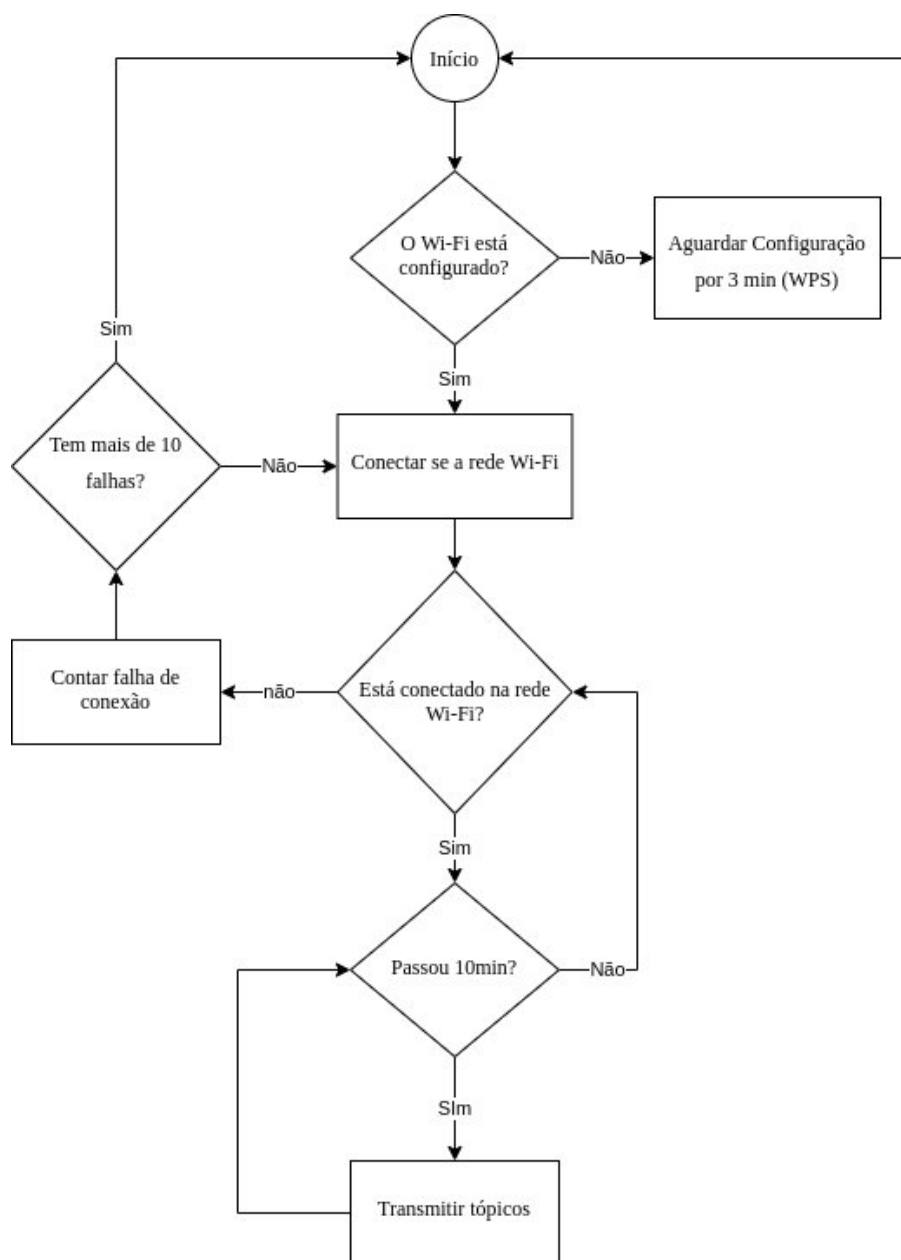
Junto a isso foram construídos módulos constituídos por *tasks* independentes, dedicadas a executar uma ou mais funções específicas, facilitando o desenvolvimento da lógica principal. A Figura 6 mostra a relação de dependências do sistema.

Figura 6 - Estrutura de firmware de Agro Sensor



Fonte: Elaborado pelo autor.

A lógica da aplicação está construída em cima de tudo isso, sendo ela responsável por manter a placa conectada na rede Wi-Fi e transmitindo dados a cada 10 minutos, a Figura 7 representa o fluxo de como essa lógica funciona.

Figura 7 - Lógica de funcionamento da *firmware*

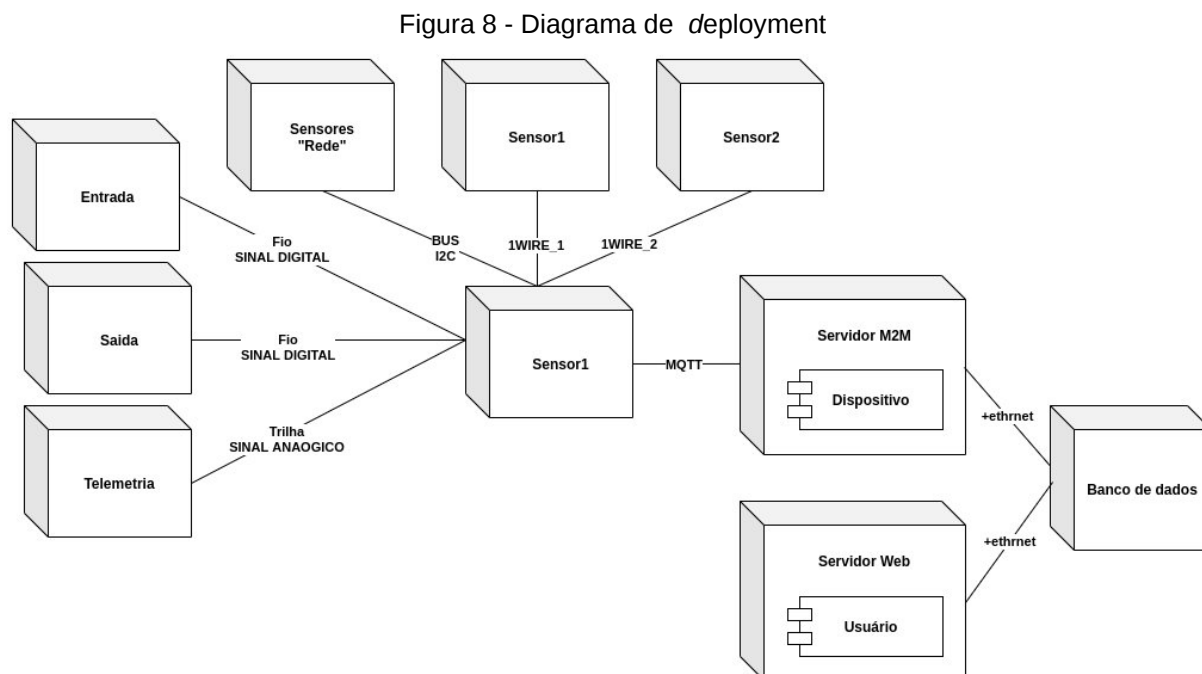
Fonte: Elaborado pelo autor.

## 5.2 Serviços Web

Como infraestrutura para o sistema foi utilizado o serviço de nuvem da Oracle Cloud na modalidade Always Free para hospedar os serviços.

Todos os sistemas encontram-se divididos em uma topologia modular a fim de facilitar a manutenção e o processo de desenvolvimento, dessa forma os serviços foram virtualizados em contêineres separados utilizando a tecnologia Docker para isso. No entanto, todos encontram-se na mesma instância. Isso se dá por conta da

modalidade utilizada para hospedagem que contém limitações, por ser gratuita. A Figura 8 apresenta a arquitetura elaborada para o sistema.



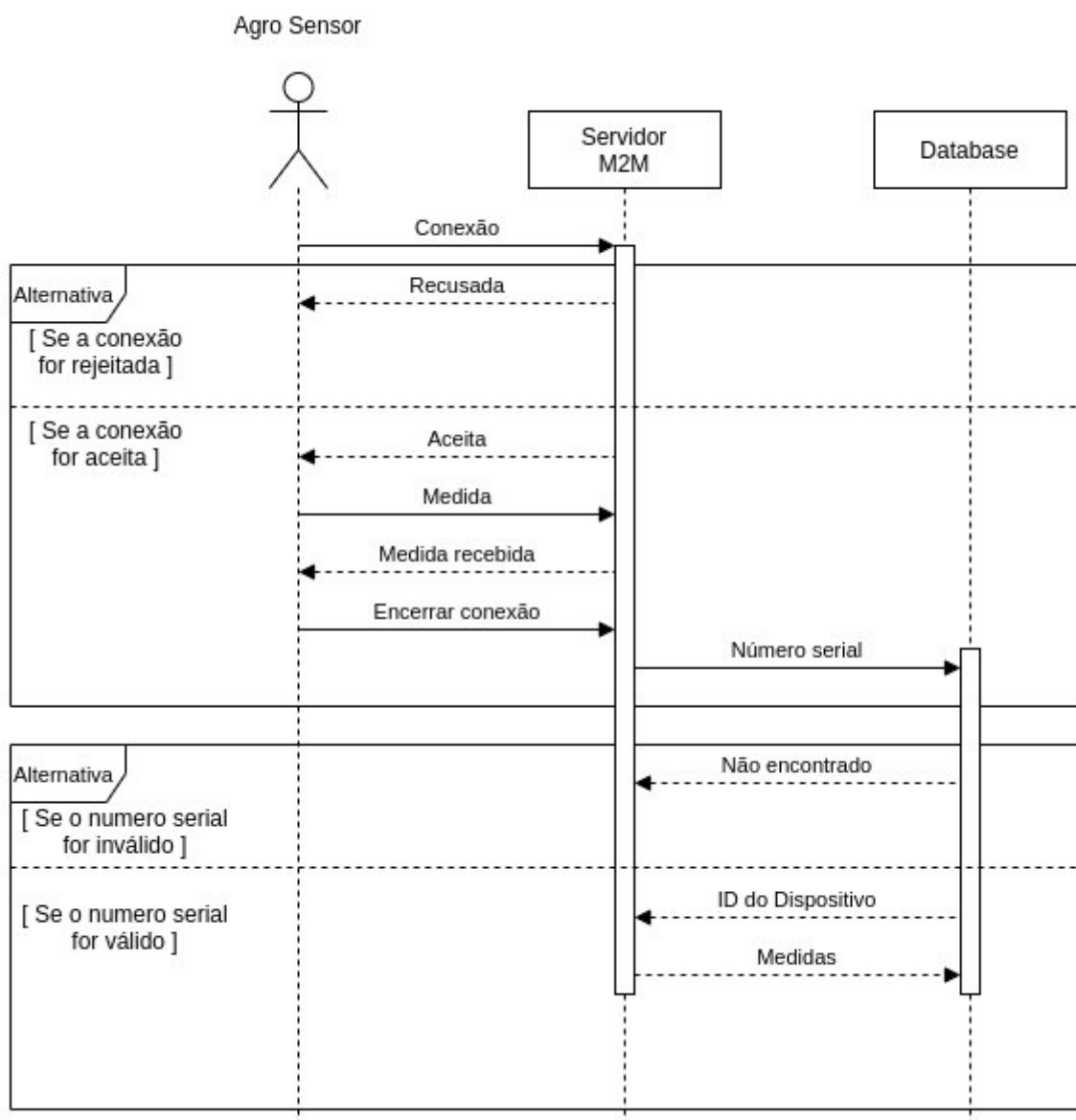
Fonte: Elaborado pelo autor.

### 5.2.1 Servidor machine to machine

O servidor M2M é responsável pela comunicação com dispositivo físico, repassando as informações coletadas para os demais serviços. Nele é preciso lidar com o protocolo MQTT e com o banco de dados ao mesmo tempo. A tecnologia adequada para isso é o *framework* Node-RED, pois o esmo foi inicialmente desenvolvido para essa finalidade, tendo assim inúmeros recursos que facilitam o processo de desenvolvimento.

Para que o dispositivo possa transmitir os dados ao sistema que posteriormente serão salvos no banco de dados o servidor contempla um CRUD junto de um *broker* MQTT. A Figura 9 do apêndice apresenta como é o processo de comunicação entre o Agro Sensor, servidor M2M e o serviço de banco de dados.

Figura 9 - Sequência de comunicação entre dispositivo e servidor m2m



Fonte: Elaborado pelo autor.

Utilizando o próprio *broker*, o servidor assina todos os tópicos disponíveis, sendo eles umidade, temperatura, tensão de entrada e tensão da bateria. Assim, quando qualquer dispositivo transmite um dado o servidor recebe essa informação e, dessa maneira, o tópico é construído de forma que permita o sistema receber informações de todos os sensores e posteriormente identificá-los isoladamente, para que possam ser salvos no banco, como mostrado a seguir:

- `mqtt://m2m:1883/[...]/codigoSerial/temperature`
- `mqtt://m2m:1883/[...]/codigoSerial/humidity`



- mqtt://m2m:1883/[...]/codigoSerial/vcc
- mqtt://m2m:1883/[...]/codigoSerial/vbat

O código serial identifica um único dispositivo, essa informação é utilizada para decidir se os dados serão salvos ou não. Além disso, os dispositivos possuem quatro possibilidades de *status* de conexão que utiliza as informações recebidas:

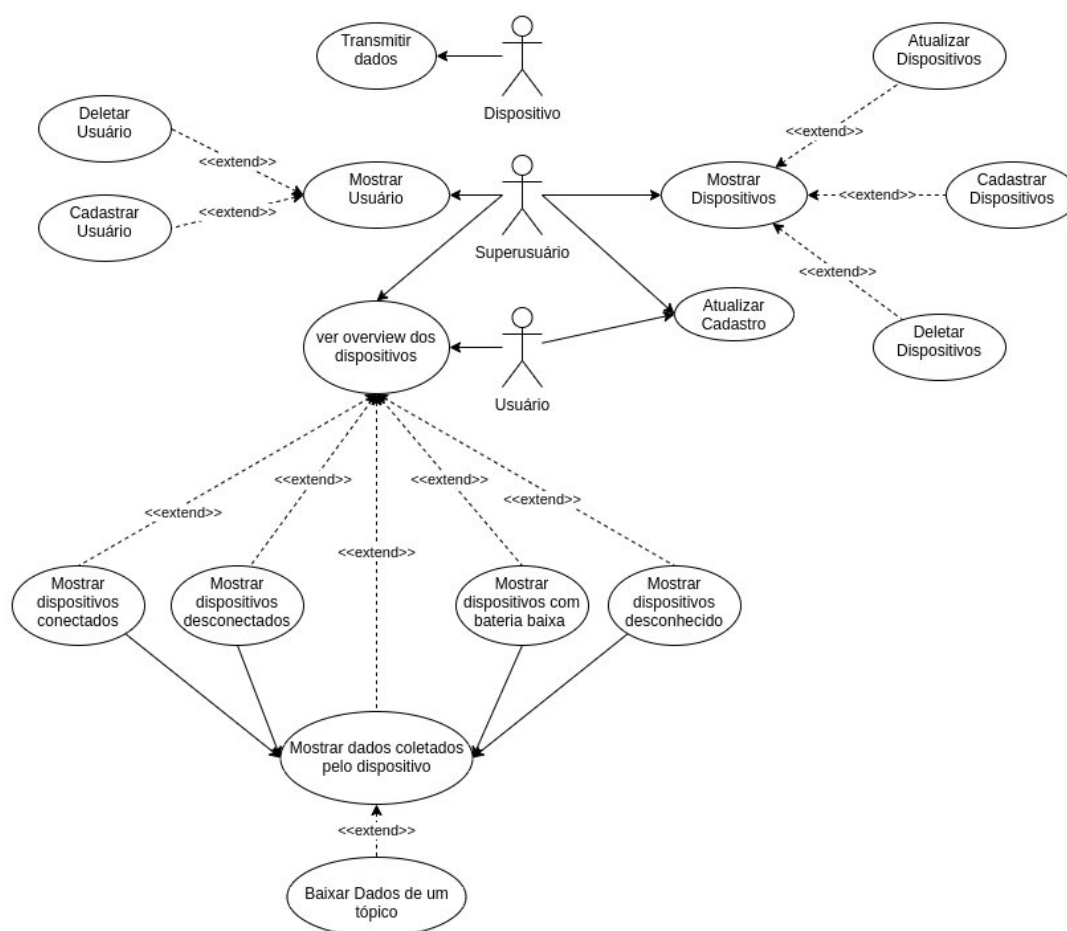
1. **Desconhecido:** quando um dispositivo é cadastrado na plataforma, por padrão ele começa com esse status que persiste até que o sensor transmite alguma informação para o servidor.
2. **Conectado:** Ao receber um dado o dispositivo passa para esse *status*, que persiste por uma hora após o último envio.
3. **Desconectado:** Caso passe um período de uma hora sem que o dispositivo transmita alguma informação o servidor altera o *status* do Agro Sensor para desconectado.
4. **Alimentação baixa:** Ocorre quando o sensor transmite uma telemetria com a tensão da bateria menor que 4.5 volts. Isso significa que o dispositivo está na iminência de perder a conexão com o sistema.

### 5.2.1 Serviço web

O serviço *web* permite que o usuário tenha acesso aos dados coletados pelos sensores e possa gerenciar os dispositivos conectados, dessa forma sendo responsável pela interface homem máquina. Para o desenvolvimento web foi utilizado o *framework* PHP Codeigniter que trabalha na arquitetura MVC e, assim como os outros componentes do sistema, este também se encontra virtualizado em um contêiner separado.

Dentre os recursos implementados, estão as funções de gerenciamento para adicionar e remover dispositivos ou outros usuários. Estas funções estão disponíveis apenas para o superusuário, consequentemente o controle do ambiente é centralizado. Porém o uso principal do sistema é a visualização dos sensores e dados respectivos, permitindo fácil acesso a qualquer dispositivo cadastrado, e para isso um sistema de busca que foi criado, podendo localizar o sensor por status ou algum termo comum. A Figura 10 mostra os demais casos de uso implementados no serviço web.

Figura 10 - Diagrama de casos de uso



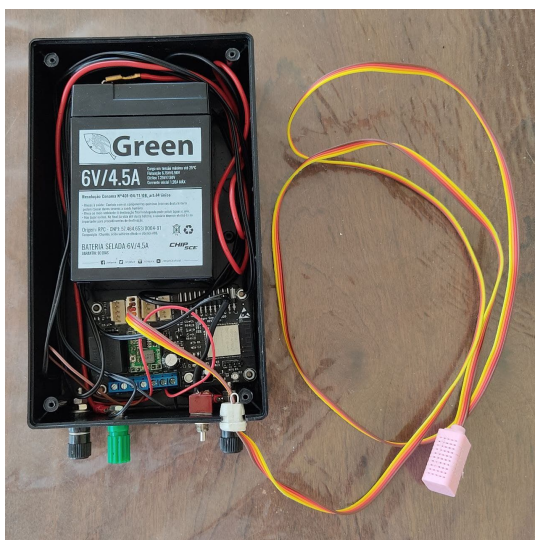
Fonte: Elaborado pelo autor.

### 5.3 Teste do Sistema

Para realizar o teste do sistema uma estrutura física foi construída, sendo composta por uma caixa para proteger conjunto do Agro Sensor e bateria mostrado na Figura 11, e uma célula fotovoltaica de 5W. O equipamento foi colocado longe de estruturas e árvores de forma que o painel solar não sofra com interferência de sombras ao decorrer dos dias, e a sonda foi enterrada próximo ao tronco de uma muda de jabuticabeira como apresentada na Figura 12.

O teste teve início no dia 30 de outubro de 2021, sem tempo previsto para término. Porém, só com a bateria e desconsiderando perdas o dispositivo pode operar por 7 meses e meio, sendo que na maior parte do tempo o painel solar fornece energia carregando a bateria em condições ótimas. O objetivo do teste é de investigar possíveis falhas e melhorias.

Figura 11 - Caixa do conjunto Agro Sensor e bateria



Fonte: Elaborado pelo autor.

Figura 12 - Teste de campo do Agro Sensor



Fonte: Elaborado pelo autor.

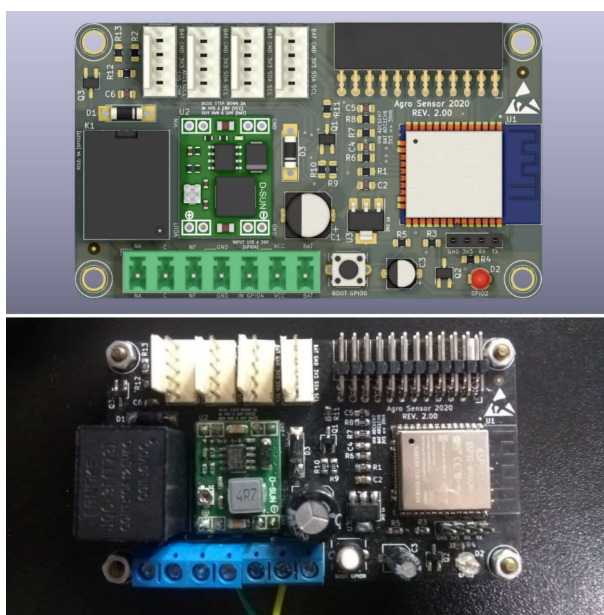
## 6 RESULTADOS OBTIDOS

### 6.1 Placa Agro Sensor

O circuito foi projetado garantir uma flexibilidade maior, podendo atuar em diferentes cenários, entretanto, no decorrer do projeto notou-se que esses recursos tornaram o dispositivo mais caro sendo inviável em certas aplicações, por exemplo em cenários com necessidade de uma grande quantidade de sensores, onde só seriam utilizados para realizar medidas, junto disso tanto o *software* do Agro Sensor como o dos Serviços *web* se tornaram mais complexo.

Por esses motivos, e apesar do hardware produzido conter outros elementos, optou-se por desenvolver uma *firmware* que embarcasse apenas as funcionalidades utilizadas pelo sistema. Foi observado que para a aplicação proposta esse conjunto de requisitos desenvolvidos foi suficiente para atender o sistema e demonstrar seu funcionamento. A Figura 13 mostra a comparação entre o projeto desenvolvido em CAD e o protótipo produzido.

Figura 13 - Dispositivo Agro Sensor



Fonte: Elaborado pelo autor.

### 6.2 Serviços *web*

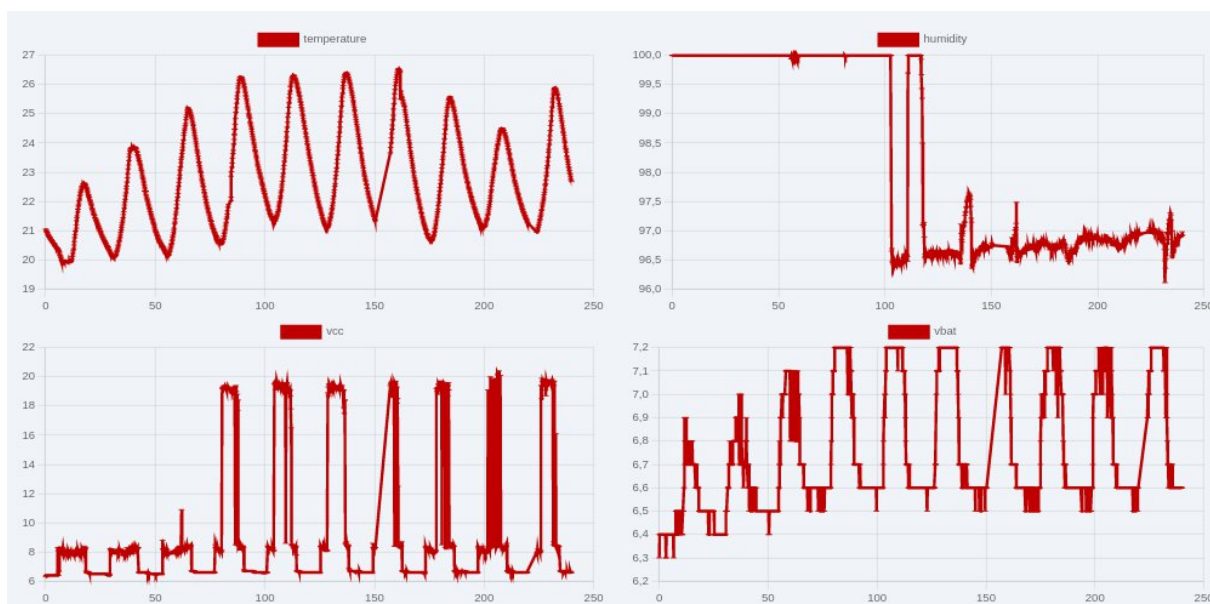
As Figuras 17 a 20 do apêndice mostram algumas telas desenvolvidas no decorrer do projeto. Em páginas como “gerenciar de usuários” (Figura 19) e “gerenciar de dispositivos” (Figura 18), assim como formulários foram reutilizados alterando



alguns detalhes de *layout* e cores, isso foi feito para simplificar o desenvolvimento e padronizar o visual da página.

Outro ponto a ser observado são os gráficos gerados na página de visão geral, onde é possível visualizar os dados de um único dia por vez ou grupo de dias, no entanto esse modo de visualizar os dados nem sempre é o ideal, é nítido que em alguns casos o gráfico gerado apresenta uma forma “ruidosa” isso ocorre por conta do tamanho da imagem e escala, como demonstrado na Figura 14.

Figura 14 - Gráfico montado no *overview* do dia 1 ao dia 10 de novembro



Fonte: Elaborado pelo autor.

O método de comunicação entre o dispositivo e servidor M2M mostrou-se muito positivo, consumindo poucos recursos do sistema embarcado e do servidor, além de facilitar o gerenciamento, demonstrando eficiência do protocolo MQTT.

A forma que a página *Overview* mantém os dados atualizados mostrou se ineficiente dada a arquitetura do projeto, no caso a cada 10 segundos um código JavaScript realiza uma consulta ao banco, trazendo todos os dados novamente e consequentemente atualizando o que é mostrado ao usuário, isso sobrecarrega o sistema de forma desnecessária, pois poderia ter sido utilizado o *broker* do serviço M2M para atualizar os dados na medida que são recebidos pelo sistema, assim evitando processos nos demais serviços e tornando a visualização em tempo real.



## 7 CONSIDERAÇÕES FINAIS

O Agro Sensor, embora tenha tido sucesso como protótipo demonstrando possibilidade de se tornar um produto, mostra que há melhorias que podem ser feitas, tais como troca da fonte, a fim de melhorar a eficiência de consumo e usabilidade. Outra melhoria seria criar uma placa especializada apenas em tomar medidas, sendo consideravelmente mais barato que a versão completa e poderia trabalhar em conjunto ou de forma completamente independente com o Agro Sensor. Isto tornaria o projeto mais versátil.

Já, no sistema *web* as principais melhorias seriam na visualização dos dados, utilizando gráficos com escalas customizáveis, bem como na forma que os dados são atualizados na página, onde o *broker* MQTT do servidor M2M traria melhor uso do sistema.

Outra proposta seria a implementação de controle, fazendo uso outras periférias da placa, como o relê. Isso permitiria engatilhar eventos, que poderiam trabalhar junto de uma inteligência artificial tornando o processo mais integrado e eficiente. A inteligência artificial também poderia emitir alertas de diagnósticos, por exemplo, utilizando dados de telemetria interna como consumo e alimentação. Isto permitiria que o sistema indicasse quando a bateria deve ser trocada.

Desenvolver esse projeto mostrou de forma mais clara os desafios tecnológicos envolvidos para implantar esse tipo de sistema. Por exemplo, o simples fato do dispositivo permitir um controle pode tornar a infraestrutura mais cara, uma vez que é preciso manter um link constante com a nuvem. Outro obstáculo é o gerenciamento dos sensores, em razão da quantidade utilizada, mas isso pôde ser contornado com o próprio protocolo de comunicação.

Com os resultados obtidos foi possível perceber diversas melhorias e propostas que futuramente podem vir a ser implementadas no trabalho, tanto no hardware como no software. Além disso, as dificuldades enfrentadas agregaram mais experiências na execução de projeto e no uso de tecnologias.

## REFERÊNCIAS

- AMAZON WEB SERVICES. **O que é FreeRTOS?** 2021. Disponível em: [https://docs.aws.amazon.com/pt\\_br/freertos/latest/userguide/what-is-freertos.html](https://docs.aws.amazon.com/pt_br/freertos/latest/userguide/what-is-freertos.html). Acesso em: 4 nov. 2021.
- ANDRADE, Ana Paula de. **Conhecendo o codeigniter, framework PHP**. 2021. Disponível em: <https://www.treinaweb.com.br/blog/conhecendo-o-codeigniter-framework-php>. Acesso em: 2 nov. 2021.
- ASHTON, Kevin. **That 'Internet of Things' Thing**: in the real world, things matter more than ideas. 2009. Disponível em: <https://www.rfidjournal.com/that-internet-of-things-thing>. Acesso em: 20 out. 2021.
- BALLERINI, Rafaella. **HTML, CSS e Javascript, quais as diferenças?** 2021. Disponível em: <https://www.alura.com.br/artigos/html-css-e-js-definicoes>. Acesso em: 2 nov. 2021.
- BARRY, Richard. **Mastering the FreeRTOS real time kernel**: a hands-on tutorial guide. 2016. Disponível em: [https://freertos.org/Documentation/RTOS\\_book.html](https://freertos.org/Documentation/RTOS_book.html). Acesso em: 2 nov. 2021.
- BRENES, José A. *et al.* **Sistemas de apoyo a la toma de decisiones que usan inteligencia artificial en la agricultura de precisión**: un mapeo sistemático de literatura. 2020. Disponível em: <https://core.ac.uk/reader/344709670>. Acesso em: 18 out. 2021.
- COSTA, Joanne Régis... [et al.] **Cidades e comunidades sustentáveis: contribuições da EMBRAPA**. Editoras técnicas. Brasília, DF: Embrapa, 2018a. Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/184240/1/ODS-11-Cidades-e-comunidades-sustentaveis.pdf>. Acesso em 08 de out. 2021.
- COSTA, Cainã L. *et al.* **Internet das coisas (IOT): um estudo exploratório em agronegócios**: internet of things (iot): na exploratory study in agribusiness. Internet of Things (IOT): na exploratory study in agribusiness. 2018b. Disponível em: <https://www.ufrgs.br/cienagro/wp-content/uploads/2018/10/Internet-das-coisas-IOT-um-estudo-explorat%C3%B3rio-em-agroneg%C3%B3cios.docx-Cain%C3%A3-Lima-Costa.pdf>. Acesso em: 18 out. 2021.
- DOCKER, **Docker overview**, 2020. Disponível em: <https://docs.docker.com/get-started/overview/>. Acesso em: 5 nov. 2021.
- ENVISIA. **Você sabe o que é a tecnologia Node-Red?** about. About. 2019. Disponível em: <https://www.envisia.com.br/2019/01/08/voce-sabe-o-que-e-tecnologia-node-red/>. Acesso em: 2 nov. 2021.



ESPRESSIF. **ESP32**: esp-idf programming guide. ESP-IDF Programming Guide. 2021. Disponível em: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>. Acesso em: 2 nov. 2021.

FRANÇA, Renata de Souza. **Transformação agrícola digital**: o entrelaçamento da agricultura e transformação digital para o futuro inovador do setor agrícola. 2021. Disponível em: <https://periodicos.uninove.br/exacta/article/download/18745/9089>. Acesso em: 15 out. 2021.

GUEDES, Marylene. **O que é MVC?** 2020. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-mvc>. Acesso em: 3 nov. 2021.

HIVEMQ, **MQTT & MQTT 5 Essentials**: A comprehensive overview of MQTT facts and features for beginners and experts alike, 2020. Disponível em: <https://www.hivemq.com/download-mqtt-ebook/>. Acesso em: 04 nov. 2021.

IBGE. **Em 2017, o Brasil consumia 6,3 litros d'água para cada R\$ 1 gerado pela economia**. 2020. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/27607-em-2017-o-brasil-consumia-6-3-litros-d-agua-para-cada-r-1-gerado-pela-economia>. Acesso em: 20 out. 2021.

KAMIENSKI, Carlos; SOININEN, Juha-Pekka; TAUMBERGER, Markus; FERNANDES, Stenio; TOSCANO, Attilio; CINOTTI, Tullio Salmon; MAIA, Rodrigo Filev; TORRE NETO, Andre. **SWAMP**: an IoT-based smart wWater management platform for precision irrigation in agriculture. 2018. Disponível em: <https://ieeexplore.ieee.org/document/8534541/>. Acesso em: 13 dez. 2020.

LIMA, Charles Borges de; VILLAÇA, Marco V. M. **AVR e Arduino técnicas de projeto**. 2. ed. Florianópolis: Edição dos Autores, 2012.

MAGRANI, Eduardo. **A Internet das coisas**. Rio de Janeiro: FGV, 2018. 192 p. Disponível em: <http://eduardomagrani.com/livro-internet-da-coisas-2018/>. Acesso em: 20 out. 2021.

MASSRUHÁ, Silvia Maria Fonseca Silveira; LEITE, Maria Angelica de Andrade. **AGRO 4.0**: rumo à agricultura digital. 2017. Disponível em: <https://www.embrapa.br/busca-de-publicacoes/-/publicacao/1073150/agro-40---rumo-a-agricultura-digital>. Acesso em: 23 out. 2021.

MASSRUHÁ, Silvia Maria Fonseca Silveira; LEITE, Maria Angelica de Andrade; OLIVEIRA, Stanley Robson de Medeiros; MEIRA, Carlos Alberto Alves; LUCHIARI JUNIOR, Arioaldo; BOLFE, Édson Luis. **Agricultura digital**: pesquisa, desenvolvimento e inovação nas cadeias produtivas. Pesquisa, desenvolvimento e inovação nas cadeias produtivas. 2020. Disponível em: <http://www.alice.cnptia.embrapa.br/handle/doc/1126213>. Acesso em: 26 out. 2021.

MENDONÇA, Hédio Sousa. **Protocolo I2C**: comunicação entre Arduinos. 2018. Disponível em: <https://portal.vidadesilicio.com.br/i2c-comunicacao-entre-arduinov/>. Acesso em: 17 nov. 2021.

MPS (emp.). **MP1584 – 3A, 1.5MHz, 28V step-down converter**. 2011. Disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/551592/MPS/MP1584.html>. Acesso em: 17 nov. 2021.

MYSQL. **About MySQL**. Disponível em: <https://www.mysql.com/about/>. Acesso em: 4 nov. 2021.

OPENJS FOUNDATION. **Node-RED**: about. 2021. Disponível em: <https://nodered.org/about/>. Acesso em: 3 nov. 2021.

PEREIRA, Ana Paula. **O que é CSS?** 2009. Disponível em: <https://www.tecmundo.com.br/programacao/2705-o-que-e-css-.htm>. Acesso em: 2 nov. 2021.

PUHAN, Janez. **Operating systems, embedded systems, and real-time systems**. Ljubljana: Fe Publishing, 2015.

PHP. **Manual do PHP**. 2021. Disponível em: [https://www.php.net/manual/pt\\_BR/index.php](https://www.php.net/manual/pt_BR/index.php). Acesso em: 2 nov. 2021.

PHP. **História do PHP**: php tools, fi, kit de construção, e php/fi. PHP Tools, FI, Kit de Construção, e PHP/FI. 2021. Disponível em: [https://www.php.net/manual/pt\\_BR/history.php.php](https://www.php.net/manual/pt_BR/history.php.php). Acesso em: 3 nov. 2021.

SENSIRION. **Datasheet SHT3x-DIS**: humidity and temperature sensor. 2016. Disponível em: [https://www.mouser.com/datasheet/2/682/Sensirion\\_Humidity\\_Sensors\\_SHT3x\\_Data\\_sheet\\_digital-971521.pdf](https://www.mouser.com/datasheet/2/682/Sensirion_Humidity_Sensors_SHT3x_Data_sheet_digital-971521.pdf). Acesso em: 17 nov. 2021.

SILVA, Allan Leon Casemiro da; GÓES, Bruno César; PUTTI, Fernando Ferrari. **A modernização da agricultura e o caso brasileiro**. 2021. P. 53. Disponível em: [https://www.researchgate.net/profile/Ariane-Nascimento/publication/349111038\\_CAPITULO\\_5\\_INSETOS\\_ALIMENTO\\_SUSTENTAVEL\\_PARA\\_NUTRICA\\_O\\_ANIMAL/links/6021257945851589398c9798/CAPITULO-5-INSETOS-ALIMENTO-SUSTENTAVEL-PARA-NUTRICA\\_O-ANIMAL.pdf](https://www.researchgate.net/profile/Ariane-Nascimento/publication/349111038_CAPITULO_5_INSETOS_ALIMENTO_SUSTENTAVEL_PARA_NUTRICA_O_ANIMAL/links/6021257945851589398c9798/CAPITULO-5-INSETOS-ALIMENTO-SUSTENTAVEL-PARA-NUTRICA_O-ANIMAL.pdf). Acesso em: 26 out. 2021.

SILVA, Giancarlo. **O que é e como funciona a linguagem JavaScript?** 2015. Disponível em: <https://canaltech.com.br/internet/O-que-e-e-como-funciona-a-linguagem-JavaScript/>. Acesso em: 2 nov. 2021.

SILVEIRA, Paulo. **O que é SQL?** 2019. Disponível em: <https://www.alura.com.br/artigos/o-que-e-sql>. Acesso em: 2 nov. 2021.

SCHILDT, Herbert. **C completo e total**. 3. ed. São Paulo: Makron Books, 1995. Disponível em: <https://www.inf.ufpr.br/lesoliveira/download/c-completo-total.pdf>. Acesso em: 2 nov. 2021.

SOARES, Josiel Ferreira *et al.* **Os impactos da utilização da internet das coisas em fazendas e hortas urbanas**. 2020. Disponível em: <https://repositorio.animaeducacao.com.br/handle/ANIMA/3932>. Acesso em: 18 out. 2021.

RED HAT. **O que é Docker?** 2020. Disponível em: <https://www.redhat.com/pt-br/topics/containers/what-is-docker>. Acesso em: 5 nov. 2021.

RIBEIRO, Josiana Gonçalves; MARINHO, Douglas Yusuf; ESPINOSA, Jose Waldo Martínez. **Agricultura 4.0: desafios à produção de alimentos e inovações tecnológicas**. 2018. Disponível em: <https://doutoragro.com/agtechhub/wp-content/uploads/2020/07/UFG-Agricultura-4.0.pdf>. Acesso em: 10 out. 2021.

ROSSONI, *et al.* MAIA, João Carlos de Souza. **A internet das coisas e o agronegócio no Brasil: um olhar sobre a produção científica brasileira na base Scopus: the internet of things and agribusiness in brazil: a look at brazilian scientiic production. The Internet of Things and Agribusiness in Brazil: a look at brazilian scientiic production**. 2019. Disponível em: <https://periodicos.ufba.br/index.php/nit/article/view/32374>. Acesso em: 18 out. 2021.

TOTVS. **O que é HTML?** saiba como esse recurso funciona. 2020. Disponível em: <https://www.totvs.com/blog/developers/o-que-e-html/>. Acesso em: 2 nov. 2021.

WANG, K. C. **Embedded and real-time operating Systems**. Washington State: Springer, 2017.

## APÊNDICE A - Tabelas e Figuras

Tabela 2 - *Sprints* para o desenvolvimento do projeto

Sprint	Partes	Descrição
1	Hardware	Desenhar esquema elétrico
2		Projetar layout do Circuito (PCB)
3		Fabricar Circuito (PCB)
4		Montar placa
6	Firmware	Desenvolver abstração dos periféricos do circuito
7		Desenvolver módulo de configuração
5		Desenvolver módulo de telemetria interna
6		Desenvolver módulo Sensor SHT31
7		Desenvolver módulo de conexão Wi-Fi
8		Desenvolver módulo de comunicação MQTT
9		Criar lógica da aplicação
10	Banco de dados	Criar script SQL
11		Configurar ambiente
12	Servidor M2M	Criar broker
13		Desenvolver lógica de Assinatura de tópicos
14		Desenvolver lógica de validação de medidas
15		Desenvolver lógica de Sinalização do status de alimentação
16		Desenvolver lógica de Sinalização do status de desconexão
18	Servidor Web	Desenvolver cadastro de dispositivo
19		Desenvolver painel de gerenciamento de dispositivos
20		Desenvolver pagina/cadastro para atualizar dispositivo

21	Desenvolver cadastro de usuário
22	Desenvolver painel de gerenciamento de usuários
23	Desenvolver pagina/cadastro para atualizar o usuário logado
24	Desenvolver painel de visão geral dos dispositivos ( <i>Overview</i> )
25	Desenvolver página de detalhes do dispositivo
26	Desenvolver função para baixar dados coletados
27	Desenvolver função para mostrar dispositivo por status

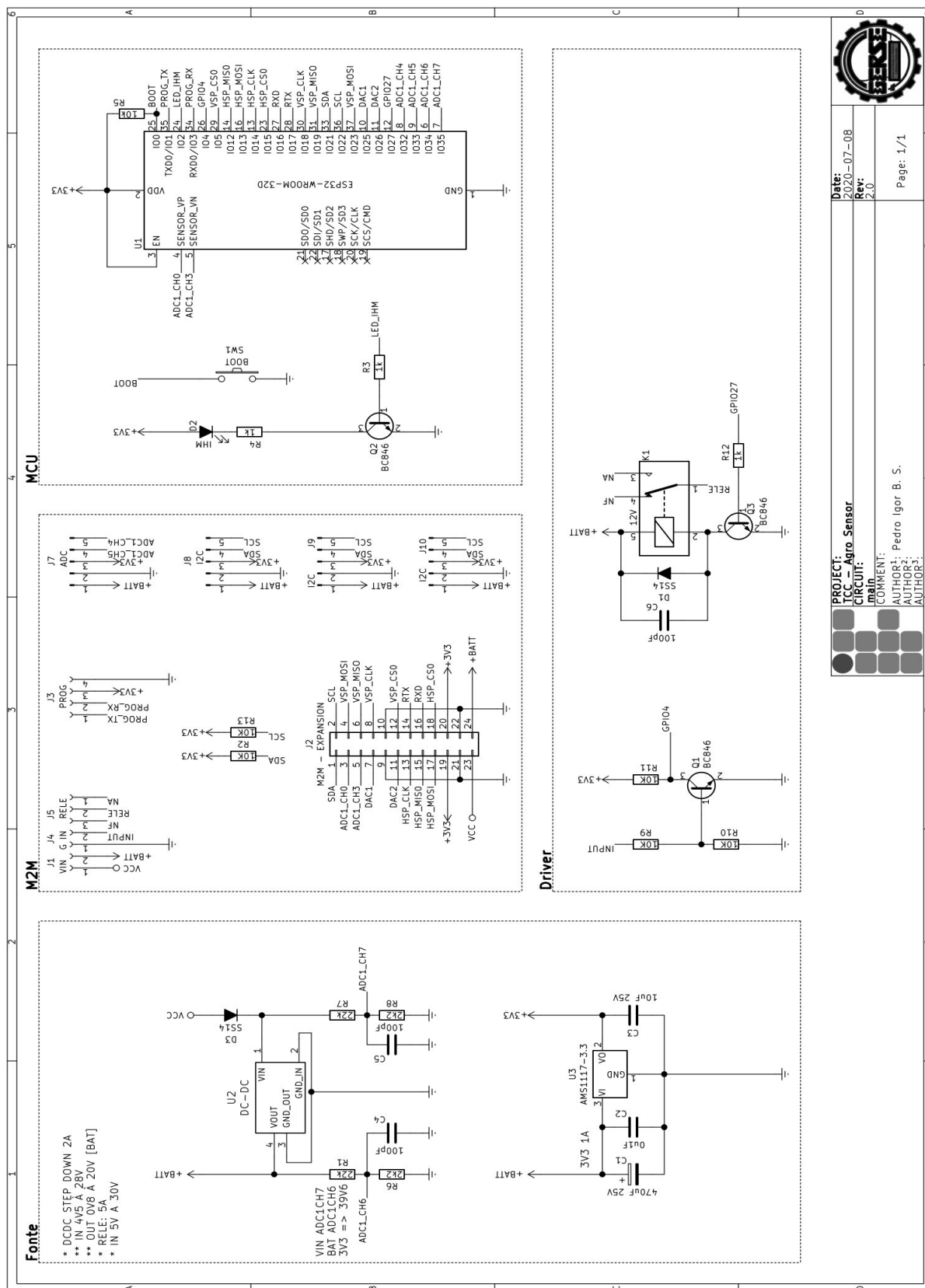
Fonte: Elaborado pelo autor.

Tabela 3 - Especificações técnicas do painel solar, bateria e modulo

<b>Painel Solar</b>			
Potência:	5W	Condição máxima de potência	
Tensão em circuito aberto:	21.5V	Tensão máxima em consumo:	17.5V
corrente de curto circuito:	0.34A	Corrente máxima em consumo:	0.29V
<b>Bateria</b>			
Tenção:	6V	Potencia:	27 W/h
Carga:	4500 mAh	Regime de carga :	7.2V
<b>Módulo ESP-32</b>			
Consumo	20mA a 240mA		

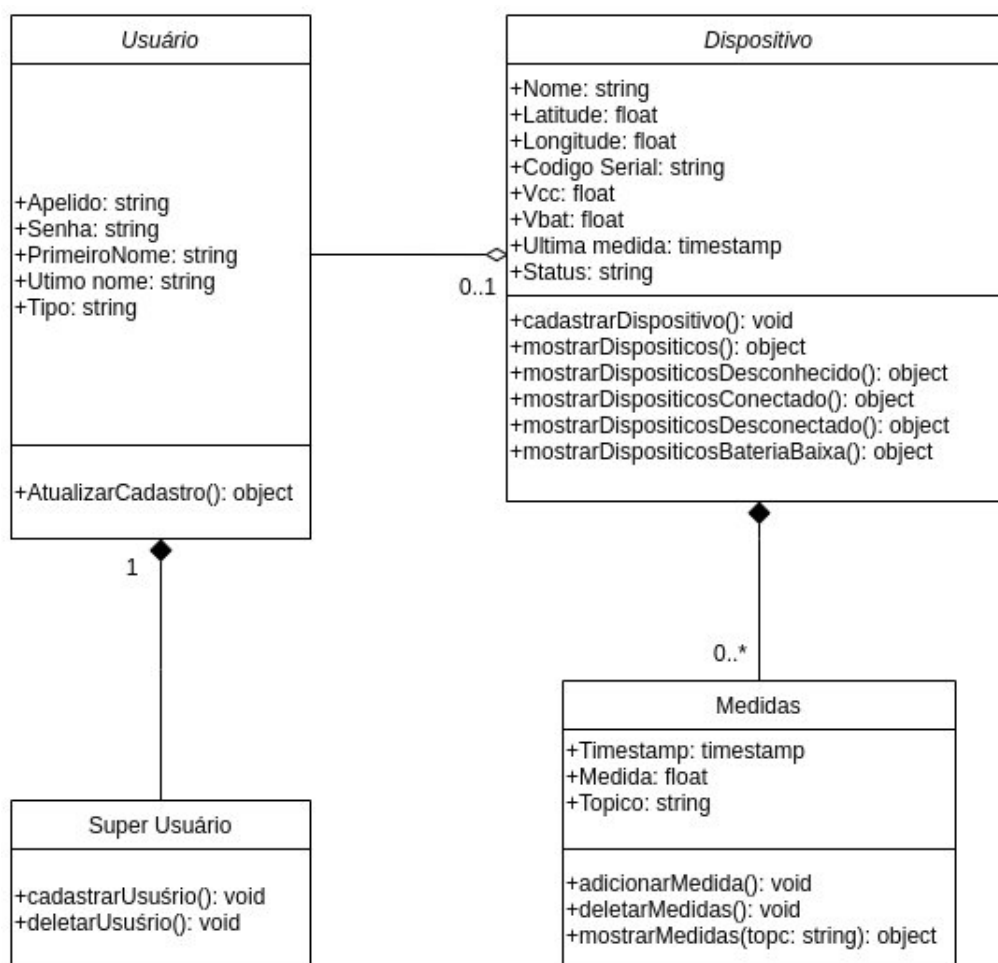
Fonte: Elaborado pelo autor.

Figura 15 - Esquema elétrico do Agro Sensor



Fonte: Elaborado pelo autor.

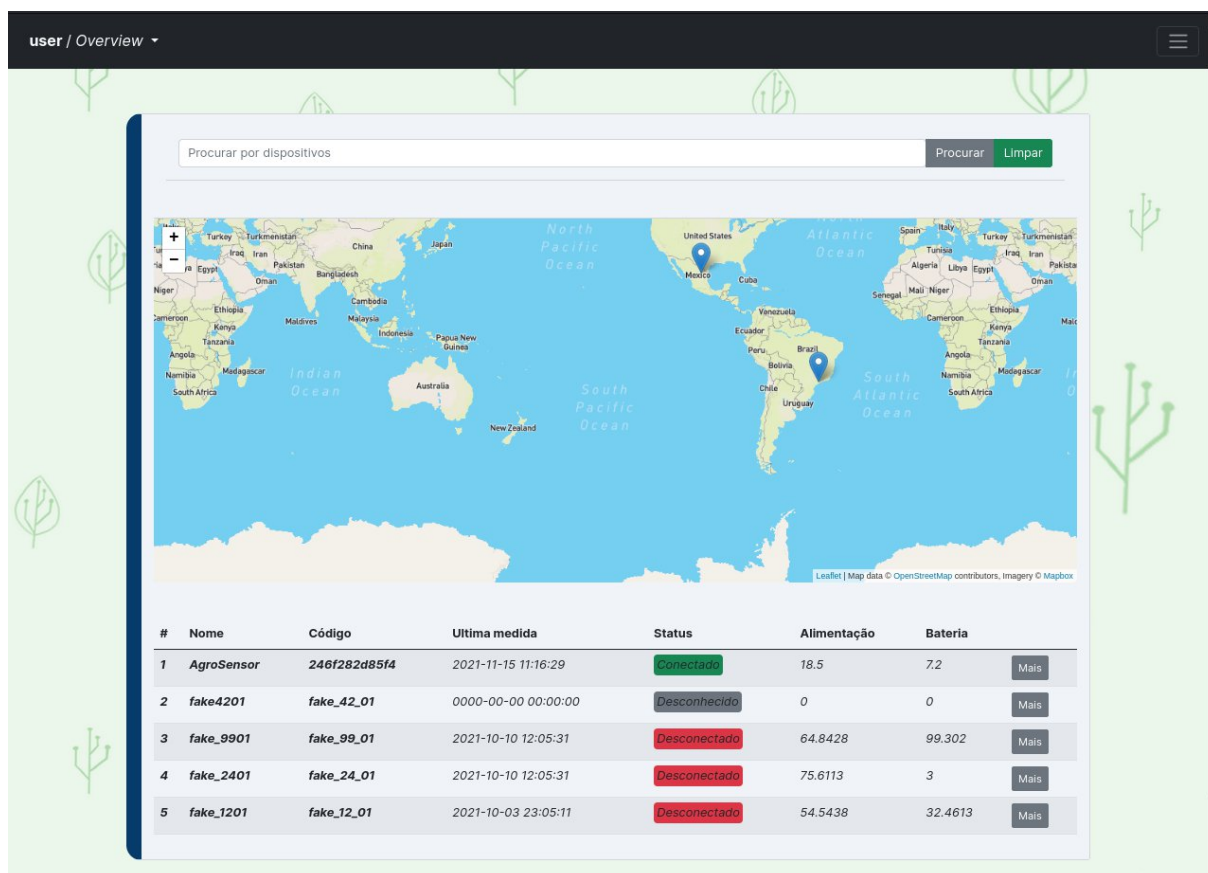
Figura 16 - Diagrama de classe



Fonte: Elaborado pelo autor.



Figura 17 - Tela de visão geral (overview)



Fonte: Elaborado pelo autor.

Figura 18 - Tela de gerenciamento de dispositivos

Pedro igoR / Gerenciar Dispositivos ▾

Dispositivos

Procurar dispositivos

Procurar

Limpar

Novo

1 AgroSensor

Esse é o protótipo "AgroSensor" desenvolvido para aquisição de temperatura e umidade do solo, ele contém uma sonda para realizar essas medidas. ESTÁ EM UMA JABUTICABEIRA COM PAINEL SOLAR E BATERIA.

Código	Status	Alimentação de entrada	Alimentação da bateria	Última medida	Latitude	Longitude
246f282d85f4	Conectado	18.5	7.2	2021-11-15 11:16:29	-23.119364248748248	-46.55508041381837

Ver

Editar

2 Fake4201

Esse é um dispositivo falso com o objetivo de mostrar como é um sensor desconhecido na plataforma, isso significa que o mesmo foi cadastrado no sistema porém nunca conectou-se ao Dashboard.

Código	Status	Alimentação de entrada	Alimentação da bateria	Última medida	Latitude	Longitude
fake_42_01	Desconhecido	0	0	0000-00-00 00:00:00	-23.644524198573677	133.85742187500003

Ver

Editar

3 Fake\_9901

Esse é um dispositivo falso com o objetivo de mostrar como é um sensor conectado na plataforma. São gerados valores aleatórios.

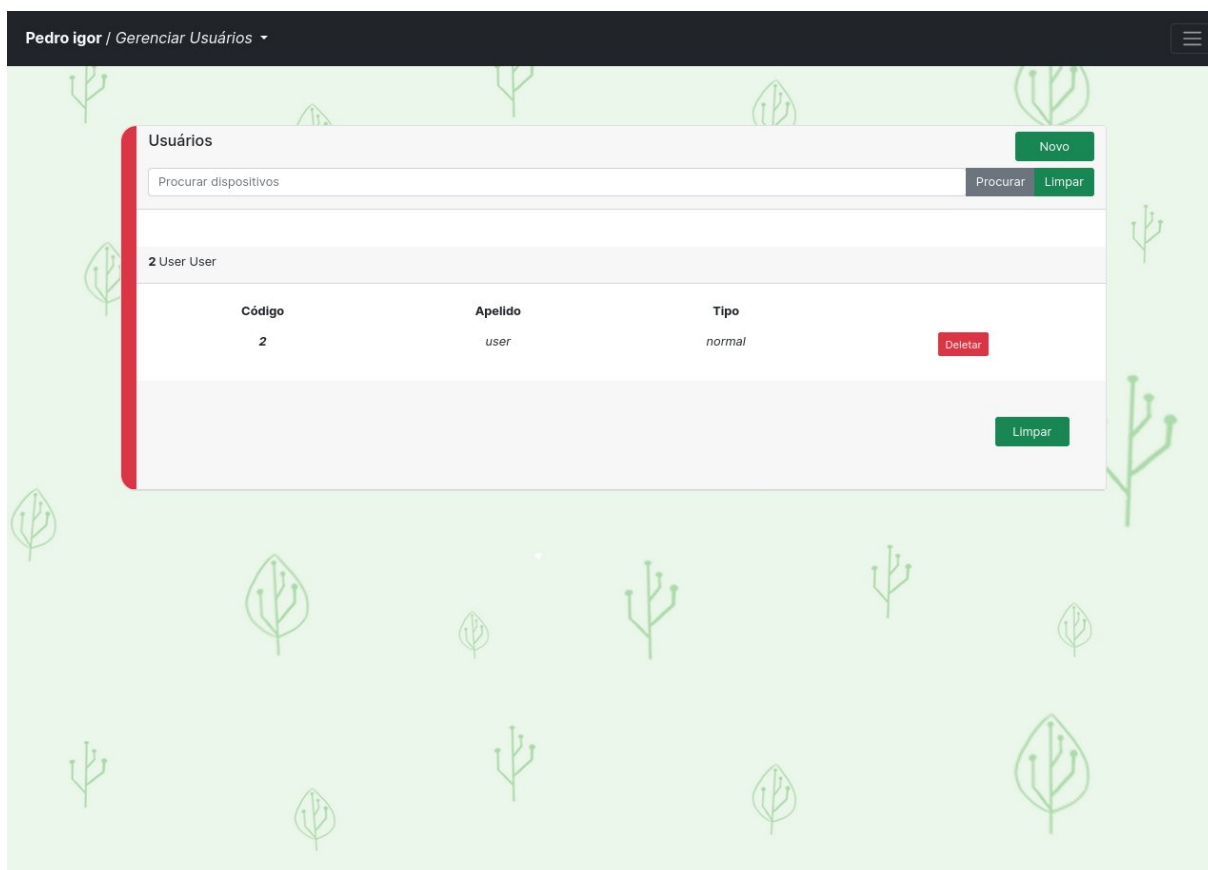
Código	Status	Alimentação de entrada	Alimentação da bateria	Última medida	Latitude	Longitude
fake_99_01	Desconectado	64.8428	99.302	2021-10-10 12:05:31	25.46311452925943	-100.986328125

Ver

Editar

Fonte: Elaborado pelo autor.

Figura 19 - Tela de gerenciamento de usuários



Fonte: Elaborado pelo autor.

Figura 20 - Formulário de cadastro ou atualização de dispositivos

Pedro Igor / Criar Dispositivo ▾

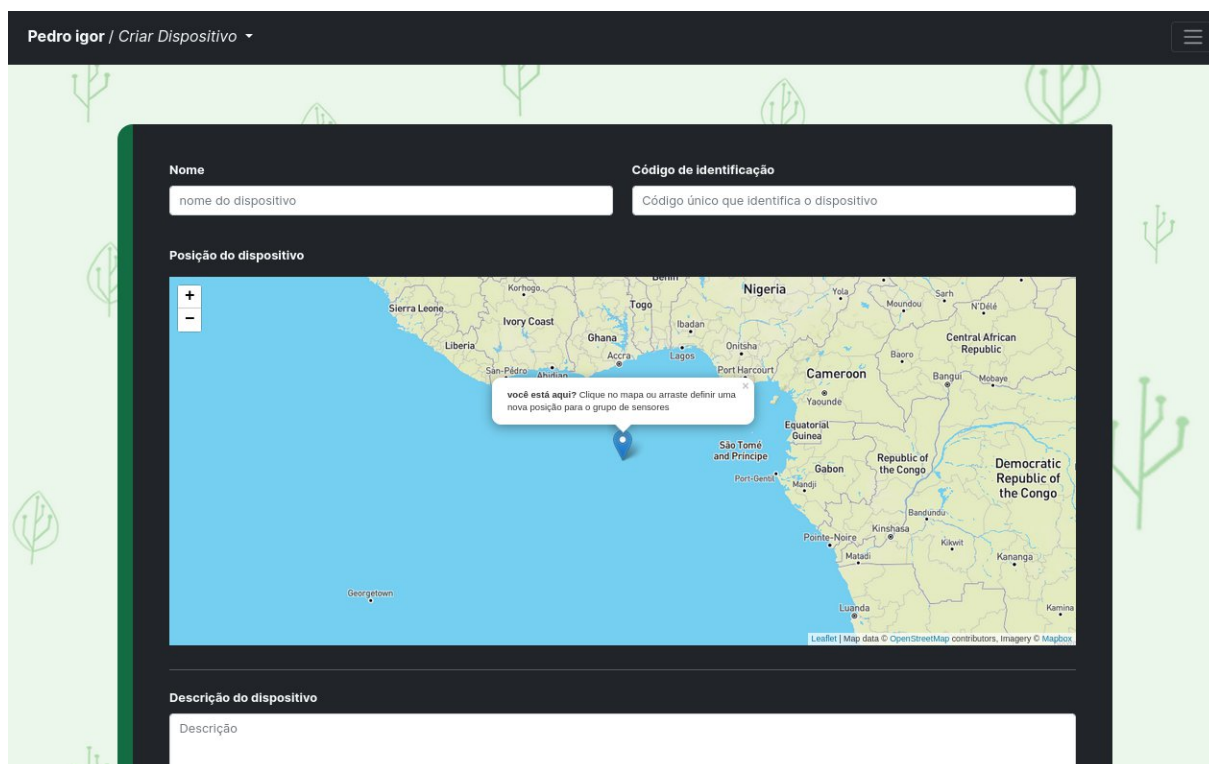
**Nome**  
nome do dispositivo

**Código de identificação**  
Código único que identifica o dispositivo

**Posição do dispositivo**

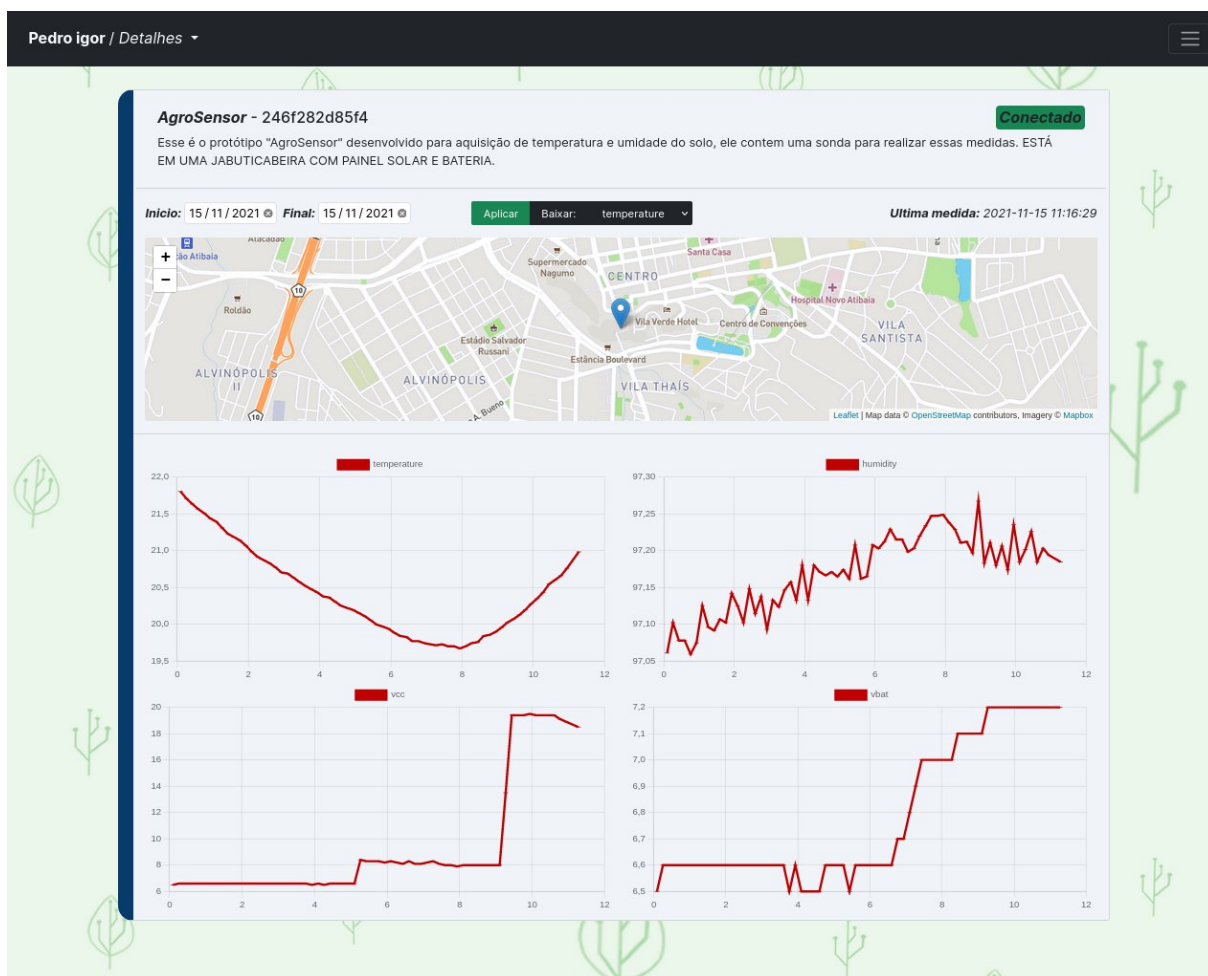
+  
-  
você está aqui? Clique no mapa ou arraste definir uma nova posição para o grupo de sensores

**Descrição do dispositivo**  
Descrição



Fonte: Elaborado pelo autor.

Figura 21 - Tela de visão detalhada dos dispositivos



Fonte: Elaborado pelo autor.