

MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
FARROUPILHA - CAMPUS SÃO VICENTE DO SUL

**SISTEMA DE IRRIGAÇÃO INTELIGENTE  
UTILIZANDO INTERNET DAS COISAS (IOT)**

**TRABALHO DE CONCLUSÃO DO CURSO SUPERIOR DE  
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS**

**THIAGO RYAN DA SILVA AIRES**

São Vicente do Sul, RS  
2024

**Thiago Ryan da Silva Aires**

**SISTEMA DE IRRIGAÇÃO INTELIGENTE UTILIZANDO INTERNET DAS COISAS  
(IOT)**

Trabalho de Conclusão de Curso apresentado  
ao Curso Superior de Tecnologia em Análise e  
Desenvolvimento de Sistemas do Instituto Fe-  
deral Farroupilha - Campus São Vicente do Sul  
(IFFar - SVS, RS), como requisito parcial para  
a obtenção do grau de **Tecnólogo em Análise e  
Desenvolvimento de Sistemas**

Orientador: Prof. D.r Gleizer Bierhalz Voss

Co-orientador: Prof. Me. Rogerio Cassanta Rosado

São Vicente do Sul, RS

2024

da Silva Aires, Thiago Ryan

Sistema de Irrigação Inteligente Utilizando Internet das Coisas  
(IoT) / por Thiago Ryan da Silva Aires. – 2024.

73 f.: il.; 30 cm.

Orientador: Gleizer Bierhalz Voss

Co-orientador: Rogerio Cassanta Rosado

Trabalho de Conclusão de Curso - Instituto Federal Farroupilha -  
Campus São Vicente do Sul, Instituto Federal Farroupilha - Campus  
São Vicente do Sul, Curso Superior de Tecnologia em Análise e Desen-  
volvimento de Sistemas, RS, 2024.

1. Modelo. 2. Latex. 3. Tcc. 4. Graduação. I. Bierhalz Voss,  
Gleizer. II. Cassanta Rosado, Rogerio. III. Título.

**Thiago Ryan da Silva Aires**

**SISTEMA DE IRRIGAÇÃO INTELIGENTE UTILIZANDO INTERNET DAS COISAS  
(IOT)**

Trabalho de Conclusão de Curso apresentado  
ao Curso Superior de Tecnologia em Análise e  
Desenvolvimento de Sistemas do Instituto Fe-  
deral Farroupilha - Campus São Vicente do Sul  
(IFFar - SVS, RS), como requisito parcial para  
a obtenção do grau de **Tecnólogo em Análise e  
Desenvolvimento de Sistemas**

**Aprovado em 07 de 03 de 2024:**

---

**Gleizer Bierhalz Voss, D.r (IFFar - SVS)**  
(Presidente/Orientador)

---

**Rogerio Cassanta Rosado, Me. (IFFar - SVS)**  
(Coorientador)

---

**Nome membro banca Sobre nome, Me. (UFSM)**

---

**Nome membro banca Sobre nome, Tecg. (UFSM)**

São Vicente do Sul, RS

2024

## **DEDICATÓRIA**

*Dedico este trabalho a .....*

## **AGRADECIMENTOS**

*Agradeço .....*

*“O valor de uma coisa depende da maneira como a abordamos mentalmente e não da coisa em si”*

(JIGORO KANO)

## **RESUMO**

### **SISTEMA DE IRRIGAÇÃO INTELIGENTE UTILIZANDO INTERNET DAS COISAS (IOT)**

AUTOR: THIAGO RYAN DA SILVA AIRES

ORIENTADOR: GLEIZER BIERHALZ VOSS

CO-ORIENTADOR: ROGERIO CASSANTA ROSADO

Aqui você escreve o resumo. Lembrando no máximo 250 palavras para tcc e 500 palavras para tese ou dissertação.

**Palavras-chave:** Modelo. latex. tcc. graduação.

## **ABSTRACT**

### **ABSTRACT TITLE**

AUTHOR: THIAGO RYAN DA SILVA AIRES

ADVISOR: GLEIZER BIERHALZ VOSS

COADVISOR: ROGERIO CASSANTA ROSADO

Here you write the summary. Remembering a maximum of 250 words for tcc and 500 words for thesis or dissertation.

**Keywords:** Model. latex. tcc. graduation.

## LISTA DE FIGURAS

1	Irrigação superficial .....	16
2	Irrigação Subsuperficial .....	17
3	Irrigação por aspersão – sistema convencional .....	18
4	Microirrigação - Irrigação Por Gotejamento .....	18
5	Embrapa - Encanteiramento para o plantio de pimenta .....	19
6	Internet das coisas no setor de serviços .....	20
7	Componentes de um microcontrolador .....	21
8	Localização do IFFAR - SVS .....	23
9	Diagrama da Placa .....	26
10	DHT11 .....	27
11	Sensor de umidade do solo .....	28
12	Interface do Arduino IDE.....	30
13	Interface do ThingsBoard.....	32
14	Diagrama de Implatação .....	37
15	Diagrama do ESP32 Transmissor .....	38
16	Diagrama do ESP32 Receptor .....	39
17	Diagrama do Usuário.....	39
18	Diagrama do Administrador .....	40
19	Diagrama Entidade-Relacionamento .....	41
20	Diagrama de Classe .....	42
21	Diagrama View .....	42
22	Diagrama Controller .....	43
23	Hardware .....	45
24	Código da função para transmitir os dados pelo LoRa. ....	46
25	Código da função para receber os dados pelo LoRa.....	46
26	Código da comunicação com o ThingsBoard.....	47
27	Código da comunicação com o ThingsBoard. ....	48
28	Criando um dispositivo. ....	49
29	Ultimos dados recebidos. ....	49

## **LISTA DE TABELAS**

## **LISTA DE ABREVIATURAS E SIGLAS**

IoT	Internet das Coisas
IFFAR	Instituto Federal Farroupilha - Campus São Vicente do Sul
LoRa	Long Range
MQTT	MQ Telemetry Transport
HTTP	Hypertext Transfer Protocol
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
MVC	Model-View-Controller
UML	Unified Modeling Language
DER	Diagrama Entidade-Relacionamento

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	14
1.1	OBJETIVOS .....	14
1.2	OBJETIVO GERAL .....	14
1.3	OBJETIVO ESPECÍFICOS .....	15
1.4	ORGANIZAÇÃO DO TRABALHO .....	15
<b>2</b>	<b>REFERENCIAL TEÓRICO.....</b>	16
2.1	SISTEMA DE IRRIGAÇÃO .....	16
2.2	CULTIVO EM SULCO E SISTEMA CONVENCIONAL.....	19
2.3	INTERNET DAS COISAS (IOT).....	20
2.4	MICROCONTROLADORES .....	21
2.5	INSTITUTO FEDERAL FARROUPILHA - CAMPUS S.V.S (IFFAR - SVS) ...	22
<b>2.5.1</b>	<b>Setor de Horticultura .....</b>	23
2.6	TRABALHOS RELACIONADOS .....	23
<b>3</b>	<b>MATERIAIS E MÉTODOS .....</b>	26
3.1	MATERIAIS .....	26
3.1.1	<b>Heltec ESP 32 LoRa WiFi V2 .....</b>	26
3.1.2	<b>DHT-11 .....</b>	27
3.1.3	<b>Módulo Sensor de Umidade do Solo .....</b>	28
3.1.4	<b>Long Range (LoRa) .....</b>	29
3.1.5	<b>Arduino IDE .....</b>	29
3.1.6	<b>C++ .....</b>	30
3.1.7	<b>MQ Telemetry Transport (MQTT) .....</b>	31
3.1.8	<b>ThingsBoard .....</b>	31
3.1.9	<b>Hypertext Transfer Protocol (HTTP) .....</b>	32
3.1.10	<b>XAMPP .....</b>	33
3.1.11	<b>PHP .....</b>	33
3.1.12	<b>MySQL .....</b>	33
3.1.13	<b>Hyper Text Markup Language (HTML) .....</b>	34
3.1.14	<b>Cascading Style Sheets (CSS) .....</b>	34
3.1.15	<b>JavaScript (JS) .....</b>	35
3.2	MÉTODOS .....	35
<b>3.2.1</b>	<b>Arquitetura Model-View-Controller (MVC) .....</b>	35
<b>3.2.2</b>	<b>Unified Modeling Language (UML).....</b>	36
<b>3.2.3</b>	<b>Diagrama de Implantação .....</b>	36
<b>3.2.4</b>	<b>Diagrama de Caso de Uso.....</b>	37
<b>3.2.5</b>	<b>Diagrama Entidade-Relacionamento (DER) .....</b>	40
<b>3.2.6</b>	<b>Diagrama de classe .....</b>	41
<b>3.2.7</b>	<b>Requisitos funcionais e não funcionais .....</b>	43
<b>3.2.8</b>	<b>Protótipo .....</b>	43
<b>4</b>	<b>RESULTADOS .....</b>	45
4.1	CONECTANDO OS MICROCONTROLADORES.....	45
4.2	UTILIZANDO O THINGSBOARD .....	48
4.3	DESENVOLVIMENTO DO SISTEMA WEB .....	49
<b>5</b>	<b>CONCLUSÃO .....</b>	50
	<b>REFERÊNCIAS .....</b>	52

<b>APÊNDICES.....</b>	<b>56</b>
<b>ANEXOS .....</b>	<b>58</b>

# **1 INTRODUÇÃO**

A importância da irrigação na agricultura é destacada pela essencialidade da água para a produtividade do solo e o desenvolvimento das plantas. Sem água, o solo se torna improdutivo, impossibilitando a absorção de nutrientes pelas plantas durante períodos de seca. Desde tempos antigos, os agricultores reconhecem a água como o principal elemento para a fertilidade do solo e o crescimento das culturas, sendo responsável por dissolver nutrientes essenciais e transportá-los para as plantas (CALZAVARA, 1953).

O desperdício de água na irrigação, além de aumentar os custos de produção, acarreta custos ambientais pelo comprometimento da disponibilidade e da qualidade da água. Essa situação tem levado muitos projetos de irrigação, em todo o mundo, a uma condição de baixa sustentabilidade econômica e socioambiental (CULLEN, 2004). Segundo OSTER; WICHELNS (2003), a sustentabilidade de projetos de irrigação somente pode ser atingida por meio de ações efetivas tanto pelo agricultor quanto por órgãos públicos, visando a minimizar o desperdício de água durante sua captação, condução e aplicação à cultura, a degradação das águas superficiais e subterrâneas e, no caso das regiões áridas e semiáridas, os problemas de salinização.

Desenvolver um sistema de irrigação inteligente utilizando Internet das Coisas (IoT) emerge como uma solução promissora para enfrentar os desafios relacionados ao manejo inadequado da água na agricultura. Este sistema permitira aos agricultores monitorar e controlar a umidade do solo, temperatura, umidade do ar e luminosidade do ambiente em tempo real. Com base nessas informações, os agricultores poderiam tomar decisões mais assertivas sobre o momento ideal para irrigar suas plantações, otimizando o uso da água e melhorando o desempenho da produção.

## **1.1 OBJETIVOS**

Nessa seção são mostrados o objetivo geral, específicos e a organização do trabalho.

## **1.2 OBJETIVO GERAL**

O principal objetivo deste trabalho é desenvolver um sistema inteligente utilizando IoT para monitoramento e controle de irrigação em estufas, hortas, campos, entre outros.

### 1.3 OBJETIVO ESPECÍFICOS

Para atingir o objetivo geral, foram estabelecidos os seguintes objetivos específicos:

- Implementar sensores de umidade do solo para medir com precisão os níveis de umidade no solo em áreas por exemplo da horta ou campo.
- Criar algoritmos de controle que utilizem os dados dos sensores de umidade do solo, para determinar quando e quanto irrigar.
- Incorporar sensores de temperatura e umidade para levar em consideração as condições ambientais ao determinar os requisitos de irrigação das plantas.
- Integração de módulos de comunicação (WiFi, Bluetooth, LoRa) para permitir a comunicação entre o sistema de irrigação e uma plataforma IoT para monitoramento e controle remoto.
- Testar o sistema em diferentes condições ambientais e ajustar os algoritmos de acordo com o desempenho observado, garantindo uma operação confiável e eficaz.

### 1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado da seguinte maneira:

- No Capítulo 2, encontra-se o referencial teórico que são modelos e estudos anteriores relevantes que ajudam a contextualizar, explicar e embasar o tópico que está sendo pesquisado.
- No Capítulo 3, são apresentados os Materiais e Métodos utilizados que descreve detalhadamente como o estudo foi conduzido, os métodos e técnicas que foram utilizados, e quais materiais foram empregados.
- Capítulo 4, são feitas as Considerações Finais baseadas nos resultados finais.

## **2 REFERENCIAL TEÓRICO**

Neste Capítulo são discutidos temas relacionados ao desenvolvimento de um sistema de irrigação inteligente utilizando IoT, que será testado e aplicado no Instituto Federal Farroupilha - Campus São Vicente do Sul (IFFAR - SVS), de São Vicente do Sul (RS) na área de Horticultura. Apresenta uma descrição de como é realizada a irrigação, o que é cultivo em sulco e plantio direto, das características do campus e alguns trabalhos semelhantes ao proposto neste trabalho.

### **2.1 SISTEMA DE IRRIGAÇÃO**

Existem diferentes sistemas de irrigação, com características próprias, custos variáveis, vantagens e desvantagens. Dependendo da forma com que a água é aplicada às plantas, os sistemas podem ser agrupados nas seguintes categorias (MAROUELLI; SILVA, 1998):

- Irrigação Superficial. Compreende os sistemas por sulcos, corrugação, faixas e inundação, nos quais a condução e a distribuição da água é feita diretamente sobre a superfície do solo, um exemplo é demonstrado na Figura 1;



Figura 1 – Irrigação superficial

Fonte: PIRES (2021)

Os sistemas superficiais são, geralmente, os que requerem menor investimento inicial e menor uso de energia. Adaptam-se à maioria dos solos, com exceção daqueles com altas taxas de infiltração, como os arenosos, mas requerem terrenos planos ou sistematizados.

- Irrigação Subsuperficial. A água é aplicada sob a superfície do solo através da criação e

controle de um lençol freático. A umidade atinge o sistema radicular da cultura através da ascensão capilar da água, como de mostrado na Figura 2.



Figura 2 – Irrigação Subsuperficial

Fonte: RIVULIS (2019)

A irrigação subsuperficial é caracterizada pelo baixo investimento inicial e baixa utilização de energia e mão-de-obra. Requer, todavia, solos planos ou sistematizados, com camada de solo permeável sobrepondo uma camada impermeável em torno de 1,5 m de profundidade.

- Irrigação por Aspersão. A água é aplicada na forma de chuva, com destaque para os sistemas convencionais portátil, semiportátil e permanente, autopropelido, ramal rolante, deslocamento linear e pivô central, um exemplo é a Figura 3. No entanto, apresenta maior uso de energia, sofre interferência do vento e, sob climas secos e quentes, tem a eficiência reduzida pela alta evaporação.



Figura 3 – Irrigação por aspersão – sistema convencional

Fonte: PIRES (2021)

Em relação aos sistemas superficiais, a aspersão requer menor uso de mão-de-obra e possibilita melhor distribuição de água sobre a superfície do solo.

- Microirrigação. Compreende sistemas como gotejamento, xiquexique, microaspersão e borbulhador ("bubbler"), nos quais a água é, em geral, aplicada ao solo, próximo à planta, em baixo volume e alta frequência, como demonstrado na Figura 4.



Figura 4 – Microirrigação - Irrigação Por Gotejamento

Fonte: MAXMAQ (2019)

Na microirrigação, a água também pode ser aplicada abaixo da superfície do solo, junto às raízes da planta, através de tubos com gotejadores, tubos porosos ou cápsulas porosas enterradas.

## 2.2 CULTIVO EM SULCO E SISTEMA CONVENCIONAL

No sistema convencional de preparo de solo, a aração deve ser feita à profundidade de 25 a 30 cm, seguida da gradagem de nivelamento, conforme Figura 5. Uma segunda gradagem é feita próximo ao plantio, para melhor nivelamento do solo, eliminação de plantas espontâneas e adequação ao sulcamento. Como algumas plantas sofrem muito com o encharcamento, é recomendada periodicamente a subsolagem. O plantio pode ser feito em canteiros, mas o mais comum é o plantio em sulcos. Os sulcos devem ter 30 a 40 cm de largura e 20 a 25 cm de profundidade. A distância entre os sulcos é de 80 cm, devendo-se fazer sua abertura deixando uma declividade de 0,2% a 0,5% para facilitar o escoamento da água sem causar erosão. Após a incorporação de matéria orgânica (1 semana antes do plantio) e dos fertilizantes (um dia antes do plantio), os sulcos devem ficar com forma de U. (MADEIRA; COSTA RIBEIRO; AMARO, 2022).



Figura 5 – Embrapa - Encanteiramento para o plantio de pimenta  
Fonte: MADEIRA; COSTA RIBEIRO; AMARO (2022)

O sistema de sulco-camalhão organiza a lavoura de modo a facilitar a irrigação por sulcos e o cultivo sobre camalhões, os quais promovem uma drenagem eficiente. Este sistema é ideal para solos planos e áreas com declividades uniformes entre 0,05% e 0,10%. Embora também possa ser aplicado em terrenos planos, a eficiência da drenagem é reduzida comparativamente às áreas com alguma inclinação. Os camalhões são formados por equipamentos que, simultaneamente, abrem sulcos e preparam uma zona de cultivo com solo mais profundo e não

compactado. Além de servirem para irrigação e drenagem, os sulcos facilitam o tráfego de máquinas agrícolas (SILVA, 2023).

### 2.3 INTERNET DAS COISAS (IOT)

A Internet das Coisas (do inglês Internet of Things (IoT)) emergiu dos avanços de várias áreas como sistemas embarcados, microeletrônica, comunicação e sensoriamento. De fato, a IoT tem recebido bastante atenção tanto da academia quanto da indústria, devido ao seu potencial de uso nas mais diversas áreas das atividades humanas. A IoT tem alterado aos poucos o conceito de redes de computadores, neste sentido, é possível notar a evolução do conceito ao longo do tempo (SANTOS et al., 2016).

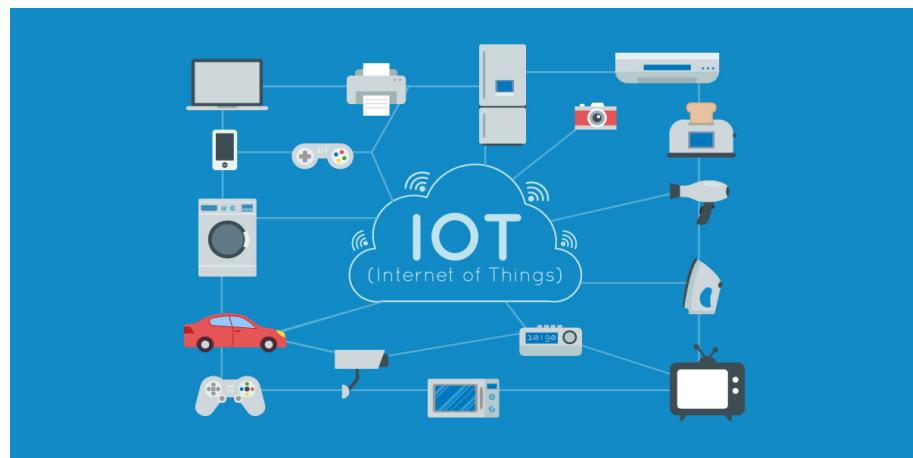


Figura 6 – Internet das coisas no setor de serviços  
Fonte: TECH (2021)

A Internet das Coisas, em poucas palavras, nada mais é que uma extensão da Internet atual, que proporciona aos objetos do dia-a-dia (quaisquer que sejam), mas com capacidade computacional e de comunicação, se conectarem à Internet. A conexão com a rede mundial de computadores viabilizará, primeiro, controlar remotamente os objetos e, segundo, permitir que os próprios objetos sejam acessados como provedores de serviços. Estas novas habilidades, dos objetos comuns, geram um grande número de oportunidades tanto no âmbito acadêmico quanto no industrial (SANTOS et al., 2016).

Nos últimos anos, a Internet das Coisas (IoT) emergiu como uma das tecnologias mais importantes do século XXI. Com a capacidade de conectar objetos do cotidiano, como eletrodomésticos, carros, termostatos e babás eletrônicas, à Internet por meio de dispositivos incor-

porados, tornou-se possível uma comunicação perfeita entre pessoas, processos e objetos. Utilizando computação de baixo custo, nuvem, big data, análise avançada e tecnologias móveis, objetos físicos agora podem compartilhar e coletar dados com mínima intervenção humana. Nesse mundo hiperconectado, sistemas digitais conseguem registrar, monitorar e ajustar cada interação entre itens conectados, permitindo uma integração harmoniosa entre o mundo físico e o digital (ORACLE, 2024).

## 2.4 MICROCONTROLADORES

Microcontroladores são circuitos integrados que possuem em seu interior todos os componentes necessário ao seu funcionamento dependendo unicamente da fonte de alimentação externa. Pode-se dizer que os microcontroladores são computadores de um único chip. Um sistema microprocessado é composto por uma unidade central de processamento CPU e um conjunto de periféricos necessários os seu funcionamento, na Figura 7 mostra os componentes de um microcontrolador. Dentre este periféricos podemos destacar a memória de dados, a memória de programa e o circuito de clock. Os microcontroladores diferem dos sistemas tradicionais por já integrarem os seus periféricos dentro do próprio componente (KERSCHBAUMER et al., 2013).

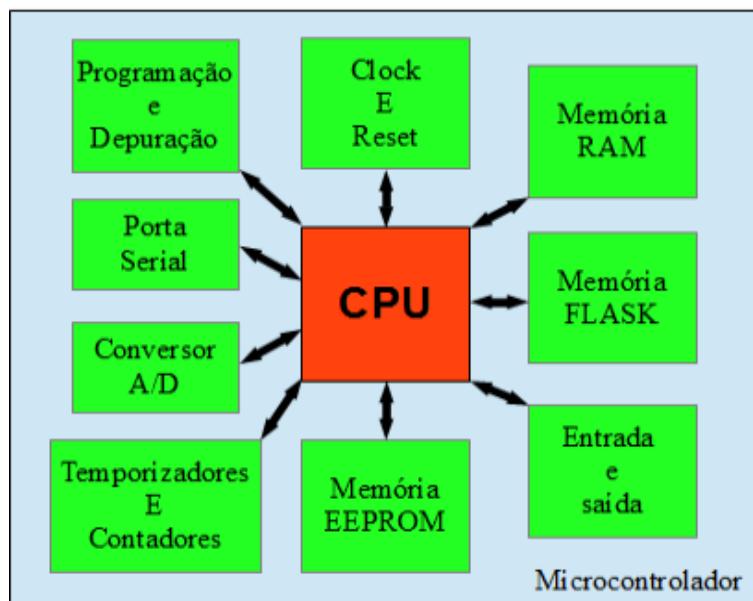


Figura 7 – Componentes de um microcontrolador  
Fonte: (KERSCHBAUMER et al., 2013).

O microcontrolador Wemos Lolin ESP32 é o dispositivo responsável por coordenar a

ilha de medição de grandezas elétricas. Ele desempenha várias funções cruciais, incluindo ligar e desligar dispositivos elétricos conectados à ilha através de relés, além de ler e interpretar dados de tensão e corrente presentes no circuito quando algum dispositivo elétrico está em funcionamento. A partir desses dados, o ESP32 realiza os cálculos de potência necessários e gera as informações de interesse do usuário. Com suas capacidades avançadas de processamento e conectividade, o ESP32 não só monitora e controla os dispositivos elétricos, mas também fornece uma interface eficiente para a coleta e análise de dados, facilitando a gestão de energia e melhorando a eficiência operacional (AMORIM JR; RIBEIRO; COLISTETE JR, 2019).

O microcontrolador é a integração do microprocessador com interfaces de entrada/saída (I/O) e outros periféricos, como temporizadores, em um único chip. Com quantidade limitada de RAM e ROM, os microcontroladores são amplamente usados em sistemas de controle embarcados, como o Arduino. Surgido na Itália em 2005, o Arduino foi criado para facilitar o controle de projetos e protótipos de maneira mais acessível do que os sistemas disponíveis na época. A plataforma Arduino é Open-source, baseada em hardware e software, e é amplamente utilizada em automação e robótica, permitindo a adição de diversos componentes eletrônicos programáveis para atividades específicas (TARGA; SILVA; CEZAR, 2019).

## 2.5 INSTITUTO FEDERAL FARROUPILHA - CAMPUS S.V.S (IFFAR - SVS)

O IFFAR-SVS, localizado no município de São Vicente do Sul, RS, desempenha um papel crucial na promoção da educação técnica e profissional na região. Fundado em 17 de novembro de 1954 como Escola de Iniciação Agrícola, o campus tem uma longa história de comprometimento com o desenvolvimento educacional e socioeconômico da comunidade local. Ao longo dos anos, o instituto passou por várias transformações e adaptações para atender às necessidades em evolução da agricultura, tecnologia e indústria. Oferecendo uma variedade de cursos técnicos, de graduação e pós-graduação, o Campus São Vicente do Sul do Instituto Federal Farroupilha continua a ser uma fonte vital de educação e inovação na região, preparando os estudantes para enfrentar os desafios do mercado de trabalho atual e futuro (IFFAR, 2024a).

No âmbito do IFFAR-SVS, a agricultura desempenha um papel significativo, com destaque para o uso de estufas e hortas como recursos pedagógicos e de pesquisa. As estufas proporcionam um ambiente controlado para o cultivo de uma variedade de plantas, permitindo experimentos científicos, estudos práticos e produção agrícola em condições ideais. Além disso, as hortas são utilizadas como espaços de aprendizagem prática, onde os estudantes podem ad-

quirir habilidades em cultivo, manejo de culturas e práticas agrícolas sustentáveis. O Campus São Vicente do Sul busca integrar o ensino teórico com a experiência prática, proporcionando aos alunos uma formação abrangente e preparando-os para contribuir de forma significativa para o setor agrícola e para a comunidade em geral (IFFAR, 2024b).

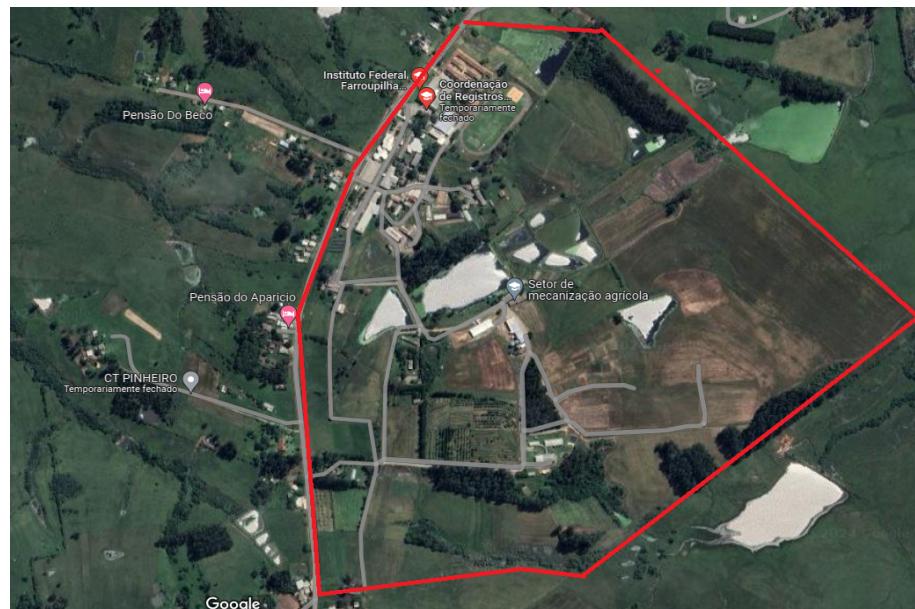


Figura 8 – Localização do IFFAR - SVS

Fonte: Google Maps.

### 2.5.1 Setor de Horticultura

O setor de Olericultura e Hortaliças, é uma área dedicada ao estudo e à prática da produção de hortaliças e vegetais. O termo "olericultura" refere-se à ciência que estuda a produção de hortaliças em geral, enquanto "hortaliças" abrange uma ampla variedade de plantas cultivadas para consumo humano, como alface, tomate, cenoura, entre outras. No setor são desenvolvidas atividades relacionadas ao cultivo de hortaliças em diferentes sistemas, como o cultivo convencional, hidropônico, orgânico, entre outros. São realizadas diversas etapas do processo produtivo, desde o planejamento e preparo do solo até a colheita e comercialização dos produtos (IFFAR, 2024b).

## 2.6 TRABALHOS RELACIONADOS

No projeto de TORRES et al. (2021), é proposta a implementação de um sistema de sensoriamento em Internet das Coisas (IoT) para monitoramento remoto da umidade, temperatura

e umidade do solo em pequenas plantações. Utilizando dispositivos como o sensor DHT11, sensor de umidade do solo e a placa TTGO T-beam com chip LoRa e protocolo LoRaWAN, o projeto demonstrou viabilidade por meio de testes que evidenciaram uma resposta satisfatória e acesso remoto em tempo real. No entanto, sugere-se melhorias como a automação da irrigação e a unificação dos circuitos, visando aprimorar a eficiência e a praticidade do sistema.

HENRIQUES et al. (2021) foca no desenvolvimento e implementação de um sistema de irrigação automatizada, baseado nos princípios da IoT. Utilizando o protocolo de comunicação MQTT e o microcontrolador Esp-wroom-32, o sistema monitora dados de umidade do solo e temperatura ambiente através de sensores em um plantio caseiro. Esses dados são transmitidos para a plataforma IoT TagoIO, onde são visualizados em um dashboard através de gráficos, permitindo um acompanhamento visual das condições do cultivo. No entanto, seria interessante explorar como a automação da irrigação poderia ser integrada ao sistema, proporcionando uma solução mais abrangente para a gestão do cultivo.

O trabalho de BENTES et al. (2023) apresenta uma etapa inicial de desenvolvimento de uma ferramenta destinada a facilitar o controle, monitoramento e automação do cultivo de plantas em residências. Por meio de uma aplicação móvel para Android, os usuários podem coletar em tempo real dados provenientes de sensores de um dispositivo embarcado (Esp32) associado a uma planta específica. Esses sensores monitoram a luminosidade, umidade, temperatura e umidade do solo, enquanto a aplicação Android exibe esses dados de forma simples e intuitiva. No entanto, seria interessante explorar como a plataforma poderia ser expandida para atender a cultivos em maior escala, como estufas comerciais, e se há possibilidade de integração com outros sistemas de monitoramento e controle.

O trabalho de SALINDO et al. (2023) apresenta um protótipo de sistema de irrigação automatizado de baixo custo, desenvolvido com a plataforma Arduino Uno, que é uma placa microcontrolada por ATMEGA328. Ele discute a importância da água na irrigação e nas plantações, descrevendo as características dos componentes e do software. O sistema monitora a umidade do solo com um sensor de baixo custo e aciona uma bomba d'água por meio de um relé. O protótipo é acessível, custando cerca de R\$ 500,00, e pode ser personalizado com sensores adicionais. Seu funcionamento adequado comprova a viabilidade do Arduino para automatizar a irrigação a um custo acessível.

Elias Naves SOLDÁ et al. (2023) propõe um sistema autônomo de irrigação para hortaliças, visando aprimorar o uso eficiente da água. O sistema utiliza um Kit de desenvolvimento

Arduino, incluindo uma central de processamento e válvula solenoide para controle de vazão, juntamente com sensores de umidade, temperatura e luminosidade (LDR). Dois hidrômetros foram adicionados à linha de irrigação para uma medição precisa. O protótipo foi testado no Sítio Bela Vista em Nova Canaã Paulista, com dados coletados ao longo de um ciclo de produção de 35 dias. Os resultados indicam que o sistema automatizado de irrigação pode manter a umidade do solo em níveis ideais, contribuindo para uma produção de hortaliças de melhor qualidade.

Os cinco trabalhos abordam temas similares, porém com soluções distintas. Enquanto o primeiro utiliza a placa TTGO T-beam com LoRa e a plataforma ProIoT, o segundo emprega o Esp-wroom-32 e a plataforma TagoIO, o terceiro utiliza o Esp32 e um aplicativo Android como plataforma, o quarto busca desenvolver com um baixo custo e o quinto propõe um sistema autônomo de irrigação para hortaliças, visando aprimorar o uso eficiente da água. O projeto em desenvolvimento propõe a utilização do LoRa para o envio de dados entre os Esp32, os quais serão posteriormente enviados para um site web. Este sistema será projetado para monitorar e controlar plantas utilizando o alcance que o LoRa fornece.

Diferente ou semelhantemente aos trabalhos citados, este trabalho em desenvolvimento propõe a utilização do LoRa para o envio de dados entre os Esp32, os quais serão posteriormente enviados para um site web. Este sistema será projetado para monitorar e controlar plantas utilizando o alcance que o LoRa fornece.

### 3 MATERIAIS E MÉTODOS

Este capítulo descreve todos os recursos tecnológicos e as abordagens metodológicas adotados para a elaboração do presente trabalho.

#### 3.1 MATERIAIS

Nesta seção, é apresentado a linguagem de programação, o sistema gerenciador de banco de dados, o microprocessador e a linguagem de marcação a serem usados.

##### 3.1.1 Heltec WiFi LoRa 32 WiFi V2

O Heltec WiFi LoRa 32 V2 é uma placa de desenvolvimento IoT robusta da Heltec Automation, popular desde seu lançamento em 2017. Equipado com o microprocessador ESP32-S3FN8 de dois núcleos até 240 MHz, oferece WiFi, BLE, LoRa e display OLED. Ideal para aplicações em cidades inteligentes, fazendas e controle industrial, possui interface USB tipo C com proteção contra ESD e curto-círcuito, além de gerenciamento de bateria de lítio integrado. Com suporte ao ambiente de desenvolvimento Arduino e um design otimizado para baixo consumo de energia e desempenho de circuito RF (AUTOMATION, 2023), na figura 9 apresenta o diagrama da placa.

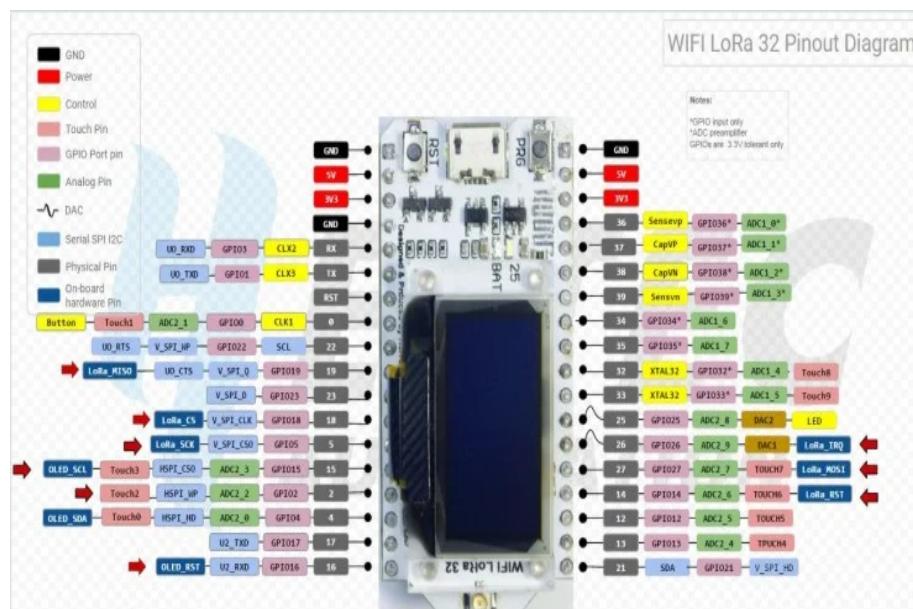


Figura 9 – Diagrama da Placa  
Fonte: (AUTOMATION, 2023)

O Heltec WiFi LoRa 32 V2 será utilizado para receber dados dos módulos de sensores, como temperatura, umidade e umidade do solo. Estes dados serão enviados tanto para o ThingsBoard quanto para o XAMPP utilizando os protocolos MQTT e HTTP. O uso do MQTT garantirá uma comunicação eficiente e confiável com o ThingsBoard, enquanto o HTTP será utilizado para enviar os dados para o XAMPP, onde serão armazenados e gerenciados no banco de dados MySQL. Essa integração permitirá monitorar e analisar os dados em tempo real, possibilitando a tomada de decisões baseadas em informações precisas e atualizadas.

### 3.1.2 DHT-11

O DHT11 é amplamente reconhecido como um sensor confiável para medição de temperatura na faixa de 0 a 50 °C e umidade de 20 a 90%. Este componente versátil é especialmente popular em projetos de automação residencial, monitoramento ambiental e controle climático devido às suas especificações notáveis. Com uma precisão de  $\pm 2^{\circ}\text{C}$  para temperatura e  $\pm 5\%$  para umidade, o DHT11 oferece medições precisas e consistentes. Compatível com plataformas como Arduino, Raspberry Pi e ESP32, o sensor é de fácil integração e possui uma interface digital que simplifica a leitura de dados (MAKERHERO®, 2023).

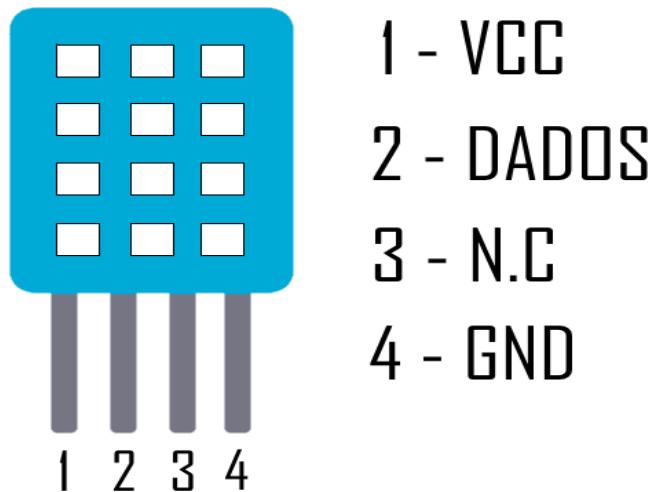


Figura 10 – DHT11  
Fonte: (MAKERHERO®, 2024)

O DHT11 será empregado para capturar dados precisos de temperatura e umidade do ar no Heltec WiFi LoRa 32 V2. A integração do DHT11 com o Heltec WiFi LoRa 32 V2, combi-

nada com o uso de protocolos como MQTT e HTTP, garantirá uma coleta e transmissão eficaz desses dados para plataformas como ThingsBoard e XAMPP, onde poderão ser visualizados e analisados pelos usuários finais.

### 3.1.3 Módulo Sensor de Umidade do Solo

Um sensor de umidade do solo é um dispositivo projetado para medir a quantidade de umidade presente no solo, como mostrado na Figura 11. Amplamente utilizado em aplicações como irrigação de solo, hortas, plantações, estufas e na indústria agrícola e agropecuária, esses sensores requerem que uma parte esteja enterrada no solo para funcionar corretamente. Eles operam detectando a resistividade elétrica do solo, o que varia conforme a umidade. Quanto menor a umidade do solo, maior será o sinal de saída do sensor, e vice-versa. Isso permite ajustes precisos no controle de irrigação, garantindo que as plantas recebam a quantidade ideal de água para um crescimento saudável (ALVES, 2023a).

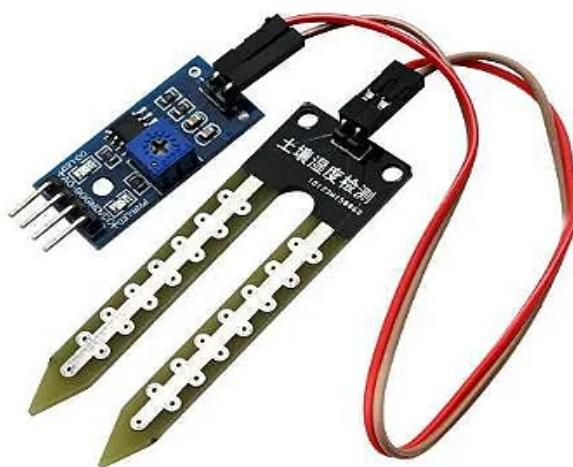


Figura 11 – Sensor de umidade do solo

Fonte: Elaborado pelo autor

O módulo sensor de umidade será empregado para capturar dados precisos de umidade do solo no Heltec WiFi LoRa 32 V2. A integração deste sensor permitirá monitorar continuamente as condições do solo, enviando os dados coletados através do protocolo MQTT para plataformas como ThingsBoard e XAMPP.

### 3.1.4 Long Range (LoRa)

A tecnologia LoRa, é uma técnica de modulação de espetro proprietária desenvolvida pela Semtech de baixo consumo e longo alcance, que se baseia na modulação CSS o que lhe confere robustez contra ruído e um maior alcance mantendo um baixo consumo energético, fazendo parte das tecnologias de LPWAN. Esta modulação troca ritmo binário por sensibilidade dentro de uma largura de banda fixa num canal. Ao aumentar a duração de símbolo com o aumento do SF o ritmo binário diminui, mas possibilita a operação a menores potências e a longas distâncias. O alcance pode variar significativamente dependendo de vários fatores, como o ambiente, a potência de transmissão, a altura das antenas e as condições atmosféricas. Em ambientes urbanos, o alcance é de cerca de 2 a 5 km, áreas rurais de 10 a 15 km e condições ideais pode chegar até 20 km ou mais (ALVES, 2023b).

No projeto, a comunicação LoRa será utilizada para fazer a troca de informações entre os módulos Heltec ESP 32 LoRa WiFi V2. Um dos módulos enviará os dados capturados dos sensores, enquanto o outro receberá esses dados, tomará a decisão de quando iniciar a irrigação e enviará os dados recebidos para os painéis de controle via MQTT e HTTP.

### 3.1.5 Arduino IDE

Arduino é uma plataforma de computação física de código aberto baseada em uma placa de E/S simples e um ambiente de desenvolvimento que implementa a linguagem Processing/Wiring. O Arduino pode ser usado para desenvolver objetos interativos independentes ou pode ser conectado a software no seu computador (por exemplo, Flash, Processing e MaxMSP) (ARDUINO MASSIMO BANZI; MELLIS, 2022). O Arduino IDE como mostrado na Figura 12, utilizado para programar essas placas, é baseado na estrutura Theia IDE e construído com Electron. As operações de back-end, como compilação e upload, são gerenciadas por uma instância do arduino-cli em execução no modo daemon, o que permite uma interação mais eficiente com o hardware (ARDUINO; KITTAAKOS, 2024).

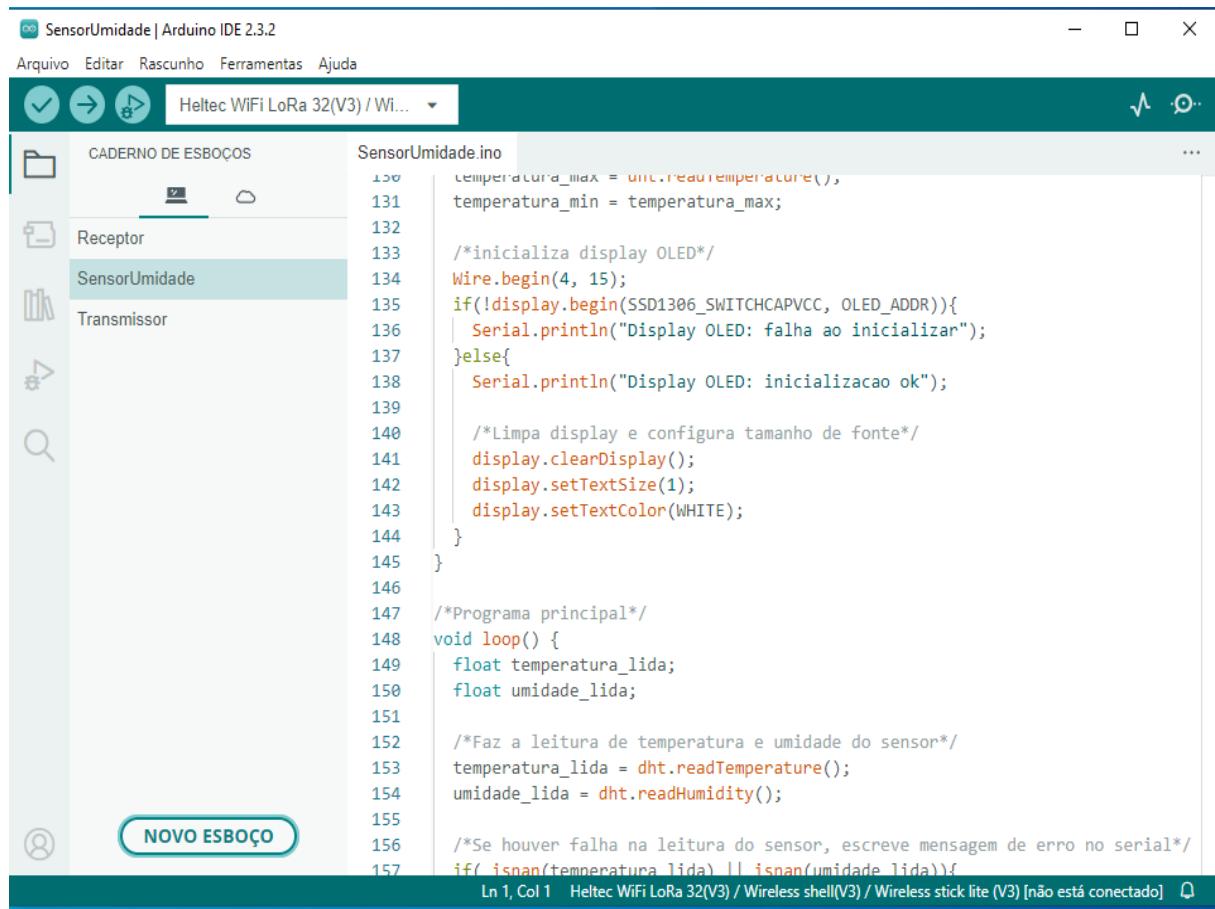


Figura 12 – Interface do Arduino IDE

Fonte: Elaborado pelo autor

Na Figura 12 é apresentada a interface da IDE, que será utilizada para desenvolvimento dos códigos para as placas e gerenciadas as bibliotecas para os módulos que serão utilizados no TCC. A IDE é utilizada para programar as placas e os módulos, calibrar os dados recebidos dos sensores e módulos no microprocessador, e realizar a depuração dos projetos. Além disso, a IDE do Arduino oferece uma interface amigável que facilita o desenvolvimento e a prototipagem rápida, sendo uma ferramenta essencial tanto para iniciantes quanto para profissionais no campo da eletrônica e programação embarcada.

### 3.1.6 C++

C++ é uma extensão da linguagem de programação C. As extensões de C++ sobre C foram introduzidas por Bjarne Stroustrup em 1980 nos Laboratórios Bell de New Jersey. Inicialmente, a linguagem era chamada "C com classes", mas o nome foi alterado para C++ em 1983. A motivação para o desenvolvimento de C++ foi a complexidade. Grandes sistemas

implementados com a linguagem C, com códigos na ordem de 25.000 a 100.000 linhas, são difíceis de controlar ou mesmo entender em sua totalidade. C++ surgiu para permitir que essa barreira fosse quebrada. O objetivo de C++ é permitir que programadores possam gerenciar e compreender programas maiores e mais complexos (RICARTE, 1995).

A linguagem C++ será utilizada para programar microcontroladores e módulos na IDE Arduino. A escolha de C++ se deve à sua eficiência e flexibilidade, permitindo um controle preciso do hardware. Com C++, será possível desenvolver funcionalidades avançadas e personalizadas para os dispositivos ESP32, como a coleta de dados de sensores de temperatura, umidade e umidade do solo.

### **3.1.7 MQ Telemetry Transport (MQTT)**

Criado em 1999 pela IBM, o MQTT é um protocolo aberto de mensagens para comunicação M2M, projetado para lidar com alta latência, instabilidade e baixa largura de banda. Padronizado pelo OASIS em 2013, está na versão 3.1.1 desde 2014 e é livre de royalties desde 2010. O MQTT usa o protocolo TCP e o padrão publisher/subscriber, onde um broker intermedia o envio de mensagens aos destinatários. Essa estrutura desacopla o produtor do cliente, permitindo comunicações de um para um, um para muitos ou muitos para muitos (TORRES; ROCHA; SOUZA, 2016).

O protocolo MQTT é uma peça fundamental no projeto, pois será utilizado para conectar ao ThingsBoard, facilitando o envio dos dados capturados pelos microprocessadores ESP32. Essa abordagem permitirá uma comunicação eficiente e em tempo real entre os dispositivos e a plataforma de IoT, garantindo que os dados de temperatura, umidade e umidade do solo sejam transmitidos de forma confiável e rápida.

### **3.1.8 ThingsBoard**

ThingsBoard é uma plataforma de código aberto para IoT que simplifica o desenvolvimento, gerenciamento e dimensionamento de projetos de Internet das Coisas, como mostrado na Figura 13. Oferece uma infraestrutura completa tanto para implementações na nuvem quanto localmente, permitindo a conexão, coleta e visualização de dados de dispositivos IoT. Com recursos avançados como provisionamento de dispositivos, análise de telemetria, criação de painéis interativos e integração flexível com outros sistemas, o ThingsBoard possibilita a criação de soluções robustas e personalizadas para uma variedade de aplicações IoT (AUTHORS,

2023).

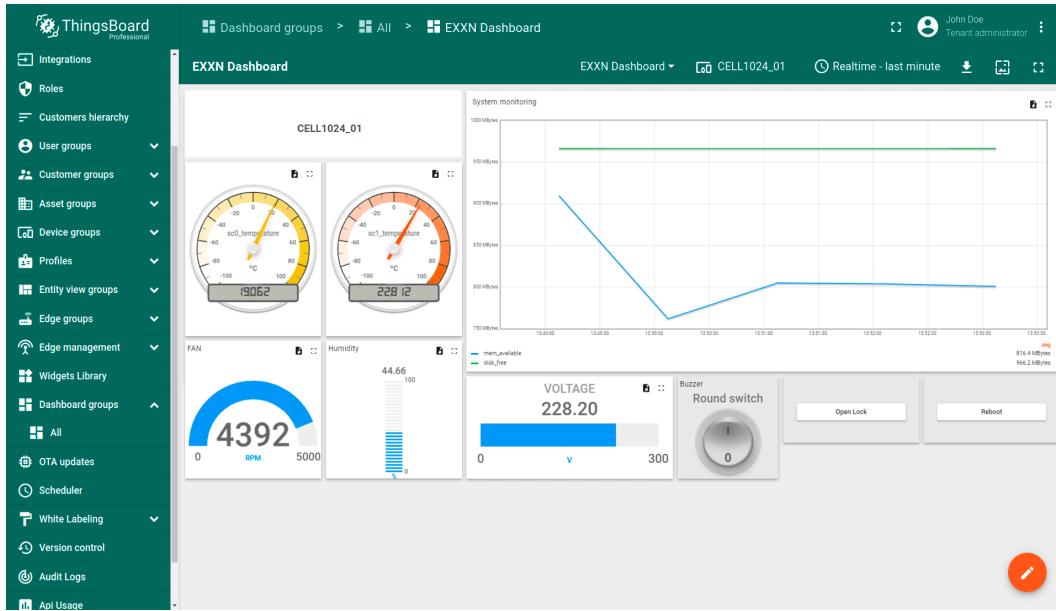


Figura 13 – Interface do ThingsBoard

Fonte: Elaborado pelo autor

O ThingsBoard será utilizado dashboard para exibir os dados enviados através do MQTT pelo ESP32 para o usuário. Com o ThingsBoard, será possível criar painéis interativos e personalizados que mostram informações como temperatura, umidade e umidade do solo em tempo real. A integração com o MQTT garante uma comunicação eficiente e confiável entre os dispositivos IoT e a plataforma, garantindo uma experiência de usuário fluida e informativa.

### 3.1.9 Hypertext Transfer Protocol (HTTP)

O HTTP é o principal protocolo responsável pela transferência de dados na internet. Ele estabelece as bases necessárias para a conexão entre um cliente e um servidor, permitindo a comunicação e troca de informações. Funcionando no modelo computacional cliente-servidor, o HTTP facilita a solicitação de recursos, como páginas web, de um servidor por um cliente, geralmente um navegador. A versão mais recente, HTTP/2, introduz melhorias significativas em desempenho e eficiência, incluindo multiplexação de streams e compressão de cabeçalhos, tornando a navegação na web mais rápida e eficiente (SANTANA, 2023).

No projeto, o protocolo HTTP será utilizado para conectar o banco de dados MySQL, parte do XAMPP, aos dados enviados pelos ESP32, que capturam informações dos módulos de sensor DHT11 (temperatura e umidade) e de umidade do solo.

### **3.1.10 XAMPP**

XAMPP é um pacote de software livre que inclui um servidor web Apache, um banco de dados MySQL, PHP e Perl. Foi desenvolvido para ser uma solução fácil de instalar e configurar, proporcionando aos usuários um ambiente de desenvolvimento web completo em seus computadores locais. O objetivo do XAMPP é simplificar esse processo, oferecendo uma distribuição de fácil instalação para desenvolvedores que desejam entrar no mundo do Apache. Ele é gratuito, livre para ser copiado sob os termos da GNU General Public Licence, embora apenas a compilação do XAMPP seja publicada sob a licença GPL (SEIDLER, 2023).

O XAMPP será utilizado para armazenar os dados enviados pelo microprocessador ESP32 através do protocolo HTTP. Além disso, será empregado para consultas ao banco de dados por meio do PHP, com o objetivo de exibir esses dados em um site desenvolvido em HTML, JS, CSS e PHP.

### **3.1.11 PHP**

O PHP é uma linguagem de programação para a criação de scripts web do lado servidor embutidos em HTML. É uma linguagem de código-fonte aberto e compatível com os mais importantes servidores web, especialmente o Apache. O PHP permite incorporar fragmentos de código em páginas HTML comuns, que são interpretados à medida que essas páginas são oferecidas aos usuários. Além disso, o PHP funciona como uma linguagem de "cola", facilitando a conexão das páginas web com o banco de dados do lado servidor (CONVERSE; PARK, 2003).

No TCC, o PHP será utilizado para fazer a conexão ao banco de dados no XAMPP, permitindo realizar operações de SELECT, INSERT, UPDATE e DELETE. Isso garantirá que os dados capturados pelos dispositivos ESP32 sejam armazenados, manipulados e exibidos de forma eficiente e dinâmica na aplicação web. Além disso, a utilização do PHP permitirá a criação de funcionalidades interativas, como formulários de entrada de dados e consultas personalizadas, melhorando a experiência do usuário e a funcionalidade geral do sistema.

### **3.1.12 MySQL**

O MySQL é um servidor e gerenciador de banco de dados (SGBD) relacional, de licença dupla (sendo uma delas de software livre), projetado inicialmente para trabalhar com aplicações de pequeno e médio porte, mas atualmente atendendo também a aplicações de grande porte,

oferecendo diversas vantagens em relação aos seus concorrentes. Possui todas as características necessárias para um banco de dados de grande porte e é reconhecido por algumas entidades como o banco de dados open source com maior capacidade para competir com programas similares de código fechado, como o SQL Server (da Microsoft) e o Oracle (MILANI, 2007).

Neste projeto, o MySQL será utilizado no XAMPP, já que é um dos produtos oferecidos por essa plataforma. No MySQL, serão armazenados os dados enviados pelos dispositivos ESP32 e realizadas consultas para exibir essas informações no HTML. A utilização do MySQL no XAMPP permitirá uma gestão eficiente e segura dos dados, garantindo a integridade e a disponibilidade das informações para serem visualizadas na página web.

### **3.1.13 Hyper Text Markup Language (HTML)**

HTML que, em português, significa linguagem de marcação de hipertexto. O hipertexto pode ser resumido como todo o conteúdo inserido em um documento para a web, com a principal característica de permitir a interligação a outros documentos na web. Essa interligação é viabilizada pelos links, que são elementos presentes nas páginas dos sites que costumamos visitar quando navegamos na internet (SILVA, 2008).

Neste projeto, o HTML será utilizado para estruturar e inserir informações provenientes das consultas ao banco de dados, bem como o texto adicionado pelos usuários. Com o HTML, poderemos definir a organização do conteúdo, utilizando elementos como parágrafos, cabeçalhos, tabelas, listas, e formulários. Isso permitirá uma apresentação clara e organizada das informações, facilitando a interação dos usuários com os dados exibidos.

### **3.1.14 Cascading Style Sheets (CSS)**

O CSS é uma das ferramentas mais amplamente utilizadas atualmente. O termo CSS significa Folhas de Estilos em Cascata, e trata-se de uma linguagem de estilo essencial para a construção de layouts de páginas e sites. Permite que você projete páginas e sistemas utilizando uma técnica completamente diferente da convencional, possibilitando uma considerável redução no tempo de trabalho. O CSS é utilizado para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML. Seu principal benefício é a separação entre o formato e o conteúdo de um documento (JOBSTRAIBIZER, 2009).

Nesse projeto, o CSS será empregado para a estilização da página web utilizada para exibir as consultas do banco de dados. Com o CSS, poderemos definir a aparência visual dos

elementos da página, como cores, fontes, margens, espaçamentos e layouts. Em resumo, o CSS será fundamental para garantir uma apresentação profissional e eficiente dos dados capturados e exibidos pelo sistema.

### 3.1.15 JavaScript (JS)

JavaScript é a linguagem de programação da Web. A vasta maioria dos sites modernos utiliza JavaScript, e todos os navegadores modernos — em computadores de mesa, consoles de jogos, tablets e smartphones — incluem interpretadores JavaScript, tornando-a a linguagem de programação mais onipresente da história. JavaScript faz parte da tríade de tecnologias que todos os desenvolvedores Web devem conhecer: HTML, para especificar o conteúdo das páginas Web; CSS, para especificar a apresentação dessas páginas; e JavaScript, para especificar o comportamento delas (FLANAGAN, 2012).

Neste trabalho, utilizaremos JavaScript juntamente com HTML e CSS para a criação do site. Nele, serão exibidos os dados de temperatura, umidade e umidade do solo, capturados pelo ESP32 e consultados no banco de dados do XAMPP.

## 3.2 MÉTODOS

Nesta seção, serão apresentados o processo de coleta de requisitos, a arquitetura selecionada, os diagramas elaborados e os requisitos do sistema. Cada um desses elementos desempenha um papel crucial no desenvolvimento e implementação bem-sucedida do projeto.

### 3.2.1 Arquitetura Model-View-Controller (MVC)

A arquitetura conhecida como MVC surgiu no começo da década de 1980 e se tornou amplamente popular na criação de aplicações web. Sua dinâmica é simples: todas as requisições da aplicação são direcionadas para a camada Controller, que acessa a camada Model para processar a solicitação e, por fim, exibe o resultado na camada View. O padrão MVC separa claramente as responsabilidades de apresentação, lógica de negócio e controle de fluxo da aplicação, o que aumenta significativamente a capacidade de reutilização e facilita a manutenção do projeto (LUCIANO; ALVES, 2017).

No projeto em questão, o padrão MVC será adotado para gerenciar o desenvolvimento tanto do site web quanto dos códigos destinados aos ESP32. Isso permite uma organização

estruturada do sistema, onde a interface de usuário (View), a lógica de processamento e manipulação de dados (Model) e o controle de interações e fluxo da aplicação (Controller) estão claramente separados. Essa abordagem não só promove um código mais limpo e modular, como também facilita a integração de novos recursos e a manutenção contínua do sistema como um todo.

### **3.2.2 Unified Modeling Language (UML)**

A UML é uma linguagem gráfica amplamente utilizada para a visualização, especificação, construção e documentação de artefatos em sistemas de software complexos. Ela oferece uma maneira padronizada de criar planos de arquitetura de sistemas, abrangendo tanto aspectos conceituais, como processos de negócios e funções do sistema, quanto elementos concretos, como classes em linguagens de programação específicas, esquemas de bancos de dados e componentes de software reutilizáveis (BOOCH, 2006). A UML facilita a comunicação entre membros da equipe e garante que todos tenham uma compreensão clara e unificada do sistema em desenvolvimento.

No desenvolvimento deste projeto, decidiu-se pela utilização de diversos diagramas da UML, incluindo o Diagrama de Caso de Uso, o Diagrama de Classes e o Diagrama de Entidade-Relacionamento. Para isso, empregou-se a ferramenta Astah UML e LucidChart, que oferece recursos robustos para a criação e manipulação desses diagramas. O uso desses diagramas e ferramentas auxilia na organização e clareza do projeto, permitindo uma análise detalhada e estruturada dos requisitos e da arquitetura do sistema, o que é fundamental para o sucesso do desenvolvimento.

### **3.2.3 Diagrama de Implantação**

O diagrama de implantação determina as necessidades de hardware do sistema, como servidores, estações, topologias e protocolos de comunicação, ou seja, todo o aparato físico sobre o qual o sistema deverá ser executado. Esse diagrama permite demonstrar também como se dará a distribuição dos módulos do sistema, em situações em que estes forem executados em mais de um servidor (GUEDES, 2009). São empregados para a modelagem da visão estática da implantação de um sistema. Na maior parte, isso envolve a modelagem da topologia do hardware em que o sistema é executado. Os diagramas de implantação são essencialmente diagramas de classes que focalizam os nós do sistema. Os diagramas de implantação não são importantes

somente para visualizar, especificar e documentar sistemas embarcados, cliente/servidor e distribuídos, mas também para o gerenciamento de sistemas executáveis por meio de engenharia direta e reversa (BOOCH, 2006).

O diagrama de implatação feito para o projeto de mostra como se dará a ligação e comunicação com os hardware e software do sistema, representado na Figura 14.

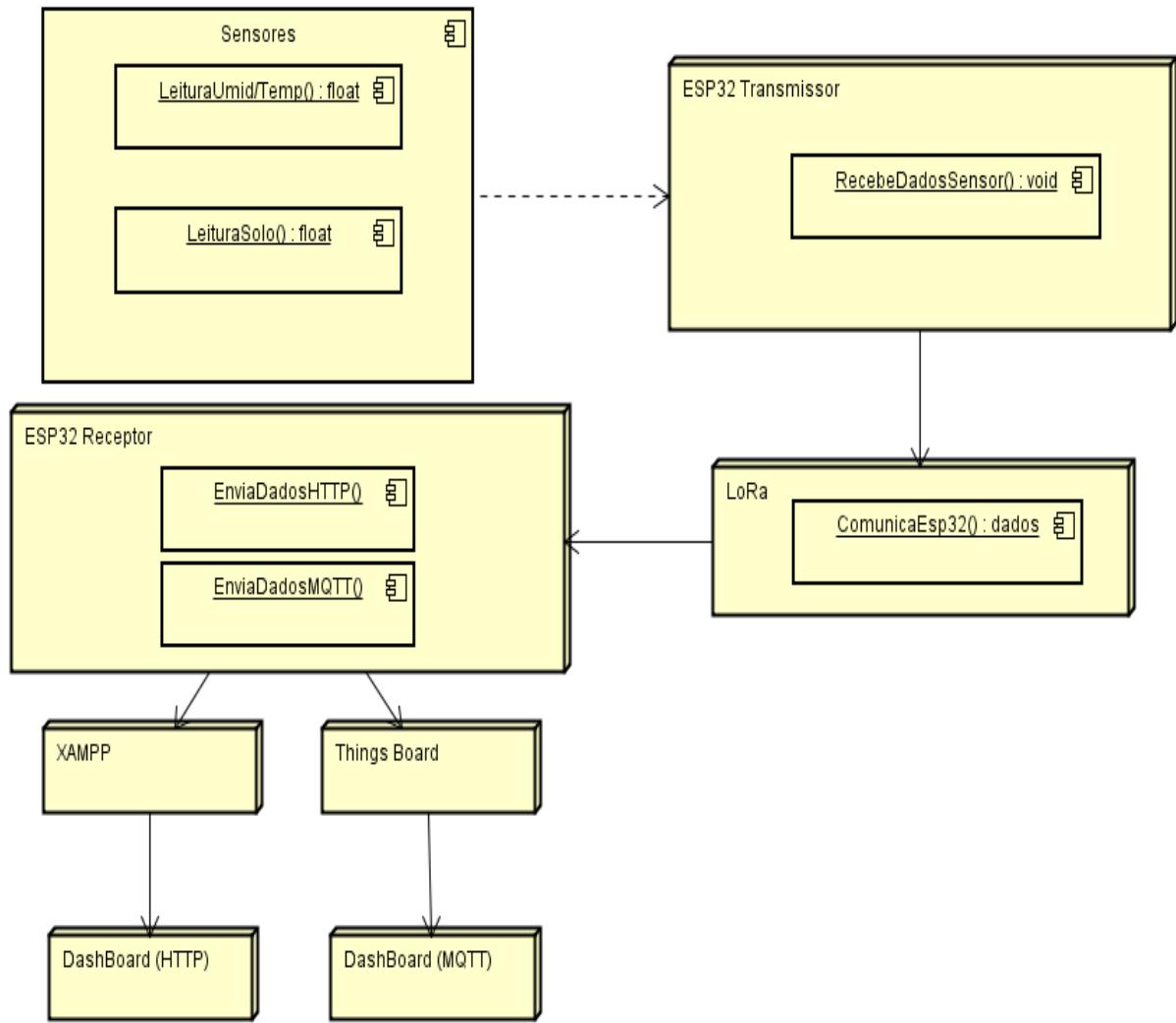


Figura 14 – Diagrama de Implatação

Fonte: Elaborado pelo autor

### 3.2.4 Diagrama de Caso de Uso

O diagrama de casos de uso tem por objetivo apresentar uma visão externa geral das funcionalidades que o sistema deverá oferecer aos usuários, sem se preocupar detalhadamente com a forma de implementação dessas funcionalidades. Este diagrama é amplamente utilizado nas fases de elicitação e análise de requisitos do sistema, mas também é consultado ao longo

de todo o processo de modelagem, servindo como base para a criação de diversos diagramas. Ele busca utilizar uma linguagem simples e de fácil compreensão, permitindo que os usuários tenham uma ideia clara de como o sistema vai se comportar. O diagrama de casos de uso identifica os atores (usuários, outros sistemas ou até mesmo hardware específico) que interagirão com o software e os serviços, ou seja, as funcionalidades que o sistema disponibilizará a esses atores, conhecidas como casos de uso (GUEDES, 2018).

No desenvolvimento deste projeto, foram criados quatro diagramas de caso de uso, conforme ilustrado nas Figuras 15, 16, 17 e 18, para especificar os requisitos do sistema de maneira clara e objetiva. Nos diagramas principais, representados nas Figuras 15 e 16, são identificados cinco atores: os sensores, o ESP32 transmissor, o ESP32 receptor, o XAMPP e o ThingsBoard. Os sensores são responsáveis por capturar os dados do solo, da temperatura e da umidade. O ESP32 transmissor recebe esses dados e os envia através do LoRa. O ESP32 receptor, por sua vez, recebe os dados do transmissor, decide se a plantação precisa ser irrigada e encaminha os dados para o XAMPP e o ThingsBoard, que exibem as informações de forma acessível e detalhada.

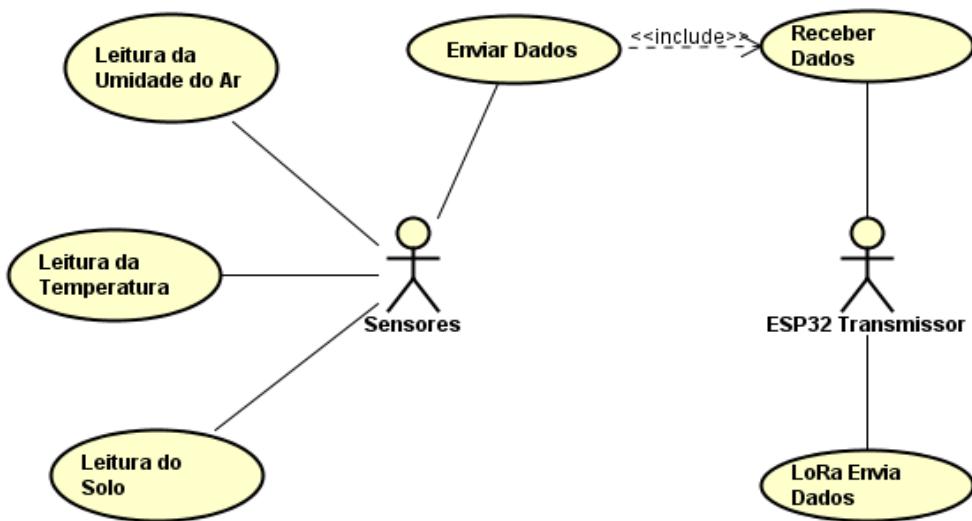


Figura 15 – Diagrama do ESP32 Transmissor  
Fonte: Elaborado pelo autor

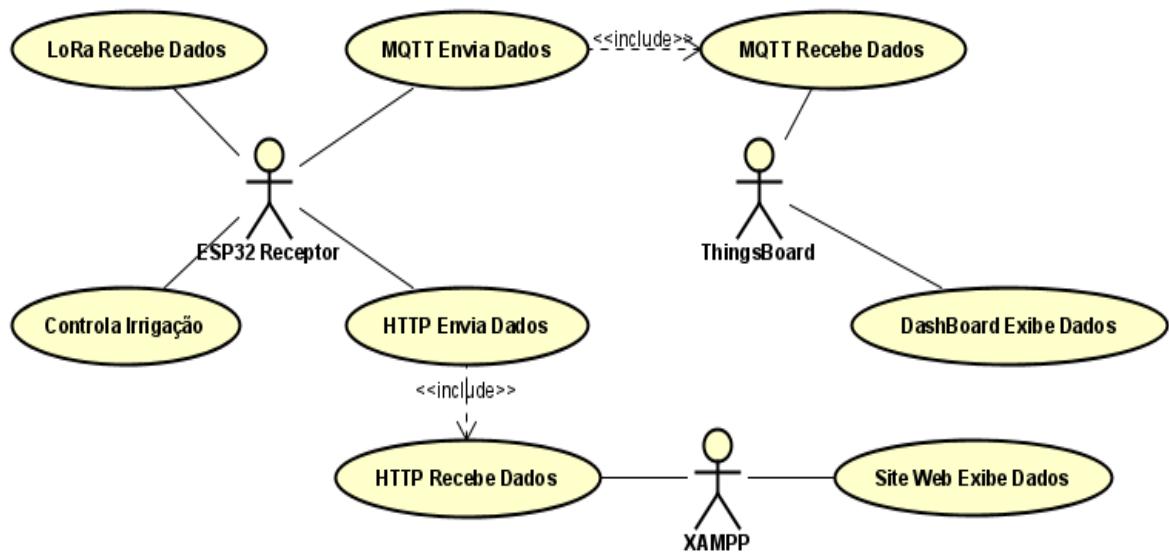


Figura 16 – Diagrama do ESP32 Receptor

Fonte: Elaborado pelo autor

No diagrama Usuário presente na figura 17, o usuário tem a possibilidade de realizar operações como logar, visualizar os dados, pesquisar os dados e fazer anotações sobre os dados.

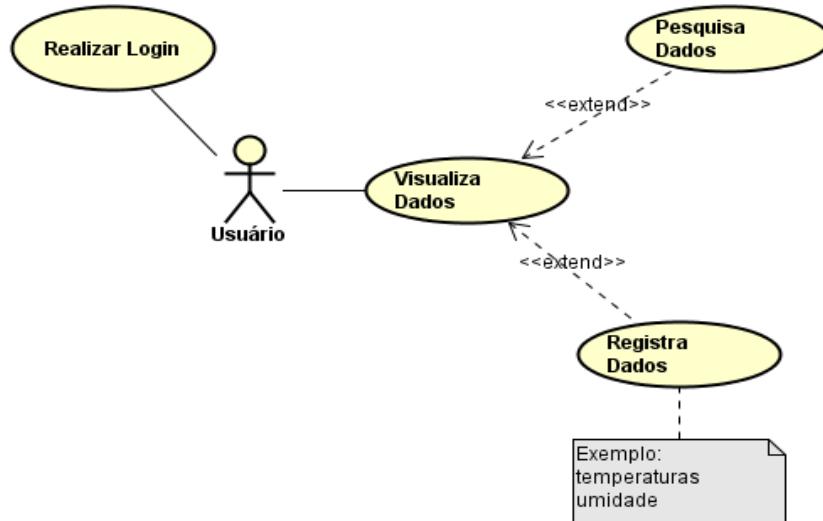


Figura 17 – Diagrama do Usuário

Fonte: Elaborado pelo autor

No diagrama Administrador exibido na figura 18, o administrador tem a capacidade de controlar os dados enviados dos ESP32, inserir, editar e excluir usuários.

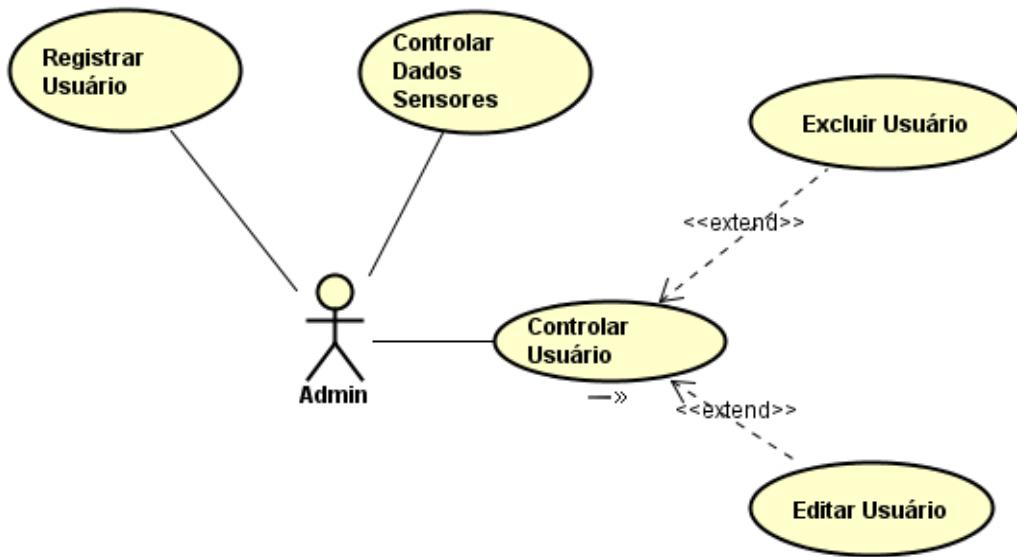


Figura 18 – Diagrama do Administrador

Fonte: Elaborado pelo autor

### 3.2.5 Diagrama Entidade-Relacionamento (DER)

O DER é um fluxograma que mostra como entidades, como pessoas, objetos ou conceitos, se relacionam dentro de um sistema. Utilizados para projetar ou depurar bancos de dados relacionais, os diagramas ER são comuns em engenharia de software, sistemas de informações empresariais, educação e pesquisa. Eles usam símbolos como retângulos, diamantes, ovais e linhas para representar entidades, relacionamentos e atributos. Entidades são substantivos e relacionamentos são verbos. Relacionam-se aos diagramas de estrutura de dados (DEDs), que focam nas relações de elementos dentro de entidades, e são frequentemente usados com diagramas de fluxo de dados (DFDs), que mapeiam o fluxo de informações para processos ou sistemas. Os diagramas ER oferecem uma visão abrangente da estrutura de um banco de dados, facilitando a visualização das interconexões dos dados (Lucidchart, 2024).

Como mostrado na Figura 19, o diagrama de entidade-relacionamento pode ser interpretado da seguinte forma: Um User pode possuir nenhuma ou várias Anotação Dados, enquanto uma Anotação pode pertencer a um ou vários User. Cada Anotação Dados está associada a um e somente um Dado do ESP32, e um ESP32 pode estar ligado a zero ou uma Anotação Dados. Além disso, um User pode ter zero ou vários Registro Login, sendo que cada Registro Login está vinculado a um e somente um User, como evidenciado na figura 19.

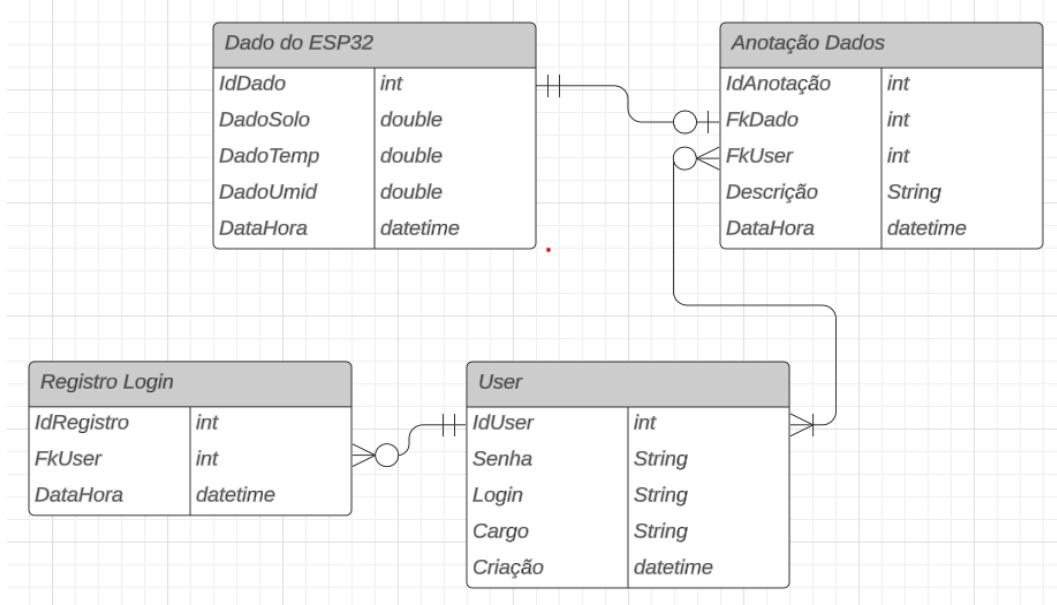


Figura 19 – Diagrama Entidade-Relacionamento

Fonte: Elaborado pelo autor

### 3.2.6 Diagrama de classe

Um diagrama de classes exibe um conjunto de classes, interfaces e colaborações, bem como seus relacionamentos. Esses diagramas são encontrados com maior frequência em sistemas de modelagem orientados a objeto e abrangem uma visão estática da estrutura do sistema. Os diagramas de classes que incluem classes ativas direcionam a perspectiva do processo estático do sistema. Os diagramas de componentes são variantes dos diagramas de classes. Os diagramas de classes que incluem classes ativas são empregados para direcionar a visão estática do processo de um sistema (BOOCH, 2006).

Como este projeto foi desenvolvido utilizando o padrão MVC, o Diagrama de Classe é dividido em pacotes que permitem organizar as funcionalidades do sistema em Model, Controller e View. Na Figura 20 é apresentado o diagrama de classe correspondente ao pacote Model, na Figura 21 é demonstrado o pacote View e na Figura 22 é exibido o pacote Controller.

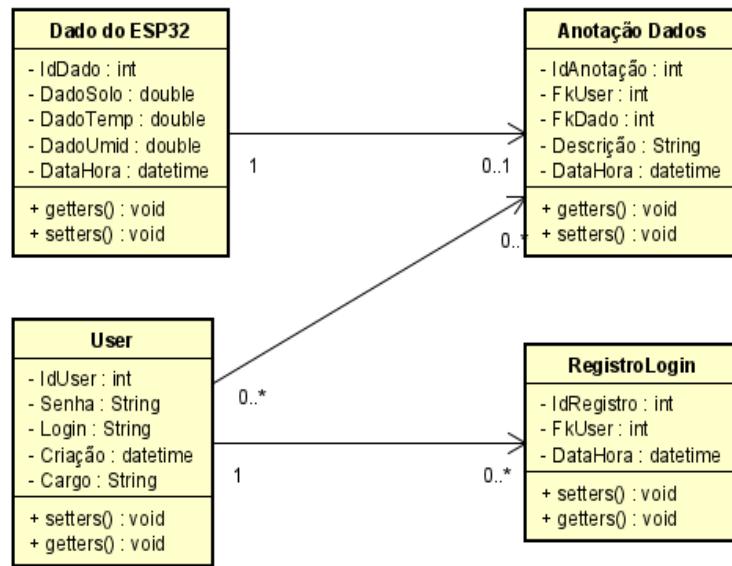


Figura 20 – Diagrama de Classe

Fonte: Elaborado pelo autor

O diagrama Model que faz parte da arquitetura MVC representa as classes e objetos, mostrando como os dados são manipulados e processados.

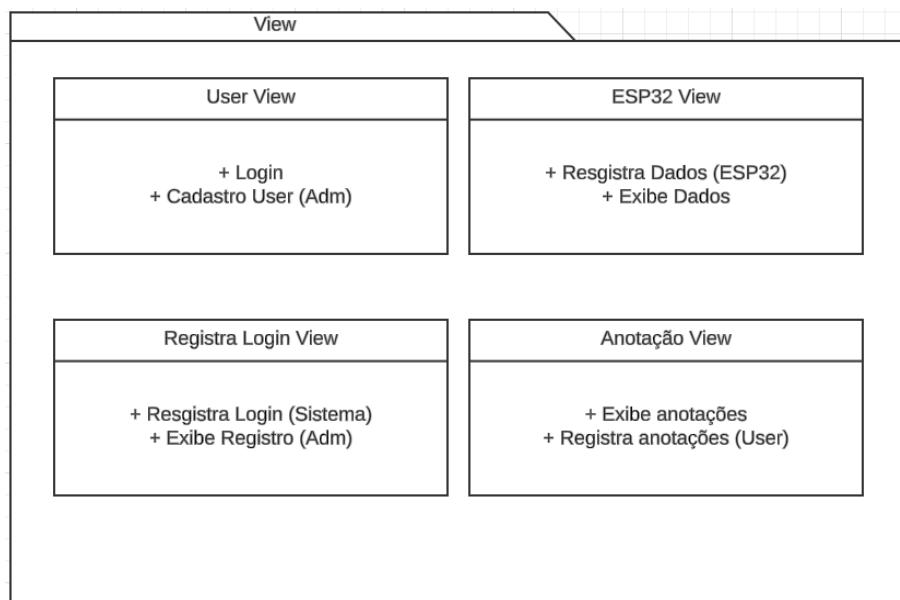


Figura 21 – Diagrama View

Fonte: Elaborado pelo autor

O diagrama View exibe a interface, apresenta os dados e cuida da interação ao usuário final.

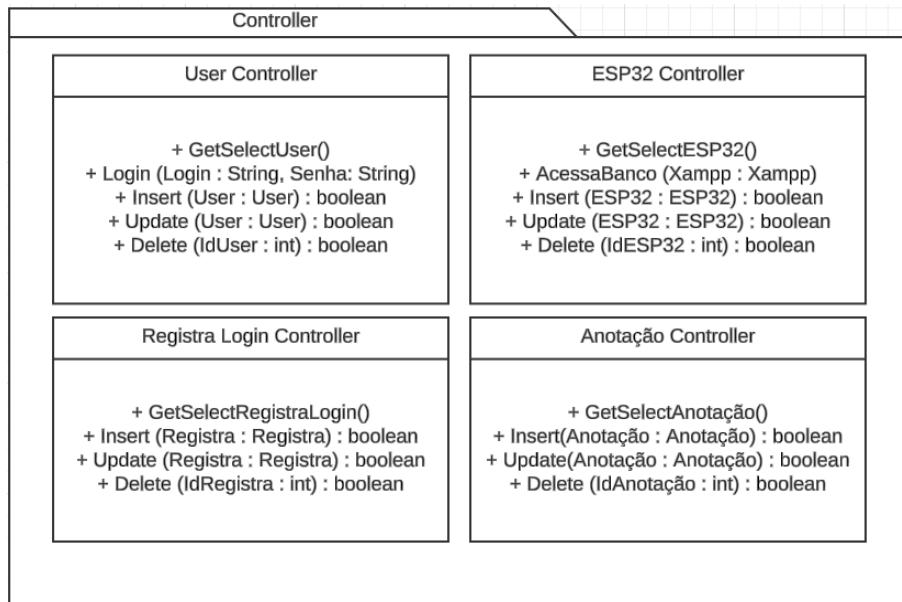


Figura 22 – Diagrama Controller

Fonte: Elaborado pelo autor

O diagrama Controller recebe as interações do usuário, processa solicitações e coordena a lógica de negócios. Atua como intermediário entre o Model e View.

### 3.2.7 Requisitos funcionais e não funcionais

A fase de levantamento de requisitos deve identificar dois tipos de requisitos: os funcionais e os não-funcionais. Os requisitos funcionais correspondem ao que o cliente quer que o sistema realize, ou seja, as funcionalidades do software. Já os requisitos não-funcionais correspondem às restrições, condições, consistências, validações que devem ser levadas a efeito sobre os requisitos funcionais. Por exemplo, em um sistema bancário deve ser oferecida a opção de abrir novas contas correntes, o que é um requisito funcional. Já determinar que somente pessoas maiores de idade possam abrir contas corrente é um requisito não-funcional (GUEDES, 2009).

No Anexo A encontra-se o Documento de Especificação de Requisitos do presente sistema. No documento são detalhados todos os requisitos e regras de negócios implementadas no sistema.

### 3.2.8 Protótipo

Na prototipagem, após o levantamento de requisitos, um protótipo do sistema é construído para ser usado na validação, quando o protótipo é revisto por um ou mais usuários que fa-

zem críticas a respeito de uma ou outra característica. O protótipo é então corrigido ou refinado com base nessas observações. Esse processo de revisão e refinamento continua até o protótipo ser aceito pelos usuários. Portanto, a técnica de prototipagem tem o objetivo de assegurar que os requisitos do sistema foram realmente bem entendidos. O resultado da validação pelo protótipo pode ser usado para refinar os modelos do sistema. Após a aceitação, o protótipo (ou parte dele) pode ser descartado ou utilizado como uma versão inicial do sistema (BEZERRA, 2006).

No Anexo B, encontram-se algumas telas do presente sistema, bem como a montagem do hardware.

## 4 RESULTADOS

Descrever o sistema implementado, apresentando as telas, como se dá a interação do usuário. Interessante partir da tela de login (quando houver), explicando tópicos como as restrições para senha e tal.

Neste capítulo, apresentamos os resultados alcançados durante o processo de desenvolvimento e implementação do sistema proposto, detalhando as etapas necessárias para alcançar seu objetivo final.

### 4.1 CONECTANDO OS MICROCONTROLADORES

Para o desenvolvimento do projeto, optou-se pelo uso de dois microcontroladores Heltec ESP32 LoRa WiFi V2 (discutido na seção 3.1.1), em conjunto com sensor de temperaturas e umidade do ar mais o módulo sensor de umidade do solo (descrita na seção 3.1.2 e 3.1.3).

A estrutura do hardware final do projeto ficou montado como mostrado na figura a seguir.

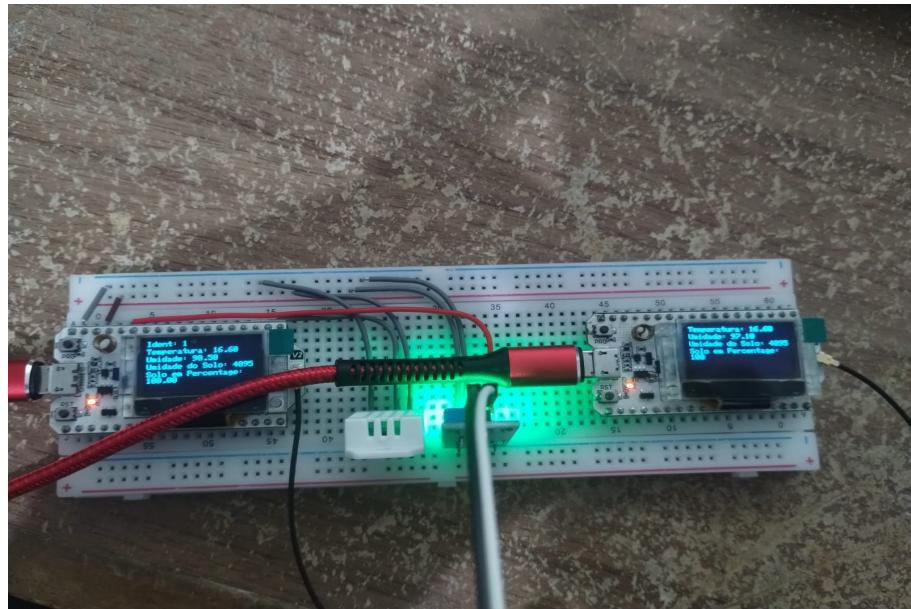


Figura 23 – Hardware  
Fonte: Elaborado pelo autor

A parte de codificação dos microcontroladores foi utilizado a IDE arduino, a linguagem C++, para a comunicação entre os ESP foi utilizado o LoRa, a comunicação ao ThigsBoard através do MQTT e ao site web usando o HTTP (discutidos nas seção 3.1.4 á seção 3.1.9).

Na figura a seguir, é mostrada a codificação para comunicação entre os ESP32 para o envio e o recebimento dos dados através do LoRa. No código, mostra a transmissão e o recebimento dos dados é feito através de uma função, mais a utilização de uma biblioteca que facilita a manipulação do LoRa.

```
void sendLoRaData(float Temperatura, float Umidade, int soilMoisture, int soilMoisturePercentage, int ident) {
    LoRa.beginPacket();
    LoRa.write(reinterpret_cast<uint8_t*>(&Temperatura), sizeof(Temperatura));
    LoRa.write(reinterpret_cast<uint8_t*>(&Umidade), sizeof(Umidade));
    LoRa.write(reinterpret_cast<uint8_t*>(&soilMoisture), sizeof(soilMoisture));
    LoRa.write(reinterpret_cast<uint8_t*>(&soilMoisturePercentage), sizeof(soilMoisturePercentage));
    LoRa.write(reinterpret_cast<uint8_t*>(&ident), sizeof(ident));
    LoRa.endPacket();
}
```

Figura 24 – Código da função para transmitir os dados pelo LoRa.

Fonte: Elaborado pelo autor

```
int packetSize = LoRa.parsePacket();
if (packetSize) {
    float Temperatura;
    float Umidade;
    int soilMoisture;
    int soilMoisturePercentage;
    int ident;

    LoRa.readBytes(reinterpret_cast<uint8_t*>(&Temperatura), sizeof(Temperatura));
    LoRa.readBytes(reinterpret_cast<uint8_t*>(&Umidade), sizeof(Umidade));
    LoRa.readBytes(reinterpret_cast<uint8_t*>(&soilMoisture), sizeof(soilMoisture));
    LoRa.readBytes(reinterpret_cast<uint8_t*>(&soilMoisturePercentage), sizeof(soilMoisturePercentage));
    LoRa.readBytes(reinterpret_cast<uint8_t*>(&ident), sizeof(ident));
```

Figura 25 – Código da função para receber os dados pelo LoRa.

Fonte: Elaborado pelo autor

Na figura a seguir, é mostrada a codificação para comunicação entre o ESP32 receptor com a plataforma ThingsBoard, para iniciar a biblioteca, enviar os dados e exibir através do MQTT.

```

/*ThingsBoard*/
#define TB_SERVER "thingsboard.cloud"
#define TOKEN "quRFAjtZ7GA0TVwLYd3J"

constexpr uint16_t MAX_MESSAGE_SIZE = 128U;

WiFiClient espClient;
Arduino_MQTT_Client mqttClient(espClient);
ThingsBoard tb(mqttClient, MAX_MESSAGE_SIZE);

void connectToThingsBoard() {
    if (!tb.connected()) {
        Serial.println("Connecting to ThingsBoard server");

        if (!tb.connect(TB_SERVER, TOKEN)) {
            Serial.println("Failed to connect to ThingsBoard");
        } else {
            Serial.println("Connected to ThingsBoard");
        }
    }
}

void sendDataToThingsBoard(float temp, float hum, float solo) {
    String jsonData = "{\"temperature\"::" + String(temp) +
    ", \"humidity\"::" + String(hum) + ", \"solo\":"+ String(solo) +"}";
    tb.sendTelemetryJson(jsonData.c_str());
    Serial.println("Data sent");
}

/*ThingsBoard*/

```

Figura 26 – Código da comunicação com o ThingsBoard.

Fonte: Elaborado pelo autor

Na figura a seguir, é mostrada a codificação para comunicação entre o ESP32 receptor com a site web para o envio dos dados e exibição através do HTTP.

```

if (WiFi.status() == WL_CONNECTED) {

    const char* serverName = "http://192.168.1.8/Irrigacao_Autonoma/insertDadosSensor";
    // Endereço do script PHP no XAMPP
    HTTPClient http;

    String httpRequestData = "temperature=" + String(Temperatura)
        + "&humidity=" + String(Umidade)
        + "&soil_moisture_percentage=" + String(soilMoisturePercentage)
        + "&soil_moisture=" + String(soilMoisture)
        + "&ident=" + String(ident);

    // Especificar URL para enviar o dado
    http.begin(serverName);

    // Enviar solicitação HTTP POST
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    int httpResponseCode = http.POST(httpRequestData);

    // Verificar a resposta do servidor
    if (httpResponseCode > 0) {
        String response = http.getString();
        Serial.println(httpResponseCode); // Código de resposta HTTP
        Serial.println(response); // Resposta do servidor
    } else {
        Serial.print("Erro ao enviar POST: ");
        Serial.println(httpResponseCode);
    }

    // Finalizar a conexão
    http.end();
}

```

Figura 27 – Código da comunicação com o ThingsBoard.

Fonte: Elaborado pelo autor

## 4.2 UTILIZANDO O THINGSBOARD

Para a exibição dos dados capturados pelos microcontroladores a primeira opção feita foi pela plataforma ThingsBoard (descrita na seção 3.1.8), comunicação entre eles se dá pelo MQTT e uma biblioteca fornecida pela propria plataforma. Só basta configurar os gráficos que queria usar e um token para adicionar na codificação nos ESP32 para saber qual é o endereço IP que tem que enviar os dados, ao logar numa conta que tenha acesso aos dados que estará visualizando em um gráfico.

Na figura 28 e 29 mostra a criação de um dispositivo no ThingsBoard para o recebimento dos dados.

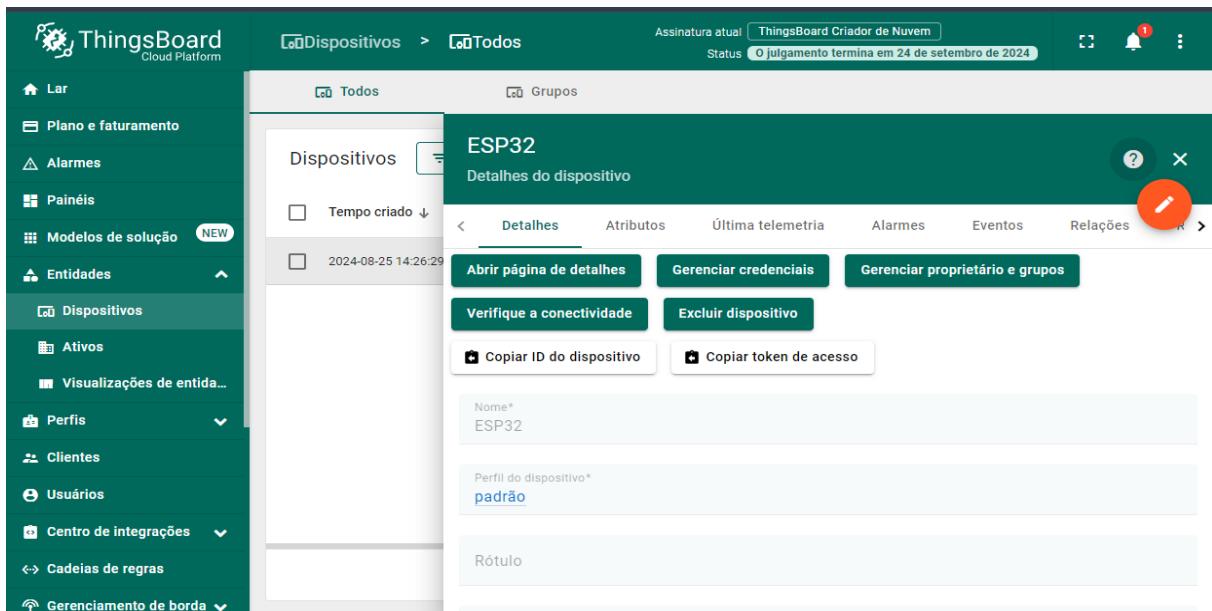


Figura 28 – Criando um dispositivo.

Fonte: Elaborado pelo autor

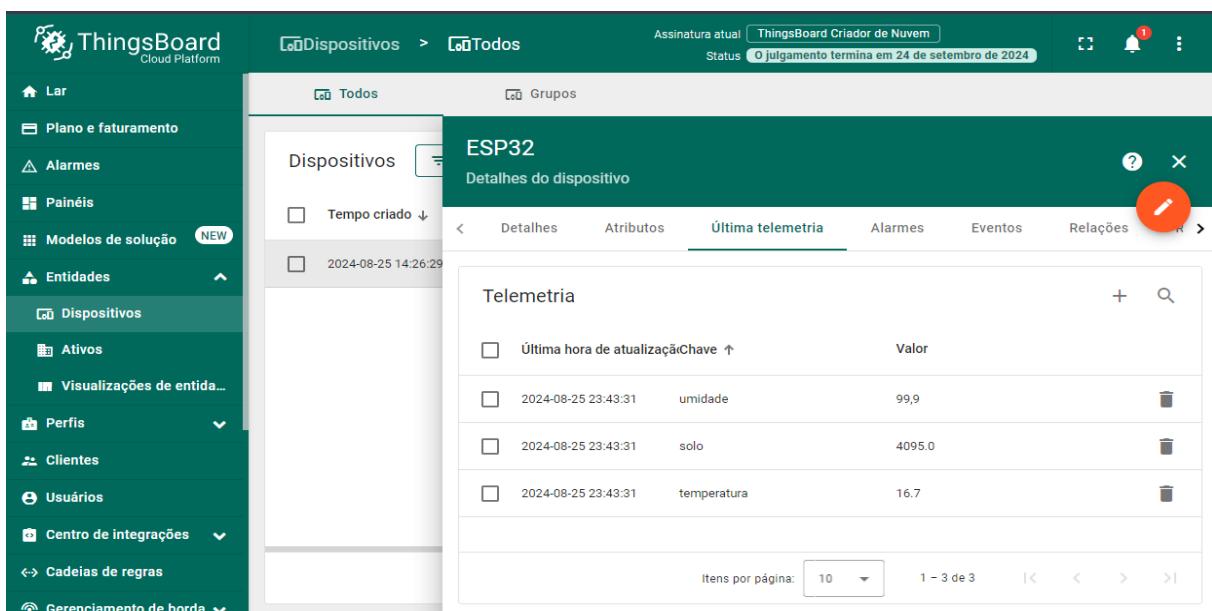


Figura 29 – Ultimos dados recebidos.

Fonte: Elaborado pelo autor

#### 4.3 DESENVOLVIMENTO DO SISTEMA WEB

## **5 CONCLUSÃO**

Está é a conclusão do trabalho ....



## REFERÊNCIAS

- ALVES, A. E. S. **Estudos do desempenho rádio LoraWan em cenários de mobilidade e trânsito de dispositivos.** 2023. Tese (Doutorado em Ciência da Computação) — .
- ALVES, J. **Sensores de Umidade de Solo:** tipos e diferenças. Acesso em: 23 de junho de 2024, Site.
- AMORIM JR, J.; RIBEIRO, C.; COLISTETE JR, R. Medição remota de parâmetros elétricos usando IoT baseado no microcontrolador ESP32. **X Encontro Científico de Física Aplicada,** [S.I.], p.109–116, 2019.
- ARDUINO, A. K.; KITTAAKOS. **ArduinoIDE.** Acesso em: 11 de junho de 2024, Site.
- ARDUINO MASSIMO BANZI, D. C. T. I.; MELLIS, D. A. **Arduino.** Acesso em: 11 de junho de 2024, Site.
- AUTHORS, . T. T. **Documentation - What is ThingsBoard?** Acesso em: 23 de junho de 2024, Site.
- AUTOMATION, C. . H. **WiFi LoRa 32.** Acesso em: 23 de junho de 2024, Site.
- BENTES, A. L. P. et al. Sumaúma: estação de monitoramento de plantas domésticas utilizando android e esp32. , [S.I.], 2023.
- BEZERRA, E. Princípios de análise e projeto de sistemas com UML. 3<sup>a</sup> Tiragem. **Editora Campus,** [S.I.], 2006.
- BOOCH, G. **UML:** guia do usuário. [S.I.]: Elsevier Brasil, 2006.
- CALZAVARA, B. B. Historia e importancia da irrigação. , [S.I.], 1953.
- CONVERSE, T.; PARK, J. **PHP:** a bíblia. [S.I.]: Gulf Professional Publishing, 2003.
- CULLEN, P. The journey to sustainable irrigation. In: IRRIGATION ASSOCIATION OF AUSTRALIA ANNUAL CONFERENCE, Adelaide. **Anais...** [S.I.: s.n.], 2004.
- FLANAGAN, D. **JavaScript:** o guia definitivo. [S.I.]: Bookman Editora, 2012.
- GUEDES, G. T. **UML 2. Uma Abordagem Prática”, São Paulo, Novatec,** [S.I.], p.32, 2009.

- GUEDES, G. T. **UML 2-Uma abordagem prática.** [S.l.]: Novatec Editora, 2018.
- HENRIQUES, L. F. A. et al. Implementação e monitoramento de um sistema de irrigação automatizado em IoT utilizando módulo ESP32 em plantio caseiro. , [S.l.], 2021.
- IFFAR. **Instituto Federal Farroupilha - Campus S.V.S, de São Vicente do Sul (RS).** Acesso em: 16 de março de 2024, Descrição institucional.
- IFFAR. **Agricultura e Estufas no IFFar - Campus S.V.S.** Acesso em: 16 de março de 2024, Descrição institucional.
- JOBSTRAIBIZER, F. **Criação de sites com o CSS.** [S.l.]: Universo dos Livros Editora, 2009.
- KERSCHBAUMER, R. et al. Microcontroladores. **Santa Catarina, Brasil,** [S.l.], 2013.
- LUCIANO, J.; ALVES, W. J. B. Padrão de arquitetura MVC: model-view-controller. **EPEQ Fafibe,** [S.l.], v.1, n.3a, p.102–107, 2017.
- Lucidchart. **O Que é Diagrama Entidade-Relacionamento.** Acesso em: 26 jun. 2024.
- MADEIRA, N. R.; COSTA RIBEIRO, C. S. da; AMARO, G. B. **Sistema convencional.** Acesso em: 23 de março de 2024, Site.
- MAKERHERO®. **Sensor de Umidade e Temperatura DHT11.** Acesso em: 23 de junho de 2024, Site.
- MAKERHERO®. **Sensor de Umidade e Temperatura DHT11.** Acesso em: 23 de junho de 2024, Site.
- MARQUELLI, W.; SILVA, W. Seleção de sistemas de irrigação para hortaliças. , [S.l.], 1998.
- MAXMAQ. **Irrigação por Gotejamento Subsuperficial (SDI).** Acesso em: 23 de março de 2024, Site.
- MILANI, A. **MySQL-guia do programador.** [S.l.]: Novatec Editora, 2007.
- ORACLE. **O que é a Internet das Coisas (IoT)?** Acessado em: 8 de julho de 2024.
- OSTER, J. D.; WICHELNS, D. Economic and agronomic strategies to achieve sustainable irrigation. **Irrigation Science**, New York, v.22, p.107–120, 2003.

PIRES, L. **Métodos de Irrigação:** sistemas, características e aplicações. Acesso em: 23 de março de 2024, Site.

RICARTE, I. L. M. Programação orientada a objetos com c++. **Notas de apoio a cursos-FEEC/UNICAMP**, [S.I.], 1995.

RIVULIS. **Conheça a Irrigação Por Gotejamento e Dobre Sua Produtividade.** Acesso em: 23 de março de 2024, Site.

SALINDO, A. B. P. et al. A plataforma Arduino como estratégia tecnológica na criação de um sistema de irrigação de baixo custo para hortaliças. , [S.I.], 2023.

SANTANA, B. **O Que é HTTP e Como Ele Permite o Acesso ao Seu Site.** Acesso em: 23 de junho de 2024, Site.

SANTOS, B. P. et al. Internet das coisas: da teoria à prática. **Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, [S.I.], v.31, p.16, 2016.

SEIDLER, K. O. **About.** Acesso em: 23 de junho de 2024, Site.

SILVA, F. B. da. **Sistema de cultivo sulco-camalhão e a diversificação de culturas no RS.** (Acessado em Abril de 2024), Disponível em:

<https://www.agrolink.com.br/artigos/>

[sistema-de-cultivo-sulco-camalhao-e-a-diversificacao-de-culturas-no-rs\\_482726.html.](https://www.agrolink.com.br/artigos/sistema-de-cultivo-sulco-camalhao-e-a-diversificacao-de-culturas-no-rs_482726.html)

SILVA, M. S. **Criando sites com HTML:** sites de alta qualidade com html e css. [S.I.]: Novatec Editora, 2008.

SOLDÁ, E. N. et al. **SISTEMA AUTÔNOMO DE IRRIGAÇÃO BASEADO NA UMIDADE DO SOLO.** UNIFUNEC CIENTÍFICA MULTIDISCIPLINAR, [S.I.], v.12, n.14, p.1–14, 2023.

TARGA, M. S.; SILVA, M. C.; CEZAR, V. R. S. Uso de microcontrolador Arduino para a determinação da permeabilidade do solo. **Revista Técnica Ciências Ambientais**, [S.I.], v.1, n.1, p.1–14, 2019.

TECH first. **Internet das coisas no setor de serviços.** Acesso em: 25 de março de 2024, Site.

TORRES, A. B.; ROCHA, A. R.; SOUZA, J. N. de. Análise de desempenho de brokers mqtt em sistema de baixo custo. In: XV WORKSHOP EM DESEMPENHO DE SISTEMAS COMPUTACIONAIS E DE COMUNICAÇÃO. *Anais...* [S.l.: s.n.], 2016. p.2804–2815.

TORRES, Y. P. et al. Sistema de sensoriamento em IoT aplicado a pequenas plantações utilizando ESP32 e LoRa., [S.l.], 2021.

## APÊNDICES

---

## .1 TESTE

## ANEXOS

---

**ANEXO A – Requisitos funcionais e não funcionais**

# **Documento de Especificação de Requisitos**

**< Sistema de irrigação autônoma >**

**< 24 de Junho de 2024 >**

**< Versão 1.0 >**

**< Thiago Ryan da Silva Aires>**

### Histórico de Revisões

Data	Versão	Descrição	Autor
24/06/2024	1.0	<Criação do documento de especificação das funcionalidades>	<Thiago Ryan>

## 1. Introdução

Este documento especifica os requisitos do sistema de **Irrigação Autônoma**, fornecendo aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

### 1.1 Visão geral do documento

Além desta seção introdutória, as seções seguintes estão organizadas como descrito abaixo.

1. **Seção 2 – Descrição geral do sistema:** apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.
2. **Seção 3 – Requisitos funcionais:** especificam todos os casos de uso do sistema, descrevendo os fluxos de eventos, prioridades, atores, entradas e saídas de cada caso de uso a ser implementado.
3. **Seção 4 – Requisitos não funcionais:** especifica todos os requisitos não funcionais do sistema, divididos em requisitos de usabilidade, confiabilidade, desempenho, segurança, distribuição, adequação a padrões e requisitos de hardware e software.
4. **Seção 5 – Regras de Negócios:** As regras de negócio estabelecem requisitos gerais para o sistema, provenientes do próprio negócio como normas, políticas, legislações etc.

### 1.2 Prioridades dos requisitos

Para estabelecer a prioridade dos requisitos, nas seções 3 e 4, foram adotadas as denominações “essencial”, “importante” e “desejável”.

1. **Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
2. **Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
3. **Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

## 2. Descrição geral do sistema

### 2.1 Abrangência e sistemas relacionados

*O sistema de Irrigação Autônoma é voltado para facilitar o monitoramento de hortas, jardins e estufas, onde o sistema toma a melhor decisão de quando irrigar as plantas e repassa os dados capturados para o banco de dados e depois exibe num dashboard.*

## 3. Requisitos funcionais

Nº Requisito	Descrição	Prioridade
RF [001]	O sistema/software deverá ser capaz de ler a umidade do solo.	Essencial
RF [002]	O sistema/software deverá ser capaz de ler a temperatura e a umidade do ar.	Essencial
RF [003]	O sistema/software deverá ser capaz de controlar a água para irrigação.	Essencial
RF [004]	O sistema/software poderá possuir um dashboard para o usuário acompanhar os dados do sistema.	Essencial
RF [005]	O sistema/software poderá possuir um gerador de relatórios em PDF dos dados.	Essencial

## 4. Requisitos não funcionais

### 4.1 Requisitos não funcionais de Desempenho

Nº Requisito	Descrição	Prioridade
RNF [001]	O sistema deverá utilizar microprocessador ESP32.	Essencial
RNF [002]	O sistema deverá utilizar o chip Long Range (LoRa).	Essencial
RNF [003]	O sistema deverá ter conexão com o WiFi.	Essencial
RNF [004]	O sistema deverá utilizar o protocolo MQTT	Essencial
RNF [005]	O sistema deverá utilizar o protocolo HTTP	Essencial
RNF [006]	O sistema deverá se utilizar site web, deverá ser feito em PHP utilizando HTML, CSS e JS.	Essencial
RNF [007]	O sistema deverá utilizar banco de dados MySQL	Essencial

## 4.2 Requisitos não funcionais de Segurança

Nº Requisito	Descrição	Prioridade
RNF [001]	O sistema deve garantir a proteção adequada dos dados, seguindo as regulamentações de privacidade e segurança.	Essencial
RNF [002]	O sistema deverá possuir o controle de acesso mediante ao perfil (Administrador ou Usuário)	Essencial

## 4.3 Requisitos não funcionais de Usabilidade

Nº Requisito	Descrição	Prioridade
RNF [001]	O sistema deverá ser intuitivo e fácil de aprender	Essencial
RNF [002]	O sistema deverá possuir cores agradáveis.	Essencial
RNF [003]	O sistema deve permitir que os usuários realizem operações com poucos cliques ou toques.	Essencial
RNF [004]	O sistema deve ser visualmente atraente e ter um design agradável, com elementos gráficos e layout bem projetados.	Essencial

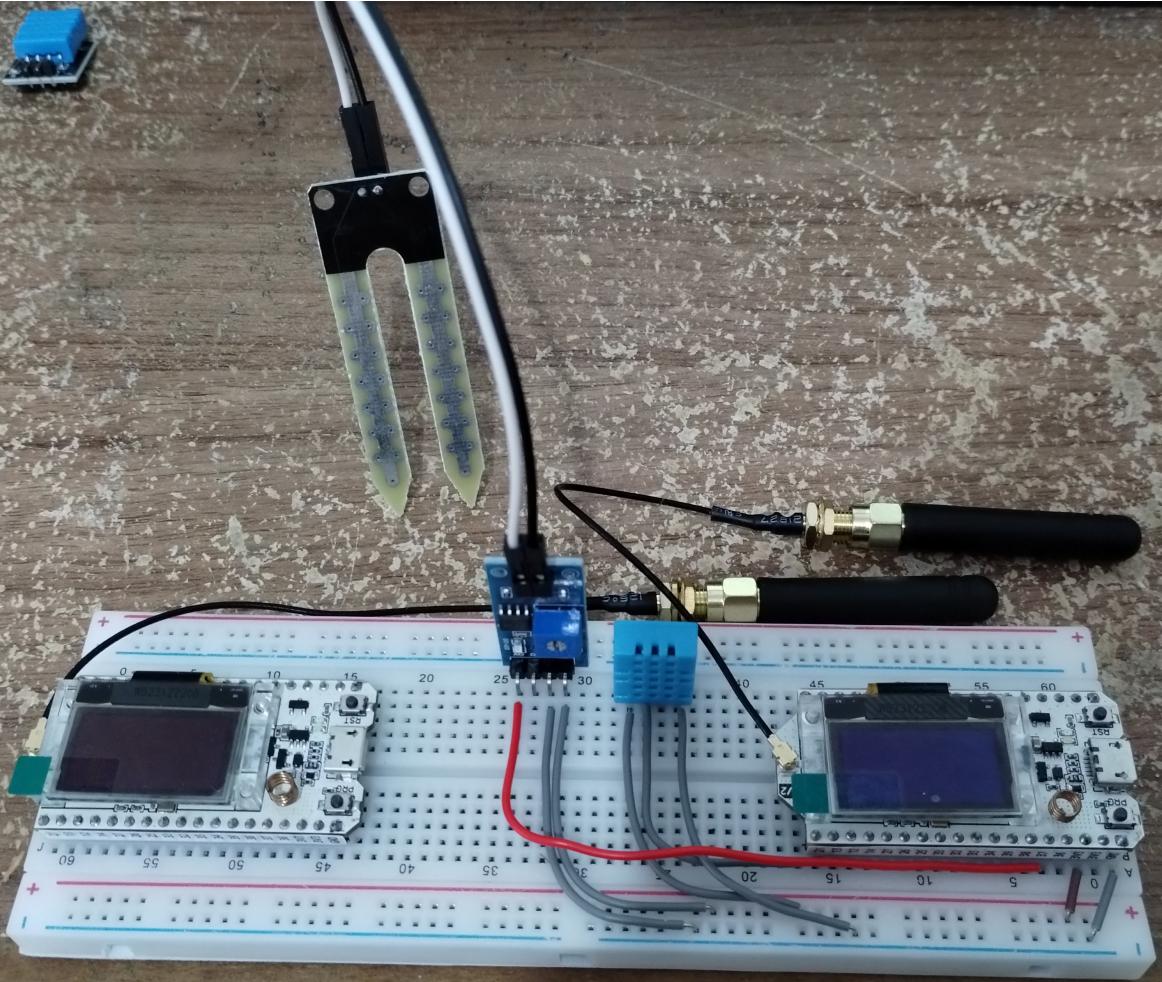
## 4.4 Requisitos não funcionais de Hardware

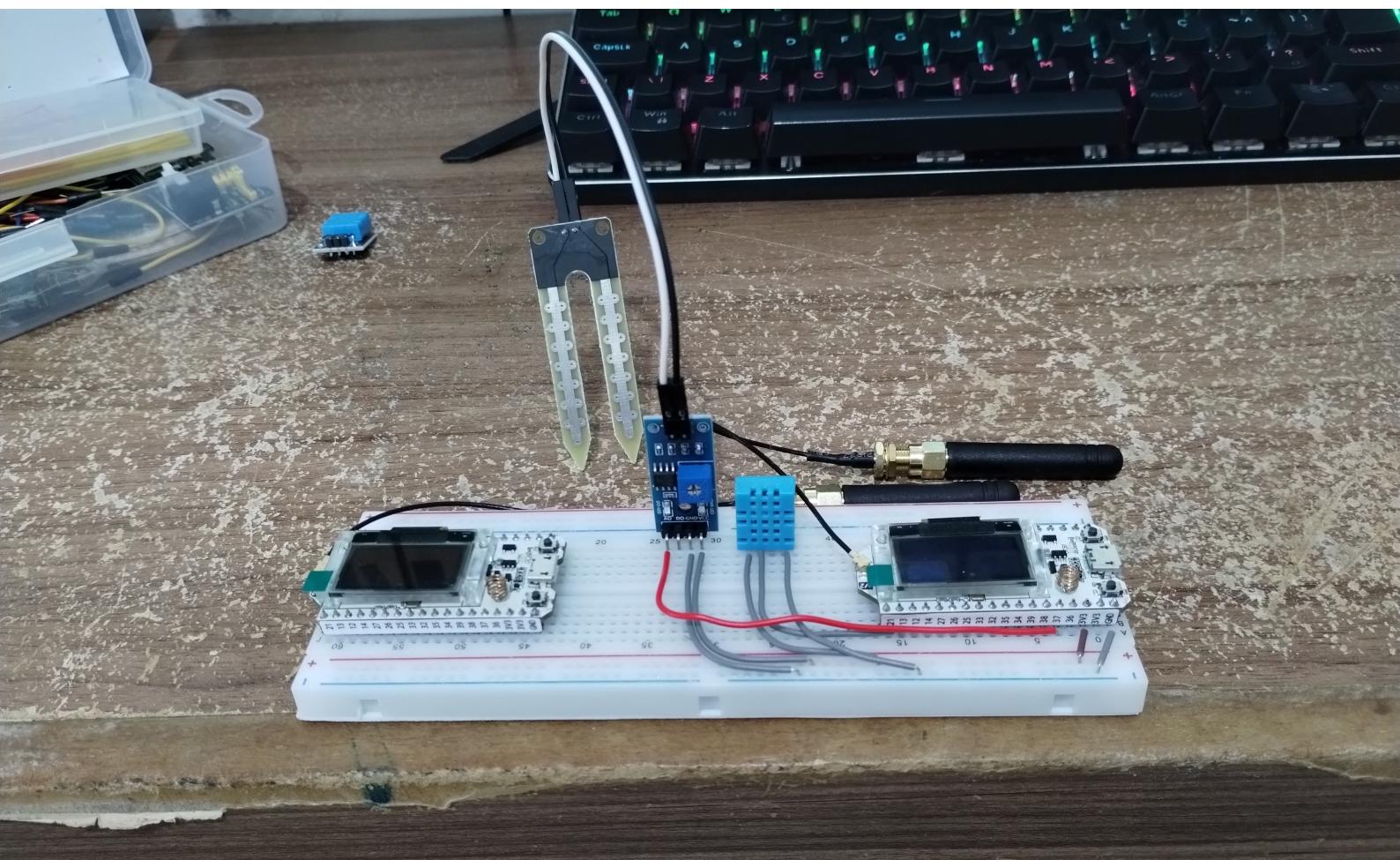
Nº Requisito	Descrição	Prioridade
RNF [001]	O sistema deverá ter a capacidade de armazenamento adequada para acomodar os dados necessários	Essencial
RNF [002]	O sistema deverá ter um servidor local (XAMPP)	Essencial
RNF [003]	O sistema deve ser eficiente em termos de consumo de energia, buscando minimizar o consumo desnecessário e otimizar a vida útil da bateria (caso aplicável).	Essencial
RNF [004]	O hardware do sistema deve ser confiável, minimizando a probabilidade de falhas.	Essencial

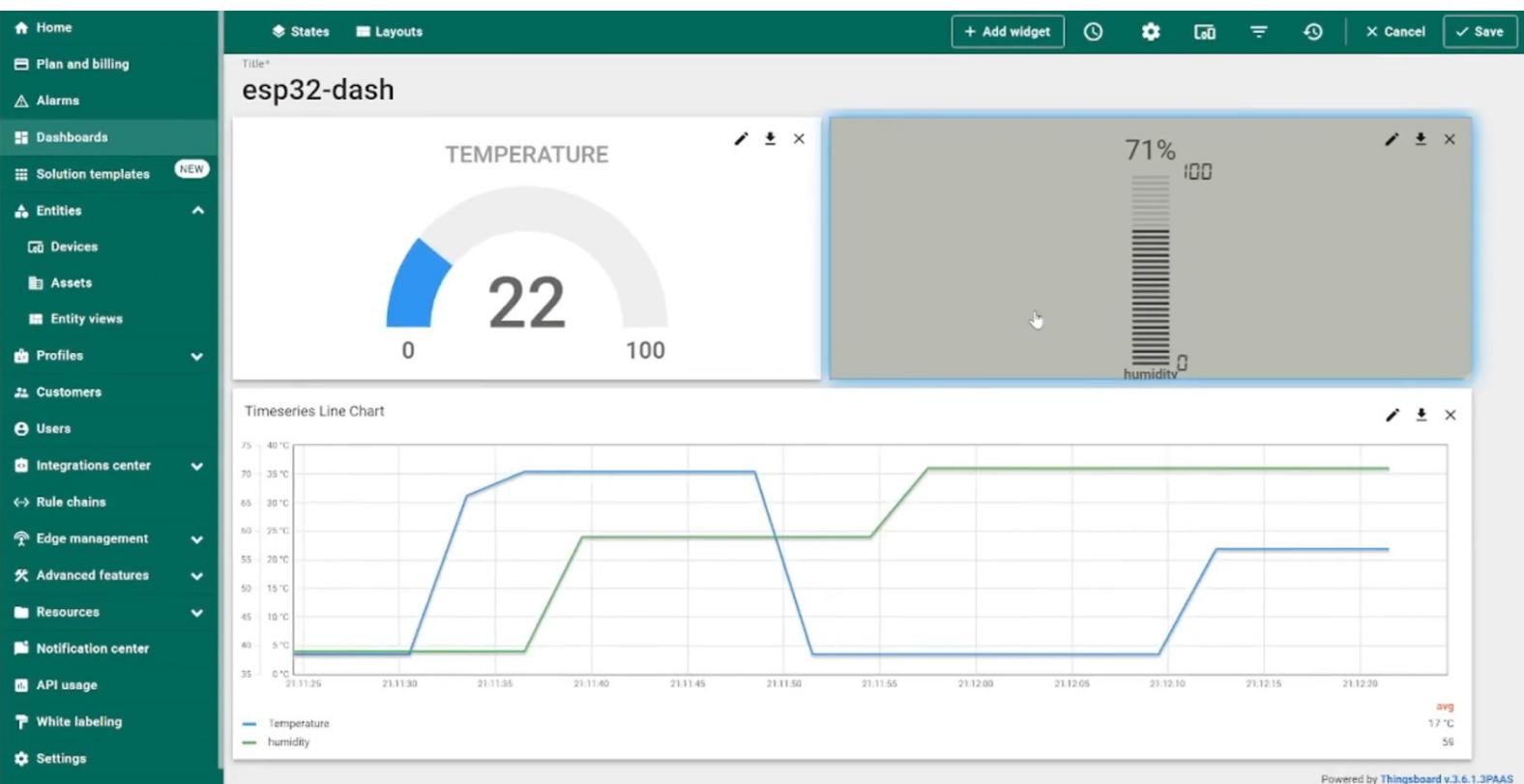
## 5. Regras de Negócios

Nº Regra Negócio	Descrição
RGN [001]	Apenas o administrador poderá cadastrar novos usuários.
RGN [002]	Apenas usuários logados poderão ver suas informações.
RGN [003]	O usuário poderá fazer reclamações/sugestões/dúvidas sobre o sistema.

**ANEXO B – Protótipagem**







## Cadastro de Usuário

Nome

Email

Senha

Confirme a Senha

Cadastrar

## Login de Usuário

Email

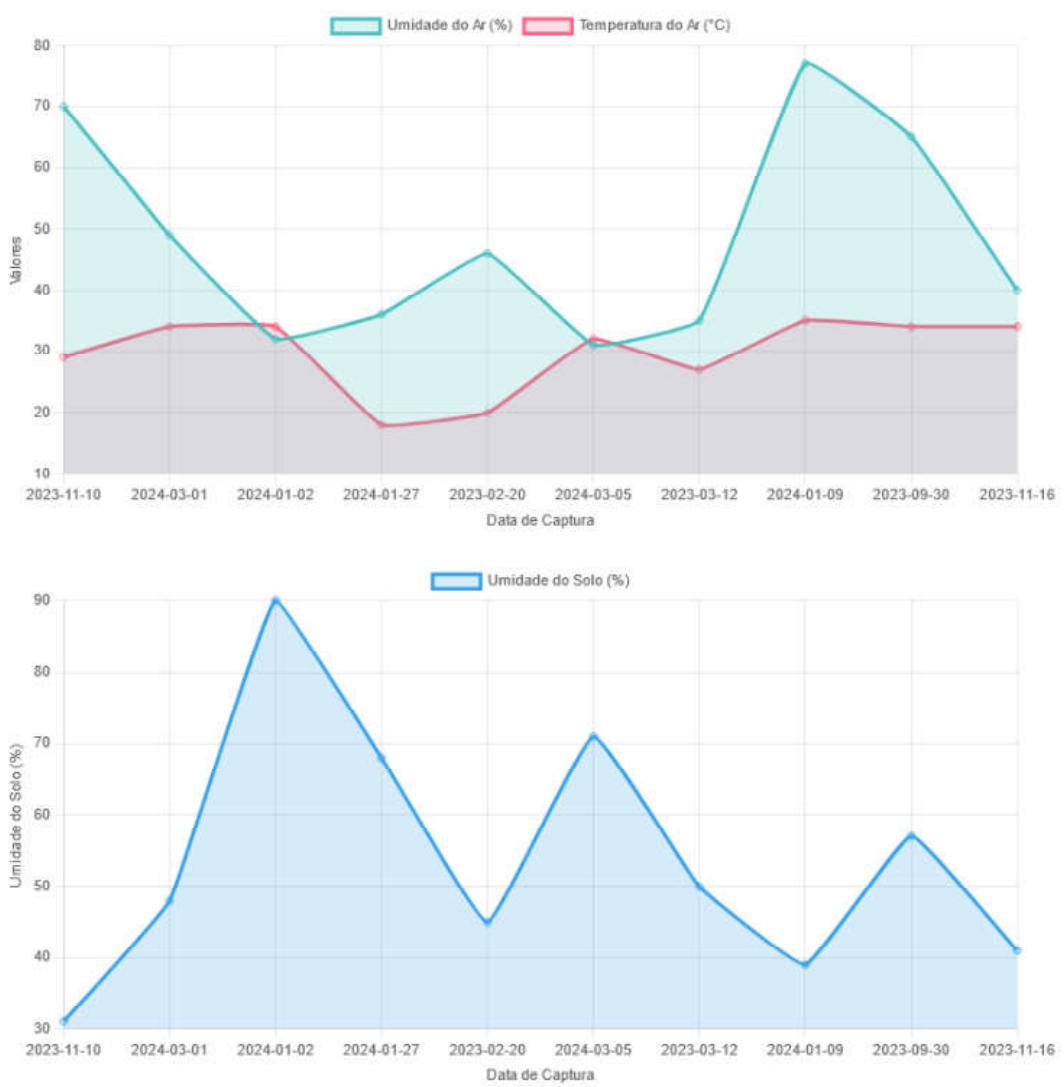
Senha

Entrar

## Dados de Sensores

Data de Captura	Umidade do Ar (%)	Temperatura do Ar (°C)	Umidade do Solo (%)
2023-08-07	30	25	61
2023-05-21	33	20	75
2024-03-13	30	14	40
2023-09-17	80	16	71
2023-12-13	55	24	42
2023-06-28	22	23	35
2023-05-29	65	31	61
2024-03-18	52	17	62
2023-06-23	72	33	38
2023-07-25	57	25	75

## Gráficos de Sensores



## Anotações sobre Dados de Sensores

Anotação

[Adicionar Anotação](#)