

**Universidade Paulista  
Instituto de Ciências Exatas e Tecnologia  
Graduação em Engenharia da Computação**

*Camila Maria do Prado Santos*

*Lucas Vinicius Siqueira da Rosa*

*Thiago Wender Gonsales*

*Yanka Beatriz Purificação dos Santos*

**SAMANTA: VASO INTELIGENTE**

**São José dos Campos - SP**

**2024**

**Universidade Paulista  
Instituto de Ciências Exatas e Tecnologia  
Graduação em Engenharia da Computação**

**Camila Maria do Prado Santos**

**Lucas Vinicius Siqueira da Rosa**

**Thiago Wender Gonsales**

**Yanka Beatriz Purificação dos Santos**

**SAMANTA: VASO INTELIGENTE**

Trabalho de curso apresentado ao Instituto de Ciências Exatas e Tecnologia da Universidade Paulista – UNIP, campus de São José dos Campos, como parte dos requisitos necessários para a obtenção do título de Bacharel em Engenharia da Computação, sob orientação do Prof.<sup>o</sup> Dr. Rodrigo Soares Moraes;

**São José dos Campos - SP**

**2024**

CIP - Catalogação na Publicação

Samanta: Vaso Inteligente / Camila Maria do Prado Santos...[et al.]. -  
2024.

84 f.

Trabalho de Conclusão de Curso (Graduação) apresentado ao Instituto  
de Ciência Exatas e Tecnologia da Universidade Paulista, São José dos  
Campos, 2024.

Área de Concentração: Internet das Coisas.

Orientador: Prof. Dr. Rodrigo Soares Moraes.

1. Vaso Inteligente. 2. Automação. 3. Sensores Ambientais. 4. IoT:  
Internet das Coisas. 5. Irrigação Automatizada. I. Santos, Camila Maria do  
Prado. II. Moraes, Rodrigo Soares (orientador).

**Camila Maria do Prado Santos**  
**Lucas Vinicius Siqueira da Rosa**  
**Thiago Wender Gonsales**  
**Yanka Beatriz Purificação dos Santos**

### **SAMANTA: VASO INTELIGENTE**

Trabalho de curso apresentado ao Instituto de Ciências Exatas e Tecnologia da Universidade Paulista – UNIP, campus de São José dos Campos, como parte dos requisitos necessários para a obtenção do título de Bacharel em Engenharia da Computação sob orientação do Prof.<sup>o</sup> Dr. Rodrigo Soares Moraes;

---

**Prof. Msc. Fernando Mauro - UNIP**

---

**Prof.<sup>o</sup> Dr. Rodrigo Soares Moraes - UNIP**

/ / 2024

**Data da aprovação**

Dedicamos este trabalho a nossa família.

## **AGRADECIMENTOS**

Gostaríamos de iniciar expressando nossa sincera gratidão ao nosso orientador, Dr. Rodrigo Soares Moraes, cuja orientação, paciência e expertise foram indispensáveis na condução deste Trabalho de Conclusão de Curso. Sua dedicação e apoio foram fundamentais para que pudéssemos alcançar este marco.

Agradecemos também aos nossos professores, que ao longo da nossa trajetória, com o seu vasto conhecimento e empenho, foram essenciais para nossa formação. Seus ensinamentos e orientações contribuíram significativamente para o nosso desenvolvimento pessoal e profissional.

Dedicamos esse trabalho aos nossos familiares e amigos, nosso profundo agradecimento pelo constante apoio e incentivo desde o início de nossa jornada acadêmica. Sua presença e encorajamento foram fundamentais para superar os desafios ao longo do curso.

Em especial, dedicamos este trabalho a Adilson Donizete da Rosa, pai de Lucas Vinícius Siqueira da Rosa. Ele foi um pai maravilhoso, sempre presente, e desejava estar aqui para ver o filho se formar. Seu apoio incondicional e sua dedicação constante à família foram fontes de força e inspiração. Embora sua presença física não esteja aqui para testemunhar esta conquista, seu amor e sacrifício são eternamente valorizados e lembrados.

Por fim, dedicamos este trabalho, com carinho e saudade, à memória de Reginaldo Furtado, amado tio e amigo, cuja ausência é profundamente sentida. Sua lembrança permanece como uma fonte constante de inspiração.

## **RESUMO**

Este trabalho apresenta o desenvolvimento de um protótipo de um vaso inteligente, denominado Samanta, voltado para o cuidado automatizado de plantas em ambientes domésticos. O sistema integra hardware e software para monitorar condições ambientais como umidade do solo, temperatura e luminosidade, utilizando sensores controlados por um Arduino Uno R3. O protótipo também inclui uma bomba peristáltica para irrigação automática, controlada por um módulo relé e gerida por um módulo ESP01 com conectividade Wi-Fi. A interface web desenvolvida com ReactJS, NodeJS e Express permite aos usuários monitorar remotamente os dados coletados e gerenciar a manutenção da planta. O projeto busca facilitar o cultivo doméstico, incentivando práticas de jardinagem sustentáveis e promovendo o uso eficiente de recursos através da tecnologia da Internet das Coisas (IoT). Além de descrever a construção e o funcionamento do protótipo, o trabalho discute os testes realizados para validar a precisão dos sensores, a eficiência do sistema de irrigação e a usabilidade da interface. Os resultados demonstram o potencial do sistema como uma solução prática para o cuidado de plantas, destacando sua aplicabilidade em diferentes contextos domésticos.

**Palavras-chave:** Vaso inteligente, Automação, Sensores ambientais, IoT (Internet das Coisas), Irrigação automatizada.

## **ABSTRACT**

This work presents the development of a smart pot prototype, named Samanta, aimed at automated plant care in domestic environments. The system integrates hardware and software to monitor environmental conditions such as soil moisture, temperature, and light, using sensors controlled by an Arduino Uno R3. The prototype also includes a peristaltic pump for automatic irrigation, managed by a relay module and an ESP01 module with Wi-Fi connectivity. The web interface, developed with ReactJS, NodeJS, and Express, allows users to remotely monitor the collected data and manage plant maintenance. The project aims to facilitate home gardening, encouraging sustainable gardening practices and promoting efficient resource use through Internet of Things (IoT) technology. In addition to describing the construction and operation of the prototype, the work discusses the tests carried out to validate the accuracy of the sensors, the efficiency of the irrigation system, and the usability of the interface. The results demonstrate the system's potential as a practical solution for plant care, highlighting its applicability in various domestic contexts.

Keywords: Smart pot, Automation, Environmental sensors, IoT (Internet of Things), Automated irrigation

## LISTA DE ILUSTRAÇÕES

Figura 1: Protótipo do Hardware.....	15
Figura 2: Aplicação WEB.....	16
Figura 3: The Smart Garden 3.....	19
Figura 4: Lua.....	19
Figura 5: Ivy.....	20
Figura 6: Arquitetura Geral do Projeto.....	32
Figura 7: Arquitetura do Hardware.....	33
Figura 8: Arduino.....	34
Figura 9: Sensores.....	35
Figura 10: Módulo Relé 5V.....	35
Figura 11: Bomba Peristáltica 12V.....	36
Figura 12: Display LCD 20X4 I2C.....	37
Figura 13: Esquema do protótipo do Vaso Samanta.....	38
Figura 14: Fluxograma do Funcionamento do Hardware.....	39
Figura 15: Modelo Cliente Servidor.....	45
Figura 16: Guia de Estilos.....	46
Figura 17: User flow.....	48
Figura 18: Gráfico de temperatura.....	48
Figura 19: Gráfico de umidade e luminosidade.....	48
Figura 20: Cadastro.jsx.....	50
Figura 21: samantaService.jsx.....	51
Figura 22: Login.jsx.....	52
Figura 23: auth.jsx.....	53
Figura 24: auth.jsx.....	54
Figura 25: samantaService.js.....	55
Figura 26: CadastroPlantas.jsx.....	56
Figura 27: Modelo Conceitual.....	61
Figura 28: Modelo Lógico.....	61
Figura 29: Modelo Físico.....	62
Figura 30: Índices.....	63
Figura 31: Tela de cadastro do usuário na aplicação.....	66
Figura 32: Tela de login do usuário na aplicação.....	67
Figura 33: Tela de home da aplicação.....	67
Figura 34: Tela de cadastro do vaso.....	68
Figura 35: Tela de home da aplicação com os vasos mapeados.....	69
Figura 36: Tela de especificações da planta.....	70
Figura 37: Tela de relatório da aplicação.....	70

## **LISTA DE ABREVIATURAS E SIGLAS**

AC - Alternating Current (Corrente Alternada)

ADC - Analog-to-Digital Converter (Conversor Analógico-Digital)

ANSI - American National Standards Institute

API - Application Programming Interface (Interface de Programação de Aplicações)

ARM - Advanced RISC Machine (Máquina Avançada de Conjunto de Instruções Reduzidas)

AT - Attention (Atenção)

AWS - Amazon Web Services (Serviços Web da Amazon)

CSS - Cascading Style Sheets

DC - Direct Current (Corrente Contínua)

DCL - Data Control Language (Linguagem de Controle de Dados)

DDL - Data Definition Language (Linguagem de Definição de Dados)

DML - Data Manipulation Language (Linguagem de Manipulação de Dados)

DOM - Document Object Model

et.al - *et alii* (E Outros)

HMAC - Código de Autenticação de Mensagem

HTTP - Hypertext Transfer Protocol

I/O - In/Out (Entrada/Saída)

ICSP - In-Circuit Serial Programming (Programação Serial em Circuito)

IoT - Internet of Things (Internet das Coisas)

JS - JavaScript

JSON - JavaScript Object Notation

JWT – Json Web token

LCD - Liquid Crystal Display (Display de Cristal Líquido)

MHz - Megahertz

n.d. - *non datatum* (Não Datado)

p. - Página

PWM - Pulse Width Modulation (Programação Serial em Circuito)

REST - Representational State Transfer (Transferência de Estado Representacional)

RFC - Request for Comments

RSA - Rivest-Shamir-Adleman

SCL - Serial Clock (Relógio Serial)

s.d. - Sem data

SDA - Serial Data (Dado Serial)

SGBD - Sistema de Gerenciamento de Banco de Dados

SHA - Secure Hash Algorithm (Algoritmo de Hash Seguro)

SQL - Structured Query Language (Linguagem de Consulta Estruturada)

TC - Controle de Transações

URL - Uniform Resource Locator (Localizador Uniforme de Recursos)

USB - Universal Serial Bus (Barramento Serial Universal)

Wi-Fi - Wireless Fidelity

WPA - Wi-Fi Protected Access (Acesso Protegido por Wi-Fi)

## LISTA DE SÍMBOLOS

- $x_i$  - Amostra individual
- $\approx$  - Aproximadamente
- $\sigma$  - Desvio-padrão
- $\div$  - Divisão
- $^{\wedge}$  - Exponenciado
- $K$  - Fator de conversão de tensão para temperatura
- $^{\circ}\text{C}$  - Graus Celsius
- $=$  - Igual
- $i$  - Índice
- $L$  - Leitura do sensor de luminosidade
- $+$  - Mais
- $\pm$  - Mais ou menos
- $\bar{x}$  - Média amostral
- $-$  - Menos
- $\times$  - Multiplicação
- $n$  - Número de amostras
- $( )$  - Parênteses
- $\%$  - Porcentagem
- $P_{lum}$  - Porcentagem de luminosidade
- $\sqrt{\phantom{x}}$  - Raiz quadrada
- $R1$  - Resistência variável do sensor de luminosidade
- $R2$  - Resistor fixo de 10k ohms.
- $R_{adc}$  - Resolução do conversor analógico-digital do Arduino
- $\Sigma$  - Somatório
- $T_c$  - Temperatura em graus Celsius
- $V_{in}$  - Tensão de entrada (geralmente 5V fornecida pelo Arduino)
- $V_{ref}$  - Tensão de referência do Arduino
- $V_{offset}$  - Tensão de referência para 0°C do sensor
- $V_{out}$  - Tensão de saída (tensão lida pelo Arduino)

$V_{sensor}$  - Tensão medida pelo sensor

$U(\%)$  - Umidade do solo em porcentagem

$L_{analog}$  - Valor lido pelo Arduino na entrada analógica

$S_{umid}$  - Valor lido pelo sensor de umidade

$L_{max}$  - Valor máximo da leitura analógica

$S_{max}$  - Valor máximo lido pelo sensor de umidade

V - Volts

## SUMÁRIO

1. INTRODUÇÃO.....	15
1.1 Justificativa do trabalho.....	16
1.2 Objetivo do Trabalho.....	17
1.2.1 Objetivo Geral.....	17
1.2.2 Objetivo Específico.....	17
1.3 Estrutura do Trabalho.....	18
2. FUNDAMENTO TEÓRICO.....	19
2.1 Análise de Mercado.....	19
2.2 Tecnologias Envolvidas no Hardware.....	20
2.2.1 Sistemas de Automação.....	21
2.2.2 Arduino UNO.....	21
2.2.3 Sensores.....	22
2.2.4 Atuadores.....	22
2.2.5 ESP01.....	23
2.2.6 Display LCD 20x4 I2C.....	23
2.3 Aplicação de IoT na Jardinagem.....	24
2.4 Tecnologias Utilizadas no Software.....	24
2.4.1 Frontend.....	25
2.4.1.1 ReactJS.....	25
2.4.1.2 Axios.....	25
2.4.2 Backend.....	26
2.4.2.1 Node.JS.....	26
2.4.2.2 Express.....	27
2.4.2.3 JWT.....	27
2.4.2.4 Bcrypt.....	28
2.4.3 Banco de Dados.....	28
2.4.3.1 MySQL.....	28
2.4.3.2 MySQL Workbench.....	29
2.4.3.3 SQL.....	29
2.4.3.4 Técnicas de Otimização.....	29
2.4.3.5 Técnicas de Normalização.....	30
2.4.4 Design.....	30
2.4.4.1 Minimalismo.....	30
2.4.4.2 Figma.....	31
2.5 Estudo de Caso.....	31
3. DESENVOLVIMENTO.....	32
3.1 Hardware.....	33
3.1.1 Montagem do Protótipo.....	33
3.1.2 Esquemático do Circuito.....	37
3.1.3 Fluxograma do Funcionamento do Hardware.....	38
3.1.4 Cálculos.....	39
3.1.4.1 Calibragem do Sensor de Umidade de Solo.....	40

3.1.4.2 Calibragem do Sensor de Temperatura.....	41
3.1.4.3 Calibragem do Sensor de Luminosidade.....	43
3.2 SOFTWARE.....	44
3.2.1 Design da Interface.....	45
3.2.1.1 Funcionalidades do Sistema.....	46
3.2.2 Desenvolvimento da Aplicação.....	49
3.2.2.1 Cadastro de Usuário.....	49
3.2.2.2 Login do Usuário.....	52
3.2.2.3 Cadastro das Plantas.....	56
3.2.2.4 Informações Principais da Planta.....	57
3.2.2.5 Relatório.....	58
3.2.3 Banco de Dados.....	59
3.2.3.1 Estrutura do Banco de Dados.....	59
3.2.3.2 Otimização.....	63
4. TESTES E RESULTADOS.....	64
4.1 Testes Funcionais.....	64
4.1.1 Hardware.....	64
4.1.1.1 Sensor de Temperatura.....	64
4.1.1.2 Sensor de Luminosidade.....	65
4.1.1.3 Sensor de Umidade de Solo.....	65
4.1.1.4 Funcionamento dos Componentes.....	65
4.1.2 Software.....	66
4.2 Resultados Obtidos.....	71
4.3 Conclusão dos Testes.....	71
5. CONSIDERAÇÕES FINAIS.....	72
5.1 Avaliação dos Objetivos Específicos.....	72
5.1.1 Integração de Sensores.....	72
5.1.2 Implementação de Banco de Dados.....	72
5.1.3 Desenvolvimento de Interface Web.....	73
5.1.4 Criação de Paineis Interativos.....	73
5.1.5 Implementação de Segurança de Dados.....	73
5.2 Contribuições e Impactos.....	73
5.3 Trabalhos Futuros.....	73
6. REFERÊNCIAS.....	75
APÊNDICE A - Código Fonte do Software.....	82
APÊNDICE B - Código Fonte do Hardware.....	83
APÊNDICE C - Código Fonte do Banco de Dados.....	84

# 1. INTRODUÇÃO

No contexto atual, a integração de tecnologias inovadoras no cuidado com plantas tem ganhado crescente relevância, especialmente para aqueles que desejam manter um ambiente verde e saudável, mas enfrentam limitações de tempo ou experiência no cultivo. Cada planta possui necessidades específicas e o cuidado com elas envolve monitorar fatores variados, como umidade do solo, temperatura ambiente e luminosidade. Reconhecendo essas particularidades, o vaso inteligente oferece uma solução automatizada e personalizada para o cuidado de plantas, desenvolvendo um protótipo de vaso inteligente que combina hardware e software para monitorar e atender às necessidades específicas de cada planta.

Como ilustrado na Figura 1, o protótipo do vaso inteligente é composto por um Arduino Uno R3, que monitora sensores de umidade do solo, luminosidade e temperatura, além de automatizar a irrigação por meio de uma bomba peristáltica. Os dados obtidos pelos sensores são transmitidos para um banco de dados via rede Wi-Fi, com suporte do módulo ESP01. O protótipo também inclui um display LCD, que cria uma interação não verbal com o usuário através de emojis de reação que expressam o “estado de humor” da planta, proporcionando uma experiência mais intuitiva e envolvente com o usuário.



Figura 1: Protótipo do Hardware

Fonte: Autor.

A Figura 2 ilustra o software desenvolvido em Javascript para o projeto de conclusão de curso. Este software é projetado para exibir dados sobre temperatura, umidade e luminosidade, processando essas informações em relatórios e gráficos que permitem monitorar o ciclo de vida da planta. O desenvolvimento da interface interativa é realizado com ReactJS, que facilita a reutilização de componentes. NodeJS é empregado para gerenciar a lógica do servidor e garantir a manutenção e integração entre frontend e backend, enquanto o Express simplifica a criação de APIs RESTful. Para assegurar a segurança, JWT é utilizado para autenticação sem a necessidade de manter o estado no servidor, e Bcrypt protege as senhas armazenadas através de técnicas avançadas de hashing. Esta combinação de tecnologias proporciona um sistema robusto e eficiente para o monitoramento da planta.



Figura 2: Aplicação WEB

Fonte: Autor

## 1.1 Justificativa do trabalho

A crescente integração entre tecnologia e sustentabilidade está impulsionando o desenvolvimento de soluções inovadoras para o cultivo de plantas em ambientes domésticos. Muitas pessoas desejam manter plantas, mas muitas vezes carecem do conhecimento necessário ou enfrentam desafios diários, como ausências e esquecimentos, resultando em cuidados insuficientes. Um sistema inteligente e automatizado pode solucionar esses problemas, gerenciando a irrigação e monitorando a condição da planta.

O uso de tecnologias emergentes, como a Internet das Coisas (IoT), para criar um vaso inteligente permite um uso mais eficiente dos recursos, reduzindo desperdícios e facilitando o cuidado com plantas, mesmo para aqueles com pouco tempo ou experiência. Para aprofundar os benefícios e a aplicabilidade do protótipo, será realizado um estudo de

caso com um pé de tomate-cereja, monitorado ao longo do desenvolvimento deste trabalho. Esse estudo permitirá avaliar a eficácia do sistema na manutenção das condições ideais para o crescimento da planta.

## 1.2 Objetivo do Trabalho

O objetivo do trabalho é desenvolver uma solução tecnológica que auxilie no cultivo doméstico de plantas, utilizando sensores para monitoramento das condições ambientais e promovendo práticas sustentáveis de jardinagem.

### 1.2.1 Objetivo Geral

Desenvolver um sistema de monitoramento acessível para o uso doméstico que forneça informações sobre o desenvolvimento da planta, com o intuito de auxiliar e incentivar o cultivo, minimizando os impactos negativos decorrentes de práticas inadequadas, seja por falta de conhecimento ou por cuidados inadequados e promover o engajamento dos usuários com práticas sustentáveis de jardinagem.

### 1.2.2 Objetivo Específico

Para a viabilizar o objetivo geral e finalizar o projeto, foram definidas as seguintes entregas necessárias para alcançar os resultados desejados:

**Integração de sensores:** Integrar sensores de umidade, temperatura e luminosidade ao sistema, garantindo a coleta de dados.

**Implementação de Banco de dados:** Implementar um banco de dados que armazene os dados coletados.

**Desenvolvimento de Interface Web:** Criar uma interface web que facilite a interação dos usuários com o sistema permitindo o monitoramento remoto.

**Criação de Painéis Interativos:** Criar painéis interativos que exibem gráficos de umidade, temperatura e luminosidade.

**Implementação de Segurança de Dados:** Implementar recursos de segurança para proteger os dados dos usuários e garantir que as informações coletadas sejam armazenadas de forma segura.

### 1.3 Estrutura do Trabalho

O trabalho está organizado em cinco capítulos principais, cada um com um foco específico para garantir uma compreensão clara e completa do desenvolvimento e dos resultados do projeto.

Neste capítulo, apresentamos a justificativa do trabalho, detalhando as motivações e relevância do projeto. Definimos os objetivos gerais e específicos, e fornecemos uma visão geral do que será abordado ao longo do trabalho.

O segundo capítulo introduz a fundamentação teórica como conceitos e assuntos básicos necessários para a compreensão do tema abordado. Serão discutidos os fundamentos teóricos que sustentam o desenvolvimento do sistema, incluindo as tecnologias utilizadas. Também é abordado o sobre o estudo de caso que será utilizado como base para o desenvolvimento do vaso inteligente.

No terceiro capítulo, abordaremos detalhadamente o desenvolvimento do aplicativo. Apresentando as ferramentas e tecnologias, a estrutura do aplicativo, e os conceitos considerados no design e implementação, oferecendo uma visão abrangente das escolhas realizadas durante o desenvolvimento.

O quarto capítulo apresenta os testes e resultados obtidos com o sistema. Discutindo os métodos de avaliação utilizados.

Por fim, no quinto e último capítulo resumimos os principais achados, e sugerimos possíveis direções para trabalhos futuros.

## 2. FUNDAMENTO TEÓRICO

Nesta seção, exploramos os conceitos e tecnologias que fundamentam o desenvolvimento de sistemas de jardinagem inteligente, abordando desde as soluções automatizadas existentes até as tecnologias utilizadas no nosso projeto.

### 2.1 Análise de Mercado

As tecnologias de jardinagem inteligente têm revolucionado o cuidado com as plantas, oferecendo soluções automatizadas que tornam o cultivo mais simples e eficiente (InovadorTech, 2024). Dispositivos como Smart Garden 3, Lua e Ivy monitoram o ambiente das plantas para garantir um crescimento saudável.

O The Smart Garden 3, mostrado na figura 3, automatiza a irrigação e fornece uma luz de crescimento ajustável, otimizando o desenvolvimento das plantas enquanto mantém um baixo consumo de energia (CLICK AND GROW, 2023).



Figura 3: The Smart Garden 3

Fonte: CLICK AND GROW

O Lua, mostrado na figura 4, utiliza sensores para monitorar o nível de água, luz e temperatura, convertendo essas informações em expressões visuais em uma tela. Seu sistema de sub-irrigação assegura que a planta receba a quantidade adequada de água conforme necessário (MULLER, 2023).



Figura 4: Lua

Fonte: MULLER

O Ivy, mostrado na figura 5, monitora fatores como temperatura, umidade e luminosidade, ajustando automaticamente as condições do ambiente e fornecendo feedback visual interativo ao usuário (SMART PLANT IVY, 2023).



Figura 5: Ivy

Fonte: SMART PLANT IVY

O Vaso Samanta foi desenvolvido buscando integrar as melhores funcionalidades de cada um dos modelos citados anteriormente. Aproveitando o monitoramento ambiental e a automação de irrigação do The Smart Garden 3, o uso de expressões visuais inspiradas no sistema de feedback do Lua e a interatividade do Ivy, o Samanta oferece um sistema completo e adaptado às necessidades de cada planta. Com sensores de temperatura, umidade e luminosidade, além de uma tela LCD que transforma essas informações em expressões visuais do “estado” da planta, o Samanta proporciona monitoramento em tempo real e uma experiência de cultivo mais prática e acessível.

## 2.2 Tecnologias Envolvidas no Hardware

A seção de Tecnologias Envolvidas aborda os componentes tecnológicos essenciais que sustentam o funcionamento do sistema de jardinagem inteligente. Esta seção destaca as tecnologias utilizadas, que são fundamentais para garantir o monitoramento eficaz e a automação dos cuidados com as plantas.

No Vaso Samanta, as tecnologias envolvidas incluem sensores para monitorar as condições ambientais das plantas e sistemas de automação para gerenciar e controlar essas condições. Essas tecnologias são integradas para criar um sistema inteligente que proporciona uma solução prática e eficiente para o cultivo doméstico de plantas.

## 2.2.1 Sistemas de Automação

Os sistemas de automação são fundamentais para a modernização e otimização de processos, tanto em ambientes industriais quanto residenciais. Utilizando tecnologias avançadas, esses sistemas controlam e aprimoram operações, reduzindo a intervenção humana e minimizando erros. De acordo com Bayer, Eckhardt e Machado (2011, p. 11), a palavra "sistema" deriva do grego *sietemiun*, que significa "combinar" ou "formar um conjunto". Portanto, um sistema é um conjunto de elementos interconectados que formam um todo organizado com o objetivo de alcançar metas específicas. Por sua vez, "automação" ou "automatização" vem do latim *automatus*, que significa "mover-se por si mesmo". Dessa forma, automação refere-se à capacidade de um equipamento para operar de maneira automática, sem a necessidade de intervenção humana (BAYER; ECKHARDT; MACHADO, 2011, p. 11).

Um exemplo da aplicação de sistemas de automação é o nosso vaso inteligente. Este dispositivo utiliza sensores para monitorar temperatura, umidade e luminosidade da planta selecionada, ajustando automaticamente a umidade do solo por meio de irrigação conforme as preferências da espécie, previamente definidas em nosso catálogo de plantas. Com essa tecnologia, o vaso otimiza as condições de crescimento das plantas sem exigir supervisão constante, facilitando o cuidado para quem deseja simplificar o manejo de suas plantas. A automação do vaso não só promove a saúde das plantas, mas também oferece uma solução prática e confiável para manter o ambiente ideal para seu desenvolvimento, destacando a versatilidade e a importância dos sistemas automatizados no dia a dia contemporâneo.

## 2.2.2 Arduino UNO

O Arduino UNO é uma placa microcontroladora baseada no chip ATmega328P. Ela possui 14 pinos digitais de entrada/saída, dos quais 6 podem ser utilizados como saídas PWM, e 6 entradas analógicas. Inclui um ressonador cerâmico de 16 MHz, uma conexão USB, uma entrada de energia, um cabeçote ICSP e um botão de reset. A placa vem equipada com todos os recursos necessários para o funcionamento do microcontrolador; é suficiente conectá-la a um computador por meio de um cabo USB ou alimentá-la com um adaptador AC-DC ou bateria para iniciar seu uso (ARDUINO, s.d.).

Entre as diversas opções de microcontroladores disponíveis no mercado, o Arduino UNO R3 foi selecionado devido à sua facilidade de uso e custo reduzido, o que o torna a alternativa mais acessível e conveniente para o projeto.

### 2.2.3 Sensores

Um sensor é um dispositivo projetado para detectar e responder de maneira eficaz a diferentes estímulos. Há diversos tipos de sensores que reagem a estímulos variados, como calor, pressão, movimento, luz, entre outros. Após captar o estímulo, o sensor emite um sinal que pode ser convertido e interpretado por outros dispositivos (MATTENE, 2024). Os sensores empregados no protótipo do vaso inteligente são analógicos, o que lhes permite medir uma ampla gama de intensidades de sinais.

As entradas analógicas podem interpretar variações de 0 a 5V, pois estão conectadas a conversores analógico-digital (ADC). Esses conversores convertem as medidas analógicas do mundo real em valores digitais que os dispositivos eletrônicos podem entender. A resolução do ADC é dada pelo número de bits. Para a maioria das placas Arduino, as entradas analógicas têm uma resolução de 10 bits, o que corresponde a  $2^{10} = 1024$  divisões de 0 a 5 V. Isso implica que a menor variação que o Arduino pode interpretar é de 0,005 V. (ROBOCORE, n.d., p. 13)

Para o projeto final, foram escolhidos o sensor de umidade do solo, o sensor de temperatura TMP36 e o sensor de luminosidade. Com esses sensores, espera-se que seja possível monitorar remotamente as condições de saúde de uma planta.

### 2.2.4 Atuadores

No Vaso Samanta, além dos sensores, também são utilizados atuadores. Conforme descrito por BALBINO DA COSTA et al. (2021, p. 19), “são dispositivos com a finalidade de produzir movimento. Recebem comandos manuais ou automáticos e modificam o ambiente com ações”. No projeto, são utilizados uma bomba peristáltica de 12V, que opera de maneira semelhante a motores comuns, e um módulo relé de 5V.

Os motores elétricos operam através de impulsos magnéticos gerados pela variação do fluxo magnético nas bobinas entre as espiras (BALBINO DA COSTA et al., 2021, p. 20). A rotação do motor é resultado da corrente elétrica que passa pelas espiras, gerando forças magnéticas que causam o movimento (BALBINO DA COSTA et al., 2021, p. 20). Esses motores podem ser alimentados por DC usando baterias ou pilhas, ou por AC conectando-os diretamente à rede elétrica. Existem também motores que podem operar com ambos os tipos de corrente (BALBINO DA COSTA et al., 2021, p. 20).

Já os relés são dispositivos eletromecânicos projetados para operar com pequenas correntes e controlar circuitos que lidam com correntes mais altas (BALBINO DA COSTA et al., 2021, p. 20). Quando uma corrente passa pela bobina do relé, ela cria um campo

magnético que aciona um mecanismo para abrir ou fechar os contatos. Enquanto houver alimentação na bobina, os contatos permanecem na posição definida, permitindo ou bloqueando a passagem de energia. O funcionamento do relé é semelhante ao de um interruptor de parede doméstico, mas, em vez de controlar fisicamente um circuito, ele faz isso através da passagem ou bloqueio de energia de um outro circuito (BALBINO DA COSTA et al., 2021, p. 20).

## 2.2.5 ESP01

Para conectar o projeto à web, foi escolhido o microcontrolador ESP01, que faz parte da família ESP8266. Segundo OLIVEIRA (n.d.), o ESP8266 é um módulo crucial para projetos de Internet das Coisas (IoT), possuindo um microprocessador ARM de 32 bits e memória flash integrada. Esse módulo oferece suporte completo para redes Wi-Fi e permite sua programação independente, sem a necessidade de outras placas de automação. O modelo ESP8266 ESP-01 é especialmente versátil e pode ser integrado ao Arduino para adicionar conectividade sem fio aos projetos. Ele é compatível com redes Wi-Fi 802.11 b/g/n e oferece criptografia WPA e WPA2, garantindo uma conexão segura e estável (OLIVEIRA, n.d.).

No contexto deste projeto, o ESP01 está sendo utilizado não como um microcontrolador, mas como um atuador do Arduino UNO. Devido à limitação de pinos analógicos disponíveis no Arduino para a leitura dos sensores do protótipo, a solução encontrada foi usar o ESP01 como uma ponte para o envio de dados para o servidor.

## 2.2.6 Display LCD 20x4 I2C

Para proporcionar uma experiência mais intuitiva e agradável aos usuários do nosso vaso, foi adicionado ao sistema um Display LCD 20x4 I2C. Esse tipo de tela é amplamente encontrado em diversos aparelhos, como monitores, televisores, câmeras digitais, relógios, calculadoras e telefones celulares, inclusive smartphones (ROBOCORE, n.d., p. 37). As telas de cristal líquido utilizam as propriedades dos cristais líquidos para modular a luz e apresentar informações eletrônicas, como texto, imagens e vídeos. Por não emitirem luz própria, esses cristais necessitam de uma luz de fundo ou refletor para gerar imagens, sejam elas coloridas ou monocromáticas (ROBOCORE, n.d., p. 37).

O módulo I2C foi incorporado ao Display LCD com a finalidade de simplificar a interface de comunicação com o Arduino, reduzindo a quantidade de conexões necessárias. O

protocolo I2C, originalmente criado pela Philips (hoje NXP), conecta vários dispositivos periféricos utilizando apenas duas linhas de comunicação: SDA e SCL (ROBOCORE, n.d.).

É interessante ressaltar que, inicialmente, o Display LCD foi idealizado para incentivar o usuário a ver o vaso como uma espécie de “bichinho virtual”, semelhante ao famoso Tamagotchi, um dispositivo eletrônico de entretenimento dos anos 90 que simulava o cuidado de um animal virtual (BANDAI, s.d.).

## 2.3 Aplicação de IoT na Jardinagem

A IoT refere-se à interconexão de dispositivos e sensores pela internet, permitindo que eles coletam, troquem e analisem dados em tempo real. Na jardinagem, essa tecnologia facilita o monitoramento preciso das condições ambientais, como umidade do solo, temperatura e luminosidade, permitindo a automação da irrigação e a otimização do uso de recursos, como água e nutrientes, resultando em práticas de jardinagem mais sustentáveis e eficazes (INOVADORTech, 2024).

A aplicação de IoT no projeto visa criar um sistema inteligente de monitoramento e controle de plantas, utilizando sensores conectados, o projeto permite a coleta de dados em tempo real sobre as necessidades das plantas, ajustando automaticamente a irrigação.

## 2.4 Tecnologias Utilizadas no Software

Esta seção detalha as tecnologias relacionadas aos dispositivos de jardinagem inteligente e seus componentes principais. Dispositivos de jardinagem inteligente geralmente incorporam várias tecnologias para monitorar e automatizar o cuidado com as plantas. Essas tecnologias incluem sensores para medir condições ambientais, sistemas de automação para controlar dispositivos e comunicação para interação com os usuários. Combinadas, essas tecnologias permitem que os dispositivos ajustem as condições de cultivo para otimizar o crescimento das plantas e melhorar a eficiência do cuidado. (SANTOS, 2024).

Neste protótipo, várias tecnologias foram selecionadas para garantir que a aplicação de jardinagem inteligente seja fácil de usar. Cada tecnologia foi escolhida com base em suas capacidades e compatibilidade com os objetivos do projeto, desde o design da interface até a gestão dos dados e a interação do usuário com a aplicação.

## 2.4.1 Frontend

O frontend refere-se à interface gráfica do usuário (GUI) com a qual as pessoas interagem, incluindo menus de navegação, elementos de design, botões, imagens e gráficos. Em termos técnicos, a estrutura visual que o usuário vê em uma página ou tela é representada pelo Modelo de Objeto de Documento (DOM). O DOM organiza e descreve a estrutura dos elementos da página, permitindo que o navegador renderize e manipule dinamicamente o conteúdo exibido ao usuário, adaptando-se às interações em tempo real (AWS, 2024).

### 2.4.1.1 ReactJS

React é uma biblioteca de JavaScript desenvolvida pela Meta, amplamente utilizada para criar interfaces de usuário em aplicações web. Sua modularidade permite que os componentes sejam reutilizáveis e compartilhados entre diferentes partes da aplicação, facilitando a manutenção e a escalabilidade à medida que o projeto cresce. Isso significa que, em caso de erro em algum componente, a correção pode ser feita de forma mais isolada e eficiente. Devido a essas características, o React se tornou uma das bibliotecas JavaScript mais populares. (NEVES, 2023).

Além disso, de acordo (WLADIMIR, 2022), o React utiliza uma extensão para JavaScript chamada JSX, que permite criar elementos diretamente dentro do código JS, facilitando a incorporação de código HTML em objetos JS. Isso simplifica a renderização de múltiplas funções e melhora a legibilidade do código.

### 2.4.1.2 Axios

De acordo com a documentação oficial do Axios, ele é um cliente HTTP baseado em promessas, compatível tanto com Node.js quanto com navegadores. No lado do servidor, o Axios utiliza o código nativo do Node.js para realizar as requisições, enquanto no lado do cliente, ele faz uso do objeto XMLHttpRequest. Abordaremos mais detalhes sobre o uso do Node.js no backend no tópico específico, mas é importante destacar que sua flexibilidade torna o Axios uma ferramenta essencial para a comunicação entre o frontend e o backend em aplicações web.

Os métodos HTTP utilizados neste projeto foram: "GET", "POST". O backend retorna os dados no formato JSON, que é uma notação de objetos JavaScript amplamente utilizada para o intercâmbio de dados. No frontend, o Axios foi utilizado para consumir essas

rotas, facilitando a comunicação e permitindo que a aplicação interaja de forma eficiente com o servidor.

## 2.4.2 Backend

O backend da aplicação gerencia a funcionalidade geral do seu aplicativo web. Interagindo com o banco de dados, para recuperar ou modificar dados relevantes. O back-end combina técnicas de simultaneidade e paralelismo, como distribuição de solicitações em vários servidores, armazenamento em cache e duplicação de dados. (AWS, 2024).

### 2.4.2.1 Node.JS

O Node.js é um ambiente de execução de código JavaScript do lado servidor (server-side), o que significa que permite criar aplicações independentes (standalone) diretamente em uma máquina servidora, sem a necessidade de um navegador para sua execução, (ALURA, n.d.).

Com o Node.js, é possível utilizar JavaScript tanto no frontend quanto no backend, o que facilita a manutenção e o compartilhamento de lógica entre as duas camadas da aplicação. Além disso, o Node.js inclui o NPM, um repositório de pacotes que simplifica a adição de funcionalidades de terceiros ao projeto. O Node.js é especialmente ideal para a construção de micro serviços e APIs, que são componentes fundamentais em arquiteturas de software modernas.

Foi utilizada a arquitetura baseada em Rotas, Controladores e Serviços, onde cada um tem uma responsabilidade. O roteamento é responsável por direcionar as requisições recebidas para as rotas corretas. Com o Node.js, é possível utilizar frameworks como o Express para definir e gerenciar as rotas do sistema de forma eficiente. Os controladores são responsáveis por receber as requisições, processar as informações necessárias e enviar as respostas adequadas. Eles encapsulam a lógica de negócios do sistema e podem fazer uso de modelos e serviços para realizar operações mais complexas. Os serviços são responsáveis por implementar funcionalidades específicas do sistema que podem ser utilizadas por diferentes partes da aplicação. Eles encapsulam a lógica de negócios mais complexa e podem ser reutilizados em diferentes contextos.

#### 2.4.2.2 Express

O Express foi utilizado no projeto para gerenciar requisições HTTP, facilitando o roteamento de endpoints através de diferentes métodos e URLs. Segundo o (MDN WEB DOCS, 2022), o Express simplifica também a configuração da aplicação, como a definição da porta de conexão e a localização dos modelos usados para gerar respostas. Outro recurso poderoso do Express é a capacidade de adicionar processos de requisição por meio de middleware, permitindo a inserção de funcionalidades em qualquer ponto da "fila" de requisições, o que torna o desenvolvimento mais modular e flexível. Além disso, o Express possui diversas bibliotecas disponíveis para trabalhar com cookies, sessões, login de usuários, parâmetros de URL, dados em requisições POST, cabeçalhos de segurança e outras funcionalidades essenciais.

#### 2.4.2.3 JWT

A questão da segurança de dados tornou-se um dos temas mais discutidos em aplicações de TI, especialmente no que diz respeito à proteção de informações sensíveis dos usuários, como e-mails e senhas. A segurança deve ser um aspecto fundamental em qualquer aplicação web, independentemente do seu tamanho, uma vez que todas as aplicações contêm dados que necessitam de proteção. Como utilizamos um sistema de login, para realizar a autenticação e autorização na API, optamos pelo uso da abordagem mais comum: o token JWT.

De acordo com (LIMA, 2021), o JSON Web Token é um padrão da indústria definido pela RFC 7519, tem como objetivo de transmitir ou armazenar objetos JSON de forma compacta e segura entre diferentes aplicações. O JWT é utilizado em dois principais cenários: quando há necessidade de realizar um processo de autorização em uma aplicação ou quando se deseja realizar a troca de informações entre sistemas.

A estrutura do JWT consiste em três partes: o Header (Cabeçalho), o Payload (Corpo) e a Signature (Assinatura). Segundo Lima (2021), o cabeçalho é composto por duas partes: o tipo do token e o algoritmo de assinatura, como HMAC, SHA-256 ou RSA. O corpo do token contém as claims, que são declarações sobre uma entidade, como usuário, além de dados adicionais. A parte da assinatura é formada pelo cabeçalho codificado, o payload codificado, a chave secreta e o algoritmo especificado no cabeçalho, que juntos são utilizados para gerar a assinatura digital do token.

#### 2.4.2.4 Bcrypt

Para evitar acessos não autorizados aos sistemas, utilizamos no projeto alguns métodos de proteção de dados, como a criptografia de senhas, que dificulta o acesso a informações armazenadas em bancos de dados. Segundo (KERLYN, 2019) o BCrypt foi desenvolvido com o propósito de ocultar senhas criadas pelos usuários, convertendo-as de texto puro em dados indecifráveis através do uso de algoritmos de hash. Essa ferramenta é considerada segura para armazenar senhas em bancos de dados e pode ser utilizada em diversas linguagens de programação.

O BCrypt possui aspectos importantes, como o uso de salt, que consiste em adicionar aleatoriamente sequências de caracteres a senha, resultando em criptografias complexas e aumentando a segurança contra ataques de força bruta.

#### 2.4.3 Banco de Dados

Os bancos de dados são essenciais para o armazenamento e gerenciamento de informações em sistemas digitais, permitindo a organização eficiente e a consulta rápida de dados. Eles facilitam o armazenamento permanente de dados e a sua recuperação quando necessário, desempenhando um papel crucial na integridade e performance das aplicações (PICHETTI et al., 2020, p. 11-12).

Para o sistema, utilizamos o MySQL para armazenar e gerenciar os dados, enquanto o MySQL Workbench foi empregado para modelar e administrar o banco de dados. Aplicamos técnicas de otimização e normalização para garantir que o sistema fosse eficiente e que os dados fossem integrados e acessíveis de forma eficaz.

##### 2.4.3.1 MySQL

O MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto, e atualmente pertencente à Oracle Corporation (GOOGLE CLOUD, 2024). Amplamente utilizado por sua eficiência e escalabilidade, o MySQL emprega a linguagem SQL para organizar e consultar dados em tabelas inter-relacionadas (MANZANO, 2011, p. 20-22).

A escolha desse sistema para o projeto se deve à sua capacidade de gerenciar grandes volumes de dados com alta performance e escalabilidade.

#### 2.4.3.2 MySQL Workbench

O MySQL Workbench é uma ferramenta gráfica para design e administração de bancos de dados MySQL, que permite modelagem de dados, criação de esquemas e execução de consultas SQL de forma visual e intuitiva (PICHETTI et al., 2020, p. 66-67).

A aplicação foi utilizada para criar e ajustar o esquema do banco de dados, além de facilitar a execução de consultas SQL e visualização de dados, garantindo a eficiência e integridade das operações.

#### 2.4.3.3 SQL

SQL é a linguagem padrão projetada para acessar e gerenciar bancos de dados relacionais. Desenvolvida inicialmente pela IBM na década de 1970 e agora padronizada pelo ANSI, SQL permite realizar operações essenciais como consultas, inserções, atualizações e exclusões de dados. A linguagem utiliza uma sintaxe simples e é dividida em várias categorias de comandos, incluindo DML, DDL, TC e DCL (PRICE, 2009, p. 31-32).

No sistema implementado o SQL é utilizado para interagir com o banco de dados MySQL, facilitando a execução de operações como consultas e atualizações de dados. As consultas SQL permitem a recuperação e modificação eficiente de informações relacionadas às condições das plantas e preferências dos usuários. A utilização do SQL foi crucial para implementar funcionalidades no sistema.

#### 2.4.3.4 Técnicas de Otimização

Técnicas de otimização são práticas empregadas para melhorar o desempenho de sistemas e bancos de dados, visando aumentar a eficiência e a velocidade das operações. Entre essas técnicas estão a criação de índices e a simplificação de consultas. A criação de índices, por exemplo, utiliza estruturas binárias para acelerar a busca e o acesso aos dados, resultando em uma melhoria significativa na velocidade das consultas (IBM, 2021). Além disso, a simplificação de consultas e a eliminação de redundâncias ajudam a reduzir o tempo de processamento e o uso de recursos. A implementação dessas técnicas é crucial para assegurar que o sistema responda rapidamente e suporte grandes volumes de transações, proporcionando uma experiência de usuário mais eficiente e satisfatória (SANTOS, 2024).

As técnicas de otimização foram aplicadas para melhorar a performance do banco de dados, incluindo a criação de índices para acelerar consultas específicas, simplificação das queries SQL e eliminação de redundâncias.

#### 2.4.3.5 Técnicas de Normalização

A normalização é um processo que organiza os dados em um banco de dados para reduzir redundâncias e dependências inconsistentes. Ela envolve dividir tabelas grandes em estruturas menores e inter-relacionadas, seguindo regras conhecidas como formas normais. Esse processo melhora a integridade dos dados e a eficiência do banco de dados, prevenindo problemas como desperdício de espaço e dificuldades de manutenção (MICROSOFT, 2024).

Para otimizar o banco de dados, aplicamos técnicas de normalização, dividimos tabelas em partes menores e mais organizadas, o que resultou na redução da redundância e facilitou a manutenção.

#### 2.4.4 Design

Segundo o canal (DESIGNERVERSO, 2021), o design é uma especialidade multidisciplinar da comunicação que cria projetos visuais com o objetivo de apresentar soluções para resolver problemas do ponto de vista estético, simbólico e prático. O design pode incluir várias vertentes, como design gráfico, design de interação, e design de interface, cada uma com seu foco e técnicas específicas.

No desenvolvimento da aplicação, o design foi fundamental para garantir que a interface do usuário seja intuitiva e eficaz. Adotamos princípios de design para criar uma aplicação que seja visualmente agradável e funcional, garantindo que os usuários possam interagir com os dispositivos de jardinagem inteligente de forma fácil e eficiente. A estética e a funcionalidade são equilibradas para proporcionar uma experiência de usuário satisfatória e envolvente.

##### 2.4.4.1 Minimalismo

O minimalismo no design é uma abordagem que busca a simplicidade e a clareza, eliminando elementos desnecessários para focar apenas no essencial. Segundo o (BLOG DA ADOBE, 2022), o design minimalista facilita a navegação ao reduzir o número de elementos visuais e utilizar uma paleta de cores restrita. Esse estilo promove uma experiência de usuário mais intuitiva, permitindo que o espectador encontre informações com mais rapidez e facilidade.

A abordagem foi adotada para garantir uma interface limpa e funcional. O design do site e da aplicação web segue os princípios do minimalismo para evitar sobrecarga de informações e proporcionar uma experiência de usuário clara e direta. Utilizamos uma paleta

de cores reduzida e uma disposição simplificada dos elementos para melhorar a interação e o foco do usuário nas funcionalidades principais da aplicação.

#### 2.4.4.2 Figma

O Figma serve para projetar design de interface e prototipagem que permite a criação colaborativa e em tempo real de layouts e protótipos. Com sua interface intuitiva e recursos avançados de design, o Figma facilita a criação de interfaces visuais interativas e a colaboração entre equipes de design. De acordo com a documentação oficial (FIGMA, 2024), a plataforma é amplamente utilizada para prototipagem de interfaces e design de produtos digitais, oferecendo ferramentas para design vetorial, prototipagem e feedback em tempo real.

Essa ferramenta foi crucial para a criação de protótipos e para o design da interface do usuário. Utilizamos o Figma para esboçar o layout, testar diferentes opções de design e obter feedback colaborativo durante o processo de desenvolvimento. A ferramenta nos permitiu criar uma interface eficiente e esteticamente agradável, alinhada aos princípios do minimalismo e às necessidades do projeto.

### 2.5 Estudo de Caso

Para validar o protótipo do Vaso Samanta, optou-se pelo cultivo de um pé de tomate-cereja como um estudo de caso, pois era uma planta que estava à disposição e que cabia em um vaso comum. Essa escolha permitiu realizar estudos e pesquisas detalhadas para determinar as medidas de referência necessárias à calibração precisa dos sensores, garantindo assim a sobrevivência e o desenvolvimento da planta.

No que se refere às condições ambientais ideais para o cultivo do tomate-cereja, observou-se que a temperatura é um fator crucial, variando entre 15°C e 30°C (PINTO, 2017, p. 20). Além disso, a luminosidade também desempenha um papel importante, devendo ser mantida em cerca de 70% ao longo do dia para assegurar o bom desenvolvimento da planta (SEMADE/MS, 2016). Outro aspecto fundamental é o teor de umidade do solo, que deve ser mantido dentro da margem mínima de 20% e 60% para promover um crescimento saudável (EOS, 2024).

Esses parâmetros são cruciais para o crescimento saudável da planta e foram considerados na calibração dos sensores do Vaso Samanta, visando otimizar as condições de cultivo e promover resultados eficazes.

### 3. DESENVOLVIMENTO

O desenvolvimento do vaso inteligente envolveu a integração de hardware e software para criar um sistema funcional e eficiente (Figura 6). Iniciamos com a seleção e montagem dos componentes de hardware, garantindo que cada parte funcionasse corretamente e de forma coordenada.

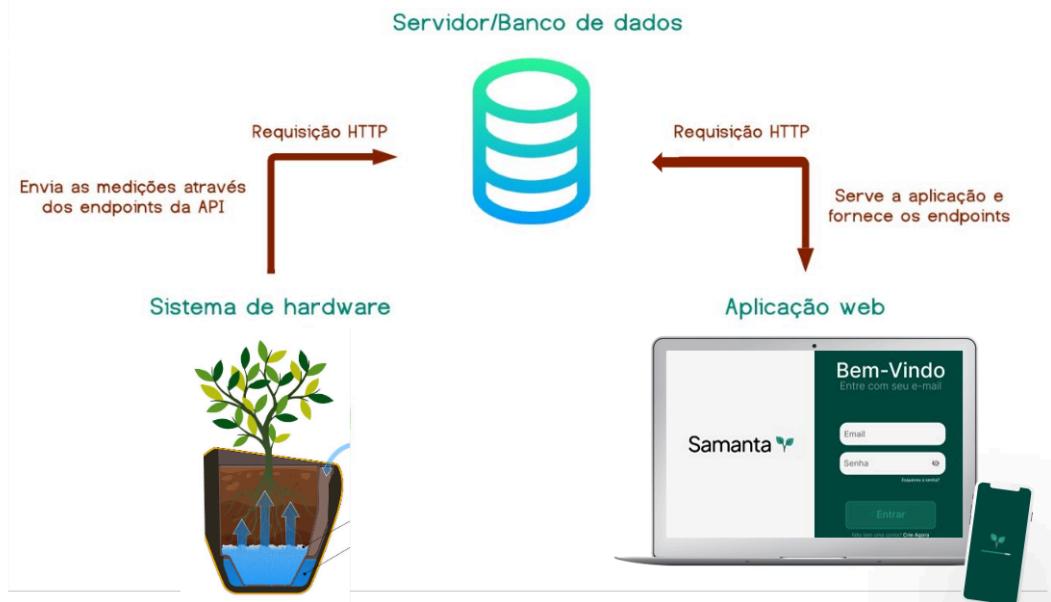


Figura 6: Arquitetura Geral do Projeto

Fonte: Autor

Simultaneamente, a aplicação web foi desenvolvida para servir como a interface principal do usuário, facilitando o monitoramento e controle do sistema. Esta interface foi projetada com um layout intuitivo e moderno, utilizando tecnologias para oferecer uma experiência de usuário eficiente e responsiva.

Além disso, o desenvolvimento da aplicação envolveu a criação de um back-end para gerenciar a comunicação entre o frontend e o hardware, além da configuração de um banco de dados para armazenar e gerenciar os dados do sistema.

A seguir, detalharemos as especificações e o funcionamento do hardware, bem como o desenvolvimento e as funcionalidades do software. Este aprofundamento fornecerá uma visão clara de como cada componente do sistema contribui para a operação geral do projeto.

### 3.1 Hardware

O sucesso de um protótipo depende não apenas de sua programação, mas também dos componentes de hardware que o compõem. No desenvolvimento de um vaso inteligente, a seleção e integração cuidadosa dos componentes de hardware são essenciais para garantir que todas as funcionalidades sejam executadas conforme planejado. Este tópico aborda os principais elementos de hardware utilizados no projeto, detalhando como cada um foi escolhido, configurado e implementado para atender às necessidades específicas do sistema. Conforme ilustrado na Figura 7, desde a escolha do microcontrolador Arduino Uno R3 até a implementação de sensores, atuadores e a interface de usuário com o Display LCD, cada decisão de hardware foi feita com o objetivo de criar um sistema eficiente, confiável e fácil de usar. A seguir, exploraremos os componentes utilizados, suas funções e como eles interagem para formar o coração físico do protótipo.

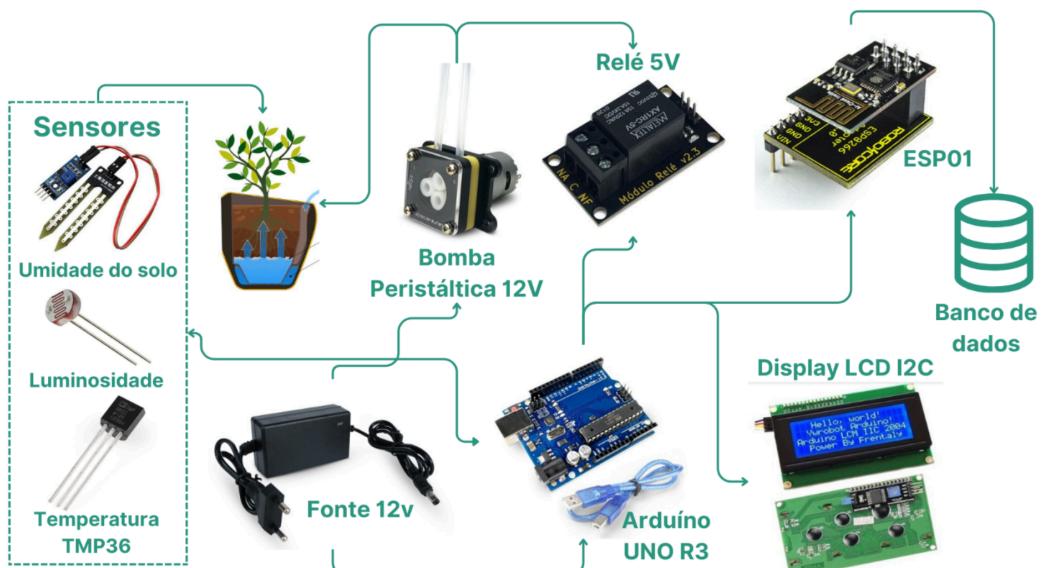


Figura 7: Arquitetura do Hardware

Fonte: Autor, AUTOCORE ROBÓTICA, CURTO CIRCUITO, MERCADO LIVRE e ROBOCORE

#### 3.1.1 Montagem do Protótipo

O Arduino exerce uma função central no projeto, atuando como o controlador principal responsável pela leitura dos dados dos sensores e pelo acionamento dos atuadores. Ele realiza a coleta de informações essenciais, como temperatura, luminosidade e umidade do solo ao longo do tempo, permitindo uma análise precisa das condições ambientais. Em

particular, o sensor de luminosidade exigiu um ajuste específico com a utilização de um resistor de 10k ohms, necessário para limitar a tensão dentro da faixa de 0 a 5V, garantindo maior precisão nas medições de luminosidade, conforme mostrado na Figura 3.

Para realizar as conexões dos sensores e atuadores ao Arduino, foi utilizada uma protoboard de 400 pontos, o que proporcionou uma montagem prática e eficiente do circuito. Além disso, para possibilitar o funcionamento remoto do projeto, foi empregado o microcontrolador ESP01, pertencente à família ESP8266. Como ilustrado na Figura 8, o ESP01 foi integrado ao Arduino por meio de um adaptador que permite seu uso com uma tensão de 5V. O ESP01 realiza a transmissão dos dados coletados pelos sensores para o servidor por meio de comandos AT, sendo esses dados armazenados em um banco de dados MySQL.

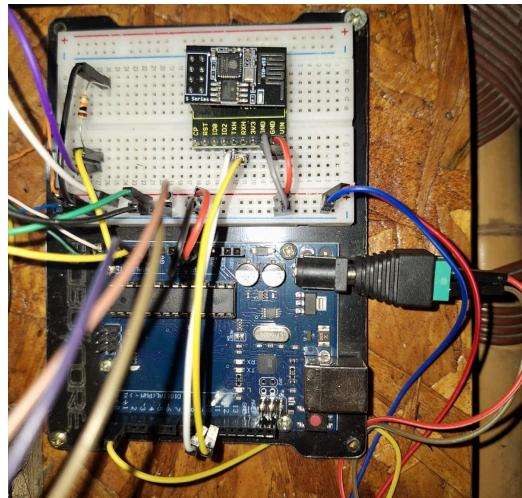


Figura 8: Arduino

Fonte: Autor

Os sensores foram estrategicamente posicionados para garantir uma colocação ideal em relação à planta, permitindo uma leitura precisa das suas necessidades. Essa disposição cuidadosa assegura que os sensores possam captar de maneira eficaz as variáveis ambientais, fornecendo dados essenciais para o monitoramento e ajuste das condições do cultivo, como apontado pelo estudo de caso.

No código do Arduino, algumas equações foram implementadas para converter as leituras brutas dos sensores em valores mais interpretáveis. Essas equações permitem que o Arduino processe e interprete as leituras dos sensores de maneira eficaz, garantindo que as condições ambientais da planta sejam monitoradas com precisão. A Figura 9 ilustra o

posicionamento final dos sensores, destacando como eles foram dispostos para otimizar a coleta de informações e, consequentemente, o cuidado com a planta.



Figura 9: Sensores

Fonte: Autor

O módulo Relé de 5V, conforme ilustrado na Figura 10, é utilizado para controlar a ativação e desativação da Bomba Peristáltica de 12V. Isso é feito através de sinais digitais enviados pelo próprio Arduino, que comanda o relé para abrir ou fechar o circuito da bomba. Dessa forma, o Arduino pode ligar ou desligar a bomba conforme necessário, de maneira segura e eficiente, garantindo o controle preciso do fluxo de água para a planta.

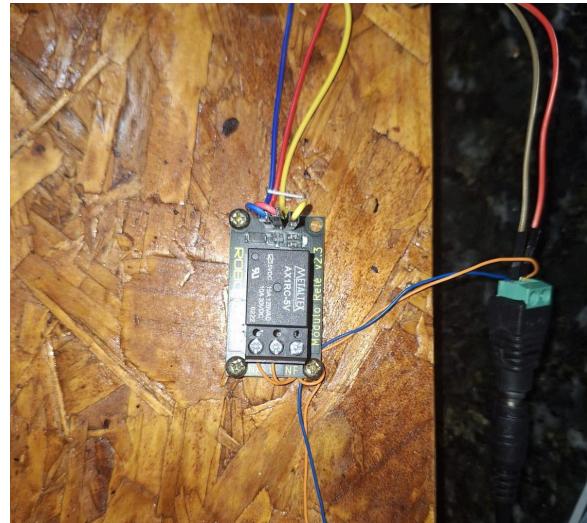


Figura 10: Módulo Relé 5V

Fonte: Autor

A Bomba Peristáltica de 12V, assim como o Arduino, é alimentada por uma fonte de 12V. Para garantir o funcionamento seguro da bomba, foi soldado um diodo 1n4007 na parte traseira do circuito, com o objetivo de direcionar a corrente em uma única direção, prevenindo possíveis danos causados por correntes reversas. Na Figura 11, é ilustrada a posição da Bomba Peristáltica na parte dianteira do protótipo, destacando sua importância no controle preciso do fluxo de líquidos no sistema.



Figura 11: Bomba Peristáltica 12V

Fonte: Autor

Na Figura 12, é possível observar o Display LCD 20x4 I2C, um componente central da interface do protótipo. Com base nas observações do estudo de caso, o display foi programado para exibir cinco variações de emojis de reação: "Feliz", "Neutro", "Triste", "Dormindo" e "Frio".

O emoji "Feliz" indica que o solo possui umidade suficiente para manter a planta bem hidratada, sendo exibido quando a umidade é igual ou superior a 60%. O emoji "Neutro" aparece quando o solo começa a secar, sinalizando que a planta pode precisar de água em breve. Essa reação ocorre quando a umidade do solo é igual ou superior a 30%, garantindo uma margem de segurança para acionar a bomba peristáltica antes que a planta sofra com a falta de água. O emoji "Triste" é exibido quando o solo está excessivamente seco, alertando o usuário sobre a necessidade urgente de hidratação para preservar a saúde da planta. Por fim, as reações "Frio" e "Dormindo" são ativadas quando a temperatura ambiente cai abaixo de 15°C ou a luminosidade está abaixo de 70%, indicando condições inadequadas para o crescimento da planta.

Esse sistema visual proporciona uma interação mais intuitiva e envolvente, facilitando o cuidado da planta e tornando o processo mais acessível para o usuário.



Figura 12: Display LCD 20X4 I2C

Fonte: Autor

### 3.1.2 Esquemático do Circuito

Para visualizar o diagrama eletrônico do projeto, foi elaborado um modelo esquemático do circuito utilizando o software EasyEDA, conforme mostrado na Figura 13. Esse esquemático oferece uma representação gráfica fundamental para compreender a disposição e a interligação dos componentes eletrônicos do sistema, além de mostrar como o circuito é alimentado por uma fonte de 12V. A utilização dessa ferramenta facilita o planejamento, a construção e a análise do circuito, garantindo uma organização clara das conexões e permitindo uma comunicação precisa sobre o funcionamento do sistema entre todos os envolvidos no projeto.

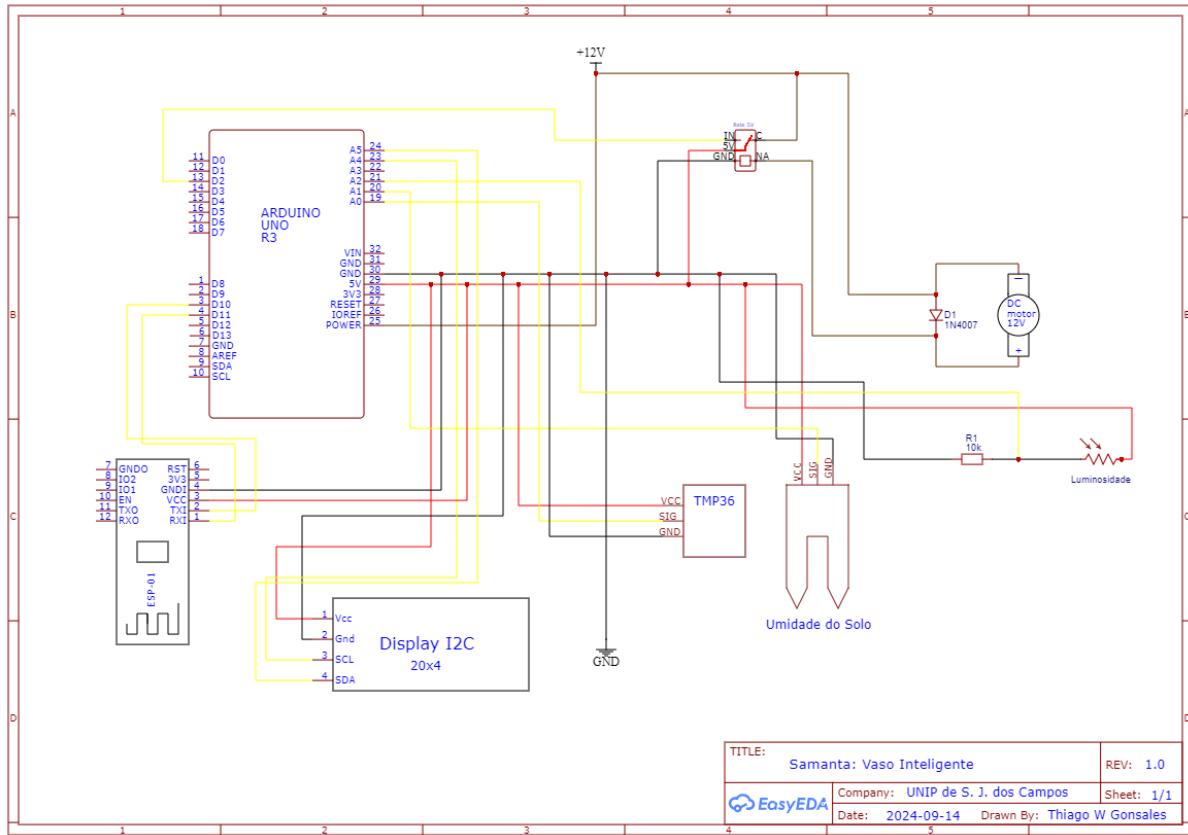


Figura 13: Esquema do protótipo do Vaso Samanta

Fonte: Autor

### 3.1.3 Fluxograma do Funcionamento do Hardware

Para ilustrar o funcionamento do hardware, foi criado um fluxograma que mostra a árvore de decisões gravadas na memória do Arduino. A Figura 14 apresenta este fluxograma, detalhando as diferentes condições e ações que o sistema pode executar com base nas entradas recebidas.

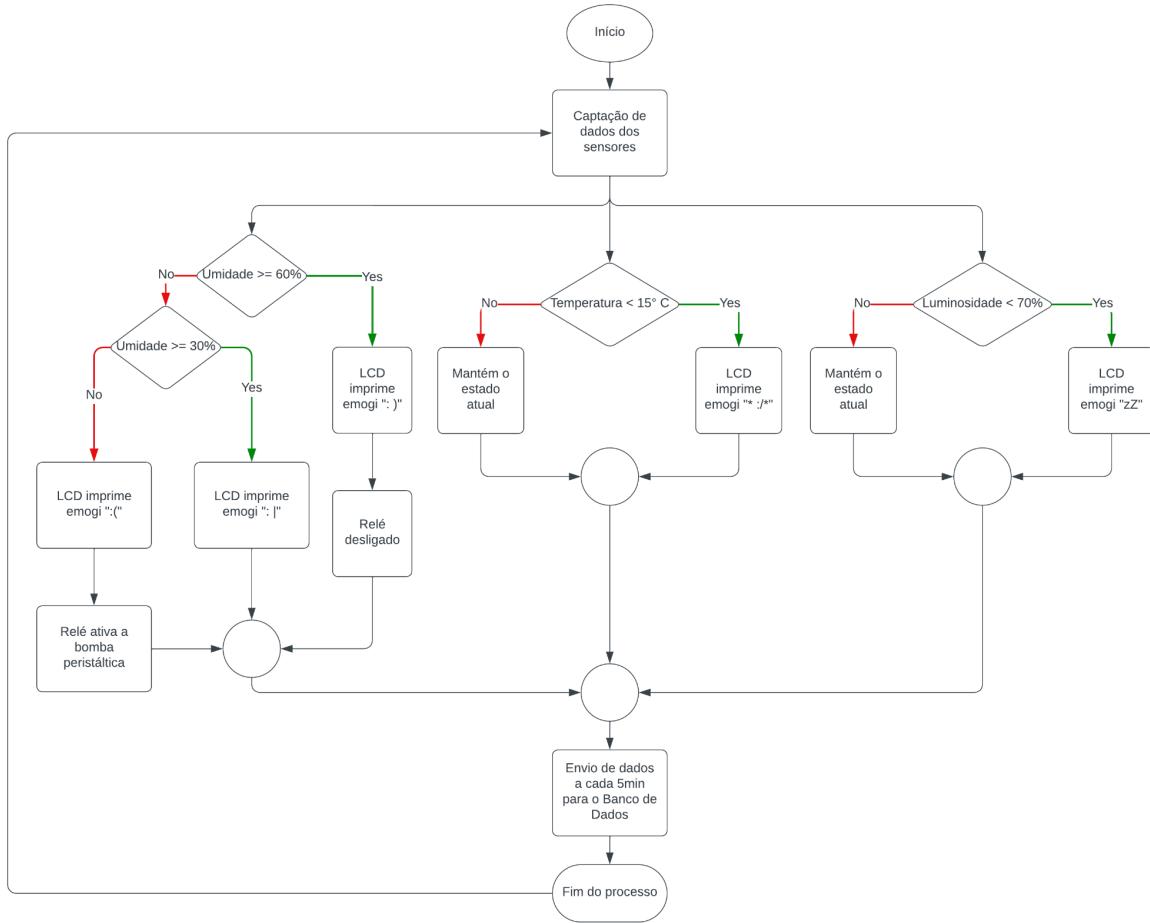


Figura 14: Fluxograma do Funcionamento do Hardware

Fonte: Autor

### 3.1.4 Cálculos

Para a execução do projeto, é importante garantir que os sensores operem dentro dos parâmetros esperados. Embora os sensores adquiridos, como o TMP36, venham com especificações de fábrica — no caso do TMP36, uma precisão de  $\pm 2^{\circ}\text{C}$  sob condições normais —, a validação em atividade é necessária para garantir a segurança e confiabilidade dos dados. Para isso, são aplicados métodos estatísticos, como o cálculo da média e do desvio-padrão, a fim de verificar se os sensores estão funcionando dentro das tolerâncias estabelecidas no estudo de caso. Essa validação assegura que os sensores atuem conforme o esperado, minimizando erros e assegurando a confiabilidade do sistema.

Para a realização do cálculo de média e desvio-padrão, serão utilizados as seguintes fórmulas:

Média Amostral:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Desvio-padrão:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$

É importante destacar que todas as amostras indicadas nos tópicos adiante foram coletadas enquanto o protótipo do Vaso Samanta estava em funcionamento em um ambiente controlado. Isso foi feito com o objetivo de minimizar a variância dos sensores e garantir a consistência das amostras.

#### 3.1.4.1 Calibragem do Sensor de Umidade de Solo

O sensor de umidade do solo utilizado no projeto opera em uma faixa de leitura que varia de 0 a 1023 unidades, quando conectado a uma entrada analógica do Arduino. Para facilitar a interpretação dos resultados e apresentar os dados em formato percentual (0% a 100%), é possível aplicar uma regra de três simples. Essa abordagem converte os valores lidos pelo sensor em porcentagem, onde 0 representa a ausência total de umidade e 1023 corresponde à saturação máxima. A equação utilizada para realizar essa conversão pode ser expressa da seguinte maneira:

$$U(\%) = \frac{S_{umid}}{S_{max}} \times 100$$

Por exemplo, se o sensor retornar o valor de 512, o cálculo será:

$$U(\%) = \frac{512}{1023} \times 100 \approx 50,05\%$$

Dessa forma, a umidade do solo é representada de maneira porcentual, o que torna os dados mais compreensíveis e facilita a análise das condições do solo e o controle do sistema.

Para avaliar a precisão do sensor, foram coletadas 40 amostras ao longo de 3 horas de funcionamento do protótipo. Com base nessas amostras, é possível calcular a média e o desvio-padrão, métricas essenciais para validar a consistência das medições realizadas pelo sensor.

As amostras coletadas foram as seguintes:

77, 77.

Dado que todas as amostras são idênticas devido ao arredondamento da tipagem da tabela, a média amostral ( $\bar{x}$ ) das leituras é de:

$$\bar{x} = \frac{2926}{40} = 77.00$$

Como todas as leituras são exatamente iguais, o desvio-padrão ( $\sigma$ ) do conjunto de dados é zero:

$$\sigma = \sqrt{\frac{(77.00 - 77.00)^2}{40-1}} = 0$$

Isso indica que, durante o período de coleta, o sensor manteve consistência nas medições.

### 3.1.4.2 Calibragem do Sensor de Temperatura

A conversão dos dados obtidos pelo sensor de temperatura conectado ao Arduino se dá em duas etapas: a conversão da leitura analógica para tensão e, em seguida, a conversão dessa tensão para temperatura em graus Celsius. As equações utilizadas são detalhadas a seguir:

#### 1. Conversão de Leitura Analógica para Tensão:

$$V_{sensor} = \left( \frac{L_{analog} \times V_{ref}}{R_{adc}} \right)$$

Essa fórmula converte a leitura analógica do sensor para um valor de tensão. Por exemplo,  $L_{analog} = 512$  e  $V_{ref} = 5V$ , a tensão medida seria:

$$V_{sensor} = \left( \frac{512 \times 5.0}{1024} \right) = 2.5V$$

#### 2. Conversão de Tensão para Temperatura em Graus Celsius:

Após a conversão da leitura para tensão, utilizamos uma segunda fórmula para converter a tensão em temperatura:

$$T_c = (V_{sensor} - V_{offset}) \times K$$

Por exemplo, se  $V_{sensor} = 2.5\text{V}$ ,  $V_{offset} = 0.5\text{V}$ , e  $K = 100$ , a temperatura em graus Celsius será:

$$T_c = (2.5V - 0.5) \times 100 = 200^\circ C$$

Para o monitoramento de um vaso de plantas, foi realizada a coleta de dados de temperatura ao longo de 3h, resultando em um total de 40 amostras. A análise das temperaturas coletadas fornece informações importantes sobre a estabilidade ambiental, que é crucial para a saúde e o crescimento das plantas.

As 40 amostras de temperatura foram:

26.17, 26.66, 26.66, 27.15, 27.15, 27.64, 25.68, 25.68, 25.68, 24.71, 28.13, 28.13, 28.61, 28.61, 28.61, 29.10, 29.59, 29.59, 29.59, 29.59, 30.57, 32.03, 23.24, 23.24, 25.68, 27.15, 27.15, 27.15, 27.15, 27.15, 27.15, 26.17, 28.13, 25.68, 28.61, 29.10

A média das temperaturas foi calculada para fornecer o valor central da temperatura ao longo do período de monitoramento. O cálculo é expressado pela equação:

$$\bar{x} = \frac{1099.47}{40} \approx 27.49$$

Calculando o desvio-padrão:

$$\sigma = \sqrt{\frac{(26.17-27.49)^2 + (26.66-27.49)^2 + (26.66-27.49)^2 + \dots + (29.10-27.49)^2}{40-1}} \approx 1.80$$

O desvio padrão de  $1.8^\circ C$  sugere que as temperaturas oscilaram cerca de  $\pm 1.8^\circ C$  em relação à média, o que reflete a estabilidade do ambiente térmico. Embora alguma variação de temperatura seja natural, monitorar o desvio-padrão permite identificar e corrigir flutuações que possam prejudicar as plantas, possibilitando ajustes para otimizar a estabilidade térmica.

### 3.1.4.3 Calibragem do Sensor de Luminosidade

Diferente dos outros sensores, o sensor de luminosidade precisa de um resistor de 10k ohms para seu funcionamento. A utilização de um resistor de 10k ohms no sensor de luminosidade está relacionada à criação de um divisor de tensão, que garante que a saída de tensão fique na faixa de 0 a 5V, a qual é compatível com a entrada analógica do Arduino. Quando dois resistores são conectados em série, a tensão aplicada ao circuito é dividida proporcionalmente com base nas resistências de cada resistor. No caso do sensor de luminosidade, o resistor de 10k ohms funciona em conjunto com a resistência variável do sensor para formar um divisor de tensão. A fórmula do divisor de tensão é:

$$V_{out} = V_{in} \times \frac{R2}{R1+R2}$$

Com essa modificação, o Arduino passa a receber sinais analógicos variando de 0 a 1023 unidades provenientes do sensor de luminosidade. Assim, é necessário converter essas unidades para uma escala de 0% a 100% para simplificar a sua utilização utilizando a seguinte fórmula:

$$P_{lum} = \left( \frac{L}{L_{max}} \right) \times 100$$

Para validar a precisão do sensor de luminosidade, foram coletadas 40 amostras durante 3 horas de análise, permitindo o cálculo da média e do desvio padrão.

Estas foram as amostras adquiridas para análise:

89.93, 90.13, 91.01, 92.08, 91.69, 91.69, 92.57, 92.18, 92.28, 93.26, 93.55, 92.38, 93.06, 92.67, 91.50, 90.71, 93.45, 93.74, 93.26, 93.84, 93.55, 92.67, 91.40, 88.95, 92.18, 90.42, 92.77, 89.54, 91.30, 91.59, 91.89, 92.16, 92.40, 92.60, 91.95, 91.77, 92.39, 92.50, 92.07, 92.33

A média amostral é obtida pela equação:

$$\bar{x} = \frac{3681,41}{40} \simeq 92,04$$

Com a média amostral, se obtém o desvio-padrão:

$$\sigma = \sqrt{\frac{(89.93-92.04)^2 + (90.13-92.04)^2 + (91.01-92.04)^2 + \dots + (92.33-92.04)^2}{40-1}} \simeq 1,14$$

Um desvio padrão de 1,14 na variação de porcentagem de luminosidade indica que os valores de luminosidade estão, em média, a 1,14 unidades percentuais de distância da média da amostra. Em outras palavras, a maior parte dos valores de luminosidade está concentrada dentro de uma faixa de aproximadamente  $\pm 1,14\%$  em torno da média. Esse valor de desvio padrão relativamente baixo sugere que os valores de luminosidade não variam muito, indicando uma baixa dispersão em relação à média. Isso significa que a luminosidade medida é bastante consistente e estável, com pequenas flutuações percentuais entre as amostras.

## 3.2 SOFTWARE

A aplicação web é um dos componentes centrais do protótipo da planta inteligente, oferecendo uma interface intuitiva e funcionalidades avançadas para o monitoramento e controle eficiente do sistema. Neste tópico, exploraremos o desenvolvimento da aplicação web em sua totalidade, incluindo a criação da interface, a implementação do frontend e backend, e a estruturação do banco de dados.

A arquitetura da aplicação foi baseada no modelo Cliente-Servidor, como mostra na figura 15, projetado para garantir uma experiência de usuário fluida e uma integração eficaz entre os diversos componentes do sistema. A interface foi desenvolvida com foco em usabilidade, proporcionando telas que facilitam a interação dos usuários com o sistema. O frontend foi projetado para oferecer uma interação visual clara e eficiente, enquanto o backend gerencia a lógica de processamento de dados e a infraestrutura da aplicação devolvendo ao cliente.

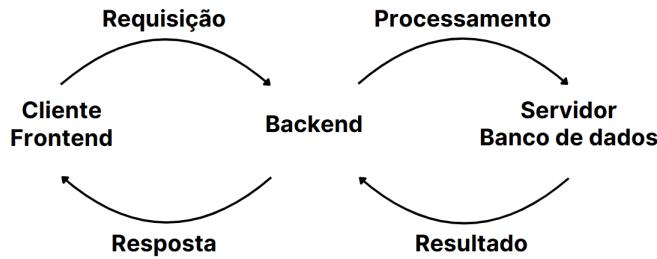


Figura 15: Modelo Cliente Servidor

Fonte: Autor

Além disso, a estrutura e organização do banco de dados foi planejada para suportar a gestão e armazenamento dos dados de forma eficaz, garantindo que a aplicação possa lidar com as necessidades de monitoramento e controle das plantas. Exploraremos também as funcionalidades implementadas para otimizar a operação do sistema e as práticas adotadas para garantir a segurança e a acessibilidade da aplicação, assegurando que o sistema seja robusto e acessível a todos os usuários.

Ao longo desta seção, apresentaremos uma visão de como cada aspecto do desenvolvimento da aplicação web contribui para a criação de uma solução completa e eficiente para o protótipo de planta inteligente.

### 3.2.1 Design da Interface

Utilizamos elementos da estética minimalista para facilitar a usabilidade do usuário, apresentando o conteúdo e os recursos do site da maneira mais simples e direta possível, para fornecer o mínimo de informação para o usuário e, ao mesmo tempo, apresentar todas as funcionalidades necessárias para o usuário.

Inicialmente, foi necessário criar um nome para o projeto. O nome “Samanta” foi dado como um trocadilho juntando as palavras “samambaia” e “planta”, gerando assim, o nome Samanta.

A partir daí, foi desenvolvido o logotipo usando a fonte Plus Jakarta San, que expressa um conceito visual, alinhando-o com a identidade planejada para ser mais amigável, além de ser legível em interfaces digitais. Em seguida, escolheu-se a fonte Inter, que foi a tipografia que melhor refletiu esse conceito no corpo do site.

A seleção das cores foi realizada levando em conta a temática relacionada às plantas, usando o sistema de cores RGB, que é o mais indicado para projetos que serão visualizados em monitores.

Além disso, a estética minimalista também foi utilizada na escolha das cores, com o objetivo de gerar uma comunicação visual coerente e atraente para o público-alvo, utilizando uma paleta de cores monocromática no layout do site, o que ajudou a limitar a quantidade de informação visual, em prol da usabilidade do usuário. Abaixo na Figura 16 está o guia de estilo que fizemos para montar a interface do site da forma mais coesa possível.



Figura 16: Guia de Estilos

Fonte: Autor

### 3.2.1.1 Funcionalidades do Sistema

Todas as telas foram organizadas para criação de um protótipo não interativo do software do projeto, para assim, ter uma noção inicial de como a interface final funcionará. Fizemos o uso do aplicativo Figma para construir este protótipo.

O fluxo de navegação do usuário, conforme indicado na figura 17, começa quando ele entra no site e aparece a tela de carregamento, indicado com o número 1. Após o carregamento, o usuário é direcionado para a tela de login (número 2), porém se a pessoa não

possuir cadastro ela poderá acessar a tela de cadastro (número 3) clicando no botão “Criar Agora” da tela de login.

Após o cadastro, o usuário será levado para a tela ilustrada no número 4, onde ele deve selecionar qual tipo de planta irá cadastrar. Após isso, a pessoa será direcionada para a tela onde ela poderá ver todas suas plantas já cadastradas, conforme ilustrado no número 5.

Ao clicar em uma das plantas cadastradas, irá aparecer a tela principal, conforme o número 6, onde mostra um mascote sorridente que reage de acordo com os dados coletados pelos sensores de umidade, luminosidade e temperatura; o nome comum e o nome científico; os dados coletados; e informações gerais daquela determinada planta escolhida.

Ao clicar no botão “Sobre”, o usuário será direcionado para a tela representada no número 7, onde tem informações sobre o projeto Samanta e sobre os desenvolvedores. Agora, se o usuário clicar no ícone de banco de dados que se encontra no canto inferior direito da tela representada no número 6, ele será levado para a tela que está representada com o número 8. Nela existem botões, que ao clicar, mostram um gráfico de acordo com a informação desejada, que podem ser os dados referentes a luminosidade, temperatura ou umidade. Além disso, o usuário pode escolher se quer ver os dados dos últimos dias, semanas ou meses, apertando os botões “D”, “S” e “M”, respectivamente.

No histórico foram utilizados dois tipos de gráficos. O gráfico de área, conforme representado na figura 18, foi escolhido para que o usuário consiga visualizar melhor as variações de temperaturas que sua planta sofreu, sendo a área escura a temperatura máxima registrada naquele período e a área clara sendo a temperatura mínima registrada. Já o gráfico de coluna, representado na figura 19, foi escolhido para representar a quantidade de umidade ou luminosidade absorvida naquele determinado período de tempo. Ambos os gráficos são meramente ilustrativos, eles não apresentam os dados coletados nos testes dos sensores.

Por fim, se acontecer algum problema no site, o usuário será levado para a tela representada com o número 9 e ao apertar o botão “Voltar”, o usuário será redirecionado para a tela que estava anteriormente.

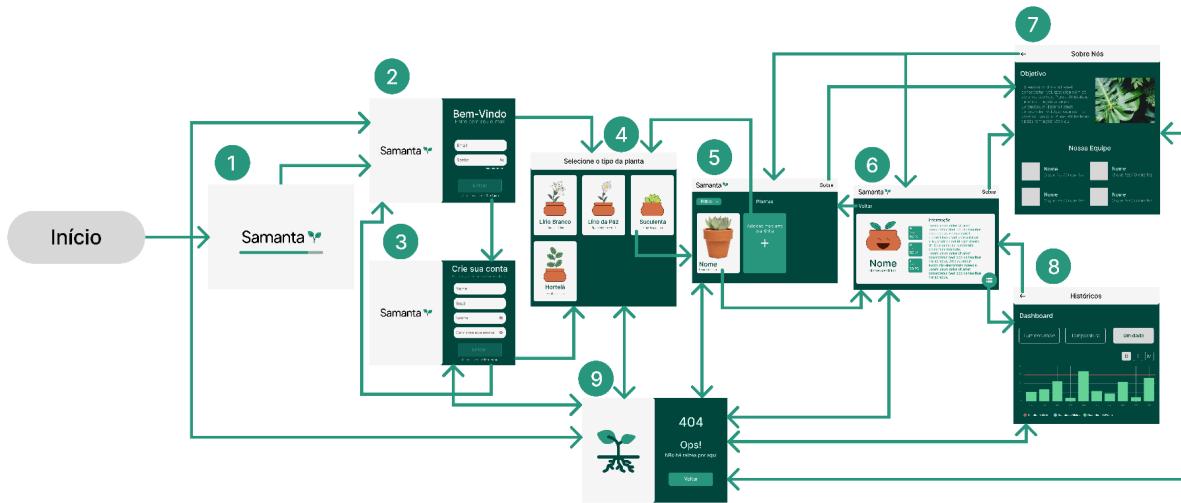


Figura 17: User flow

Fonte: Autor

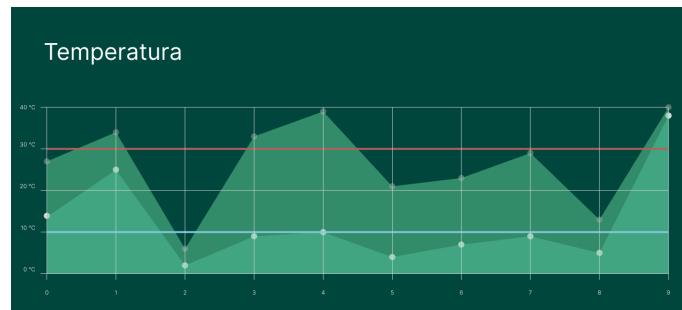


Figura 18: Gráfico de temperatura

Fonte: Autor

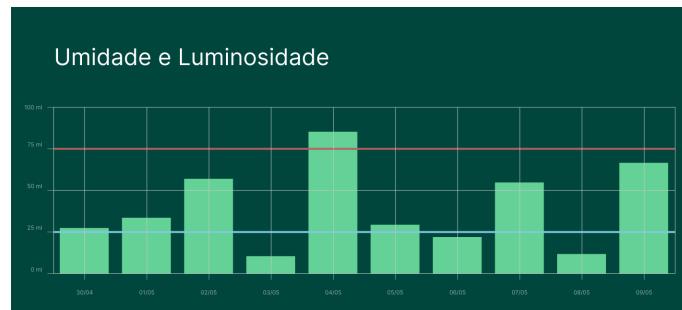


Figura 19: Gráfico de umidade e luminosidade

Fonte: Autor

### 3.2.2 Desenvolvimento da Aplicação

Nesta seção, abordaremos o processo de desenvolvimento da aplicação web voltada para o monitoramento de plantas domésticas em ambientes residenciais. A aplicação foi projetada com o objetivo de facilitar o acompanhamento de variáveis importantes, como umidade, temperatura e luminosidade, através da coleta de dados de sensores integrados a um vaso de planta.

#### 3.2.2.1 Cadastro de Usuário

A primeira funcionalidade desenvolvida na aplicação é o cadastro de usuários no sistema, que inclui a validação dos dados fornecidos e o fornecimento de feedback em caso de erro. O principal arquivo responsável por essa funcionalidade é o `Cadastro.jsx`, onde o usuário pode inserir informações como nome, login (e-mail) e senha. A lógica de envio dos dados está integrada ao backend através de uma API utilizando a biblioteca Axios.

O hook `useState` foi utilizado para gerenciar os valores dos inputs (nome, login e senha) e o estado de carregamento da página (`loading`). Ao submeter o formulário com os dados preenchidos, a função `handleButtonClick` é acionada, enviando uma requisição ao backend contendo as informações fornecidas. Esse envio é realizado pela função `cadastroUsuario`, que faz a chamada à API para registrar os dados no backend.

Quando o cadastro é concluído com sucesso, o usuário é redirecionado para a página de login, utilizando o hook `useNavigate` do React Router Dom. Contudo, caso o backend retorne um erro (por exemplo, se o login ou e-mail já estiverem cadastrados), uma mensagem de alerta é exibida para o usuário como está aparecendo na Figura 20.

```

11  const Cadastro = () => {
12    let navigate = useNavigate()
13
14    const [name, setName] = useState('')
15    const [login, setLogin] = useState('')
16    const [password, setPassword] = useState('')
17    const [loading, setLoading] = useState(false)
18    const handleSubmit = (event) => {
19      event.preventDefault();
20      console.log("Submit", { login, password });
21    }
22    const handleButton_Click = async () => {
23      setLoading(true)
24      try {
25        const resp = await cadastrarUsuario(name.toLowerCase(), login.toLowerCase(), password)
26        console.log(resp)
27        if (resp.status === 201) {
28          navigate('/');
29          setLoading(false);
30        }
31      } catch (err){
32        if(err.response && err.response.status === 401){
33          alert(` ${err.response.data.message} `)
34        }else{
35          alert('Ocorreu um erro inesperado. Por favor, tente novamente')
36          console.error("Erro inesperado: ", err)
37        }
38        setLoading(false);
39      }
40    }
41  }
42
43  export default Cadastro;

```

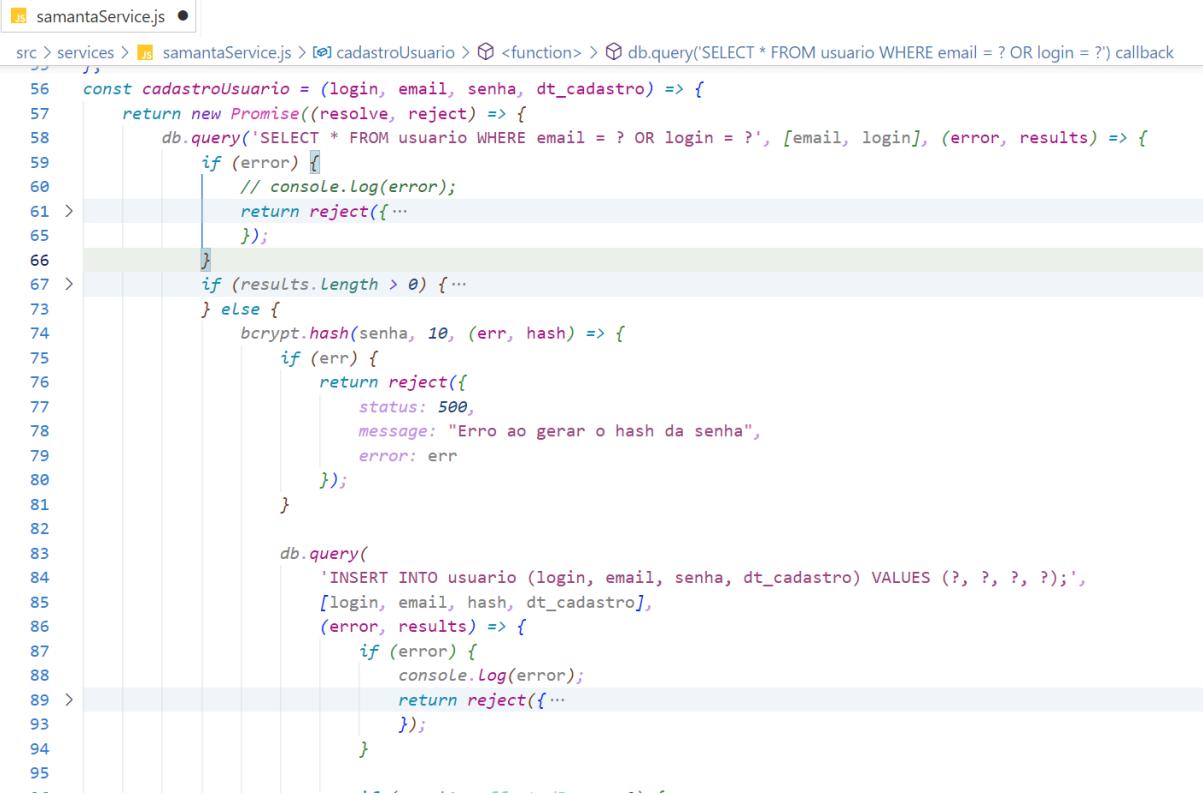
Figura 20: Cadastro.jsx

Fonte: Autor

No backend, mostrado na figura 21, a arquitetura segue um modelo baseado em rotas, controladores e serviços. O endpoint /cadastrarUsuario recebe os dados enviados pelo frontend e os direciona para o controlador responsável pelo processo de cadastro. O controlador, por sua vez, encaminha esses dados para o serviço, que realiza a lógica de validação e inserção no banco de dados. Se o cadastro for bem-sucedido, um status 201 é retornado; caso contrário, é enviado um status de erro apropriado (por exemplo, um status 401 em casos de duplicidade de e-mail ou login).

A função `cadastrarUsuario` é responsável pela interação com o banco de dados. Inicialmente, verifica-se se o login ou o e-mail já estão cadastrados. Em caso afirmativo, a função rejeita a promessa com o erro correspondente. Se o login for novo, a senha é criptografada utilizando a biblioteca `bcrypt` antes de ser inserida no banco de dados. A resposta do banco é então processada para garantir que o usuário foi criado corretamente, e uma mensagem de sucesso é retornada ao frontend.

A criptografia das senhas utilizando bcrypt assegura a proteção dos dados dos usuários. Além disso, a verificação de e-mail e login no banco de dados previne a duplicidade de contas.



```

js samantaService.js ●
src > services > samantaService.js > [o] cadastroUsuario > ⚡ <function> > db.query('SELECT * FROM usuario WHERE email = ? OR login = ?') callback
  ...
56   const cadastroUsuario = (login, email, senha, dt_cadastro) => {
57     return new Promise((resolve, reject) => {
58       db.query('SELECT * FROM usuario WHERE email = ? OR login = ?', [email, login], (error, results) => {
59         if (error) {
60           // console.log(error);
61           return reject({...
62         }
63       }
64       if (results.length > 0) {...}
65     } else {
66       bcrypt.hash(senha, 10, (err, hash) => {
67         if (err) {
68           return reject({
69             status: 500,
70             message: "Erro ao gerar o hash da senha",
71             error: err
72           });
73         }
74
75         db.query(
76           'INSERT INTO usuario (login, email, senha, dt_cadastro) VALUES (?, ?, ?, ?);',
77           [login, email, hash, dt_cadastro],
78           (error, results) => {
79             if (error) {
80               console.log(error);
81               return reject({...
82             }
83           }
84         );
85       }
86     }
87   }
88
89 >
90
91
92
93
94
95
--
```

Figura 21: samantaService.jsx

Fonte: Autor

### 3.2.2.2 Login do Usuário



```

10 <const Login = () => {
11   const buttonSx = {...}
12   ...
13   let navigate = useNavigate()
14   const [login, setLogin] = useState('')
15   const [password, setPassword] = useState('')
16   const [loading, setLoading] = useState(false)
17   const handleSubmit = (event) => { event.preventDefault() }
18   const {authLogin} = useContext(AuthContext)
19   const handleButtonClick = async () => {
20     setLoading(true)
21     await authLogin(login, password);
22     setLoading(false);
23   }
24   if (loading) { ... }
25   ...
26   return (
27     <React.Fragment>
28       <div className={style.container_main}>
29         <LogoSamanta />
30         <div className={style.forms}>
31           ...
32           <form onSubmit={handleSubmit} className={style.div_input_centralizado}>
33             <h3 className={style.titulo_acesso}>
34               Bem-Vindo!
35             </h3>
36             <div className={style.div_input}>
37               <BiUser className={style.icon} size={21} color="#00483d" />
38               <input className={style.input_Login} type="text" placeholder="E-mail" value={login} onChange={(event) => setLo
39             </div>
40           ...
41         </div>
42       </div>
43     </React.Fragment>
44   )
45 }
46
47
48
49
50

```

Figura 22: Login.jsx

Fonte: Autor

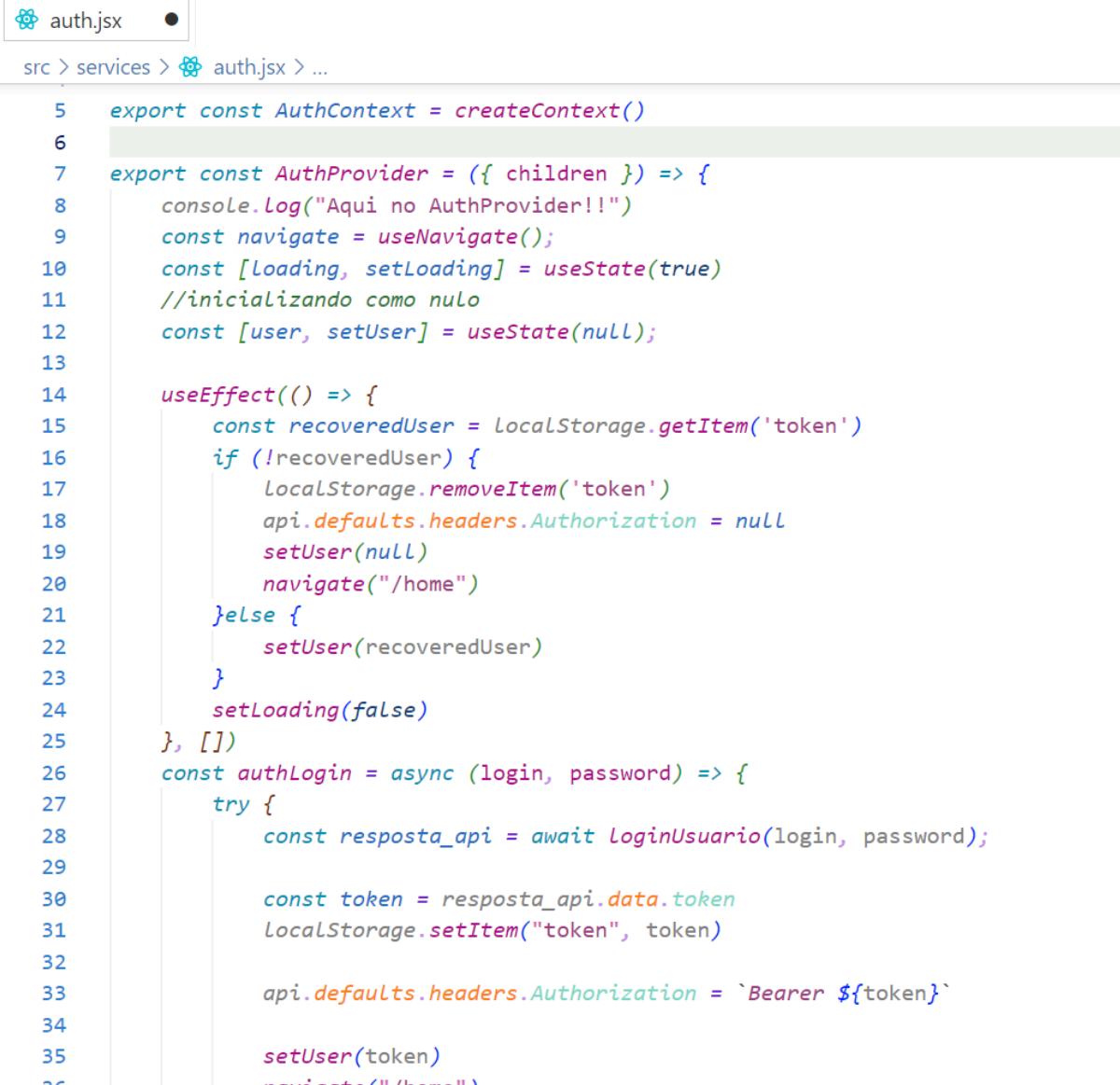
A figura 22 mostra o componente Login.jsx, que desempenha um papel fundamental na exibição do formulário de login e na interação com o backend, utilizando a Context API para gerenciar a autenticação do usuário. Ele captura as credenciais inseridas (login e senha) e, ao submeter o formulário, tenta autenticar o usuário no sistema.

No frontend, o gerenciamento do estado dos campos de login e senha é realizado por meio do hook useState, que monitora as alterações nos inputs à medida que o usuário digita. Além disso, um estado booleano, denominado loading, é utilizado para controlar o feedback visual de carregamento durante o processo de autenticação.

O formulário de login consiste em dois campos principais: e-mail e senha. O frontend também realiza uma validação básica ao capturar o evento de envio do formulário e impedir seu envio direto ao backend utilizando event.preventDefault().

Quando o usuário clica no botão de login, a função handleButtonClick é disparada, acionando a função authLogin, obtida por meio do AuthContext. Essa função envia as credenciais do usuário para o backend. Se a autenticação for bem-sucedida, o usuário é redirecionado para a página inicial (/home). Caso o backend ainda esteja processando a requisição. Se o processo de login falhar, o sistema exibe uma mensagem de erro ao usuário.

A lógica de autenticação no frontend é centralizada no arquivo Auth.jsx, exibido na figura abaixo, onde o contexto global (AuthContext) é criado para gerenciar o estado do usuário autenticado e as funções de login e logout. O estado user mantém as informações do usuário autenticado, inicialmente nulo, sendo atualizado com o token JWT retornado pelo backend após um login bem-sucedido.



```

  auth.jsx
  └── auth.jsx
    └── ...
  src > services > auth.jsx > ...

5  export const AuthContext = createContext()
6
7  export const AuthProvider = ({ children }) => {
8    console.log("Aqui noAuthProvider!!!")
9    const navigate = useNavigate();
10   const [loading, setLoading] = useState(true)
11   //inicializando como nulo
12   const [user, setUser] = useState(null);
13
14   useEffect(() => {
15     const recoveredUser = localStorage.getItem('token')
16     if (!recoveredUser) {
17       localStorage.removeItem('token')
18       api.defaults.headers.Authorization = null
19       setUser(null)
20       navigate("/home")
21     } else {
22       setUser(recoveredUser)
23     }
24     setLoading(false)
25   }, [])
26   const authLogin = async (login, password) => {
27     try {
28       const resposta_api = await LoginUsuario(login, password);
29
30       const token = resposta_api.data.token
31       localStorage.setItem("token", token)
32
33       api.defaults.headers.Authorization = `Bearer ${token}`
34
35       setUser(token)
36       navigate("/home")
37     } catch (error) {
38       console.error(error)
39     }
40   }
41 }

```

Figura 23: auth.jsx

Fonte: Autor

O estado loading gerencia o processo de verificação do token armazenado no localStorage, garantindo que, ao recarregar a página, o usuário permaneça autenticado, se

houver um token válido. O hook useEffect verifica automaticamente o localStorage e, se encontrar um token, ele é aplicado aos headers das requisições HTTP.

A função authLogin, figura 23, definida no api.js, realiza o envio das credenciais de login ao backend, por meio de uma requisição POST no endpoint /loginUsuario. Caso a autenticação seja bem-sucedida, o token JWT retornado é armazenado no localStorage, permitindo que o usuário continue autenticado em futuras sessões. Se o processo falhar, o usuário é notificado de que suas credenciais estão incorretas.

A função authLogout, figura 24, remove o token JWT do localStorage, redefine o estado user para null, e redireciona o usuário de volta à página de login, encerrando a sessão de forma segura.



```

  auth.jsx
  ●
src > services > auth.jsx > ...
7  export const AuthProvider = ({ children }) => {
26    const authLogin = async (login, password) => {
27      try {
28        const resposta_api = await LoginUsuario(login, password);
29
30        const token = resposta_api.data.token
31        LocalStorage.setItem("token", token)
32
33        api.defaults.headers.Authorization = `Bearer ${token}`
34
35        setUser(token)
36        navigate("/home")
37        return resposta_api
38      } catch (error) {
39        // console.log(error.response.status)
40        alert("Usuário ou Senha incorreto")
41      }
42    }
43
44    const authLogout = () => {
45      // console.log("Logout")
46      LocalStorage.removeItem('token')
47      api.defaults.headers.Authorization = null
48      setUser(null)
49      navigate("/")
50    }
51
52    return (
53      <AuthContext.Provider value={{ authenticated: !!user, user, loading, authLogin, authLogout }}>
54        {children}
55      </AuthContext.Provider>

```

Figura 24: auth.jsx

Fonte: Autor

A comunicação com o backend é gerenciada pelo arquivo api.js, onde a função loginUsuario envia uma requisição POST para o endpoint /loginUsuario, contendo as credenciais de e-mail e senha. O backend, processa a requisição através do controlador

samantaController.loginUsuario, que delega a lógica de autenticação ao serviço samantaService.loginUsuario.

No backend, a função loginUsuario, exibido na figura 25, realiza três etapas principais: (1) busca o usuário com o e-mail fornecido no banco de dados; (2) compara a senha fornecida com a senha criptografada armazenada utilizando o algoritmo bcrypt; e (3) se as credenciais forem válidas, gera um token JWT contendo informações como Id\_Usuario, login e email. Esse token, configurado para expirar em uma hora, é então retornado ao frontend.



```

src > services > samantaService.js > loginUsuario > <function> > db.query('SELECT * FROM usuario WHERE email = ?')
116 const loginUsuario = (email, senha) => {
117   return new Promise((resolve, reject) => {
118     db.query('SELECT * FROM usuario WHERE email = ?', [email], (error, results) => {
119       if (error) {
120         return reject({
121           status: 500,
122           message: 'Erro ao buscar usuário',
123           error: error
124         });
125       }
126       if (!results || results.length === 0) {
127         return reject({
128           status: 401,
129           message: 'Email ou senha incorretos',
130           error: null
131         });
132     })
133     bcrypt.compare(senha, results[0].senha, (err, result) => {
134       if (err) {
135         return reject({
136           status: 401,
137           message: 'Erro bcrypt',
138           error: err
139         });
140       }
141       if (result) {
142         const token = jwt.sign(
143           {
144             Id_Usuario: results[0].Id_Usuario,
145             Login: results[0].Login,
146             email: results[0].email
147           },
148           process.env.JWT_KEY,
149           {
150             expiresIn: '1h'
151           }
152         );
153         return resolve({
154           status: 201,
155           message: "Autenticação feita com sucesso",
156           token: token
157         });
158       }
159     })
160   })
161   return resolve({
162     status: 401,
163     message: 'Senha Incorreta',
164     error: error
165   });
166 }
167 )
168

```

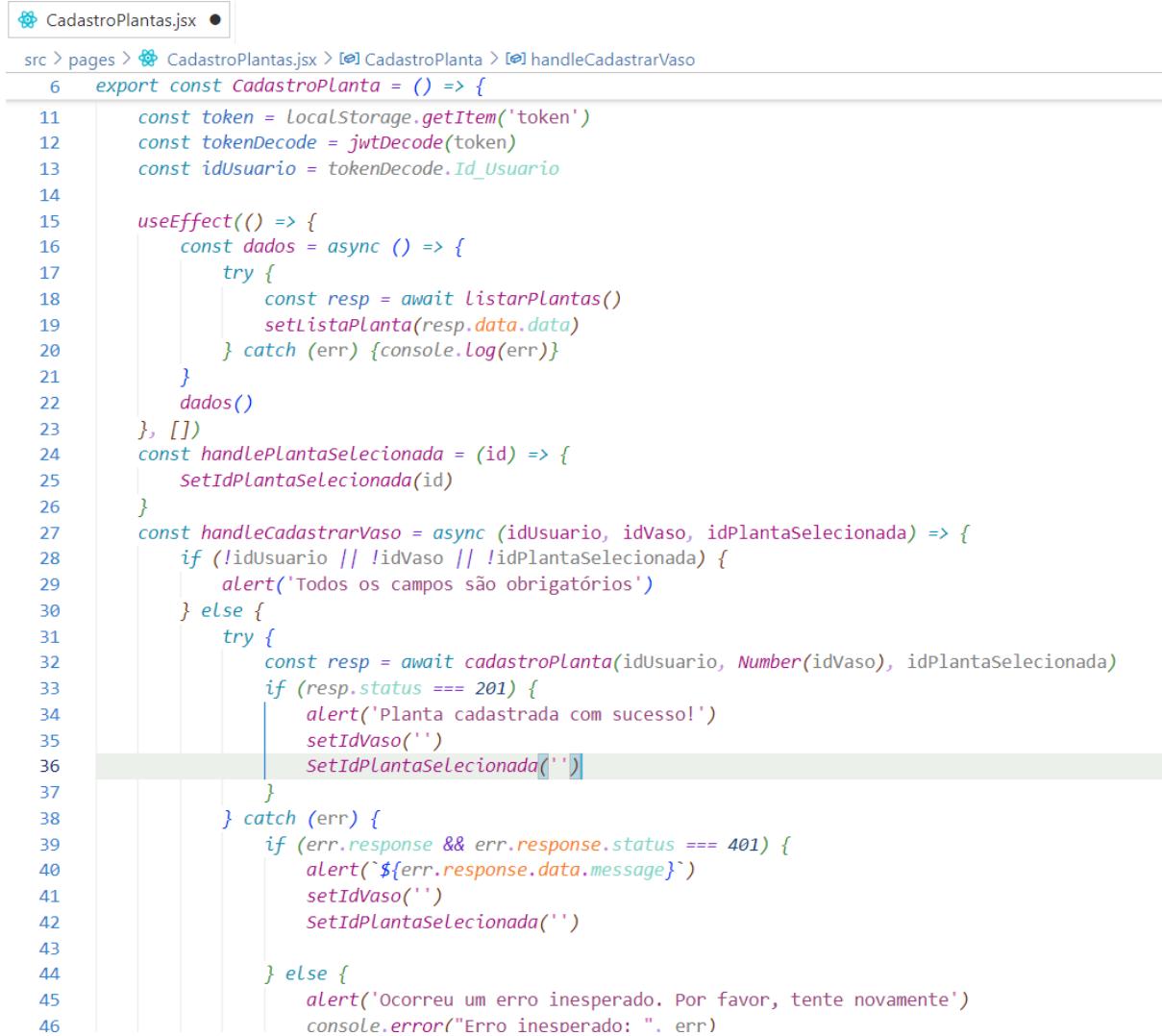
Figura 25: samantaService.js

Fonte: Autor

Se o usuário não for encontrado ou as credenciais estiverem incorretas, o backend responde com uma mensagem de erro adequada, garantindo que o frontend seja informado sobre o motivo da falha na autenticação.

### 3.2.2.3 Cadastro das Plantas

Como aparece na figura 26, o componente CadastroPlanta.jsx, é responsável por permitir que o usuário autenticado cadastre a partir de um formulário onde o mesmo pode selecionar uma planta e associá-la a um vaso.



```

  CadastroPlantas.jsx
src > pages > CadastroPlantas.jsx > CadastroPlanta > handleCadastrarVaso
6  export const CadastroPlanta = () => {
11    const token = localStorage.getItem('token')
12    const tokenDecode = jwtDecode(token)
13    const idUsuario = tokenDecode.Id_Usuario
14
15    useEffect(() => {
16      const dados = async () => {
17        try {
18          const resp = await listarPlantas()
19          setListaPlanta(resp.data.data)
20        } catch (err) {console.log(err)}
21      }
22      dados()
23    }, [])
24    const handlePlantaSelecionada = (id) => {
25      setIdPlantaSelecionada(id)
26    }
27    const handleCadastrarVaso = async (idUsuario, idVaso, idPlantaSelecionada) => {
28      if (!idUsuario || !idVaso || !idPlantaSelecionada) {
29        alert('Todos os campos são obrigatórios')
30      } else {
31        try {
32          const resp = await cadastroPlanta(idUsuario, Number(idVaso), idPlantaSelecionada)
33          if (resp.status === 201) {
34            alert('Planta cadastrada com sucesso!')
35            setIdVaso('')
36            setIdPlantaSelecionada('')
37          }
38        } catch (err) {
39          if (err.response && err.response.status === 401) {
40            alert(`${err.response.data.message}`)
41            setIdVaso('')
42            setIdPlantaSelecionada('')
43          } else {
44            alert('Ocorreu um erro inesperado. Por favor, tente novamente')
45            console.error("Erro inesperado: ". err)
46          }
47        }
48      }
49    }
50  }

```

Figura 26: CadastroPlantas.jsx

Fonte: Autor

Quando o componente é carregado, o useEffect é utilizado para fazer uma requisição à API, buscando todas as plantas disponíveis no banco de dados por meio da função listarPlantas. Esses dados são armazenados no estado listaPlanta, o nome, o nome científico e outros detalhes de cada planta, como umidade, temperatura e luminosidade. O

componente exibe essas informações ao usuário como uma lista de botões clicáveis, onde cada botão representa uma planta disponível para seleção.

O estado `idPlantaSelecionada` é atualizado sempre que o usuário clica em um desses botões, permitindo que a planta selecionada seja destacada visualmente. Isso ajuda a melhorar a experiência do usuário, deixando claro qual planta está atualmente selecionada.

Outro aspecto importante do formulário é o campo para inserção do número do vaso, controlado pelo estado `idVaso`. Esse campo coleta o número do vaso que o usuário deseja cadastrar. Tanto o campo do vaso quanto a seleção da planta são fundamentais para realizar o cadastro corretamente.

Quando o usuário clica no botão "Cadastrar Vaso", a função `handleCadastrarVaso` é executada. Ela recebe três parâmetros principais: `idUser`, `idVaso` e `idPlantaSelecionada`. Se todos os campos estiverem preenchidos, a função tenta enviar os dados para a API por meio da função `cadastroPlanta`, que faz uma requisição ao backend.

O componente `CadastroPlanta` faz uso de um serviço de API para comunicação com o backend, onde as requisições HTTP são enviadas para o endpoint `/plantasUsuario`. No backend, os dados são recebidos pelo controlador `samantaController`, que interage com o banco de dados, buscando informações de plantas associadas ao usuário e também permitindo o cadastro de novas plantas vinculadas ao vaso e ao usuário.

#### 3.2.2.4 Informações Principais da Planta

O componente `Principal.jsx` tem a função de buscar e exibir métricas relacionadas às plantas vinculadas ao usuário autenticado, como nome, umidade, temperatura, luminosidade, e as últimas atualizações das leituras e os dados ideais das condições da planta.

O estado principal da aplicação é `dadosPlantasUsuario`, que armazena as informações retornadas pela API sobre as plantas associadas ao usuário. O token JWT, armazenado no `localStorage`, é decodificado para extrair o `idUser`, o qual é utilizado nas requisições ao backend.

Dentro do `useEffect`, uma função assíncrona é implementada para realizar a requisição ao backend por meio da função `infoPlanta`, passando o `idUser` como parâmetro. A resposta da API é então armazenada no estado `dadosPlantasUsuario`. Caso ocorra algum erro durante a requisição, este é tratado de forma silenciosa, sem interferir na experiência do usuário.

A interface da página é dividida em seções, com as seguintes funcionalidades:

- Animação: Utilização do componente Lottie para exibir uma animação em formato JSON, proporcionando uma apresentação visual mais dinâmica.
- Informações sobre a Planta: Exibição do nome comum e nome científico da planta selecionada e seus parâmetros de umidade, luminosidade e temperatura. O código percorre o array dadosPlantasUsuario utilizando o método map para listar as plantas, mesmo que apenas uma seja retornada.
- Última Atualização: Apresentação da data e hora da última atualização das métricas da planta, formatada de acordo com o padrão brasileiro (dd/MM/yyyy hh:mm).

Além dessas informações, são exibidas as condições ambientais da planta:

- Umidade: Percentual de umidade do solo do vaso.
- Temperatura: Temperatura ambiente ao redor da planta, em graus Celsius.
- Luminosidade: Nível de luminosidade em torno da planta, também expresso em percentual.

No backend, a aplicação é responsável por retornar os dados das plantas associadas ao usuário por meio do endpoint /. Este endpoint realiza uma consulta no banco de dados para obter as informações do vaso, da planta e de seus parâmetros como umidade, luminosidade, temperatura que foram coletadas pelos sensores e data da última atualização. O resultado dessa consulta é enviado ao frontend, onde é processado e exibido na interface para o usuário.

### 3.2.2.5 Relatório

O componente Relatório é responsável por coletar e exibir dados ambientais de um vaso específico, apresentando-os tanto em uma tabela dinâmica quanto em um gráfico que permite a visualização dos parâmetros monitorados.

O componente utiliza o hook useLocation do React Router para capturar o id\_vaso, que identifica o vaso cujos dados estão sendo buscados. Esse id\_vaso é então passado para a função relatório, que se encarrega de consultar o backend e obter os dados.

Dentro do useEffect, uma requisição assíncrona é feita para buscar as informações do relatório desse vaso. A função relatório envia uma solicitação ao banco de dados e recupera as leituras de luminosidade, umidade, temperatura e data de atualização de cada leitura. Esses dados são armazenados no estado dadosRelatorio, que será utilizado para preencher a tabela na interface.

A biblioteca DataGrid é usada para exibir os dados em um formato tabular com as seguintes colunas:

- Data: Mostra a data e hora de cada leitura do sensor.
- Luz (%): Exibe a porcentagem de luminosidade detectada no ambiente do vaso.
- Temp. (°C): Exibe a temperatura ambiente em graus Celsius.
- Água (%): Mostra o nível de umidade no vaso em porcentagem.
- ID: Um identificador único de cada leitura.

A função relatorio interage diretamente com o banco de dados, executando uma consulta SQL que retorna as leituras de acordo com o id\_vaso fornecido. Quando a consulta é bem-sucedida, os dados são passados como uma promessa resolvida para o frontend; caso ocorra algum erro, ele é tratado adequadamente.

Esse componente oferece ao usuário uma visão clara e organizada das condições ambientais do vaso de planta. Ele integra-se com o backend para exibir as leituras em tempo real, enquanto a DataGrid permite uma visualização e manipulação eficiente dos dados. O gráfico complementa essa exibição ao fornecer uma visão mais aprofundada e dinâmica sobre a evolução dos parâmetros ao longo do tempo.

### 3.2.3 Banco de Dados

A criação e otimização do banco de dados para o protótipo do vaso foram realizadas utilizando o MySQL Workbench e o MySQL. O MySQL Workbench foi essencial para a modelagem e gerenciamento do banco de dados, oferecendo uma interface gráfica intuitiva para a criação de diagramas e a administração das tabelas. O MySQL, por sua vez, foi utilizado como o sistema de gerenciamento de banco de dados (SGBD), fornecendo a robustez necessária para a manipulação e armazenamento dos dados.

A implementação da estrutura do banco de dados envolveu a utilização da linguagem SQL, com foco em princípios de normalização para assegurar uma organização eficiente e a integridade dos dados. Essa abordagem permite um armazenamento eficaz, facilitando consultas rápidas e a manutenção da consistência dos dados.

#### 3.2.3.1 Estrutura do Banco de Dados

A definição da estrutura do banco de dados foi uma etapa crucial no desenvolvimento do sistema. Foram criadas e configuradas tabelas para armazenar e

gerenciar as informações do sistema de forma organizada e acessível. As principais tabelas estabelecidas foram:

- Usuário: Esta tabela armazena informações sobre os usuários do sistema, incluindo dados pessoais e credenciais de acesso, servindo como base para autenticação e controle de acesso.
- Usuario\_Vaso: Representa o relacionamento entre as tabelas Usuario e Vaso, permitindo associar sensores e vasos a usuários específicos. Facilita a gestão dos dados e a personalização das configurações de cada usuário, além de integrar informações sobre as plantas associadas a cada vaso.
- Vaso: Contém os dados dos vasos, incluindo a referência à planta associada. É essencial para o monitoramento e controle das condições dos vasos e das plantas.
- Planta: Armazena o catálogo de plantas disponível no aplicativo. Esta tabela inclui informações sobre as plantas cadastradas e suas preferências de cultivo, fornecendo uma base de dados para referência e cuidados específicos.
- Sensor: Esta tabela registra as medições realizadas em cada vaso, como umidade, luminosidade e temperatura. Os dados coletados pelos sensores são fundamentais para o monitoramento das condições ambientais e para a automação de cuidados com as plantas.

A estrutura dessas tabelas e seus relacionamentos foram projetados para garantir uma integração eficaz entre o sistema de hardware e a aplicação web, permitindo um fluxo contínuo de informações e uma gestão eficiente dos dados da planta inteligente e para visualizar essas estruturas, foram elaborados modelos em três níveis de abstração: conceitual, lógico e físico. Esses modelos oferecem uma visão detalhada de como os dados são organizados e manipulados:

No Modelo Conceitual definimos as principais entidades e os relacionamentos entre elas, conforme a Figura 27 proporciona uma visão geral da estrutura do banco de dados.

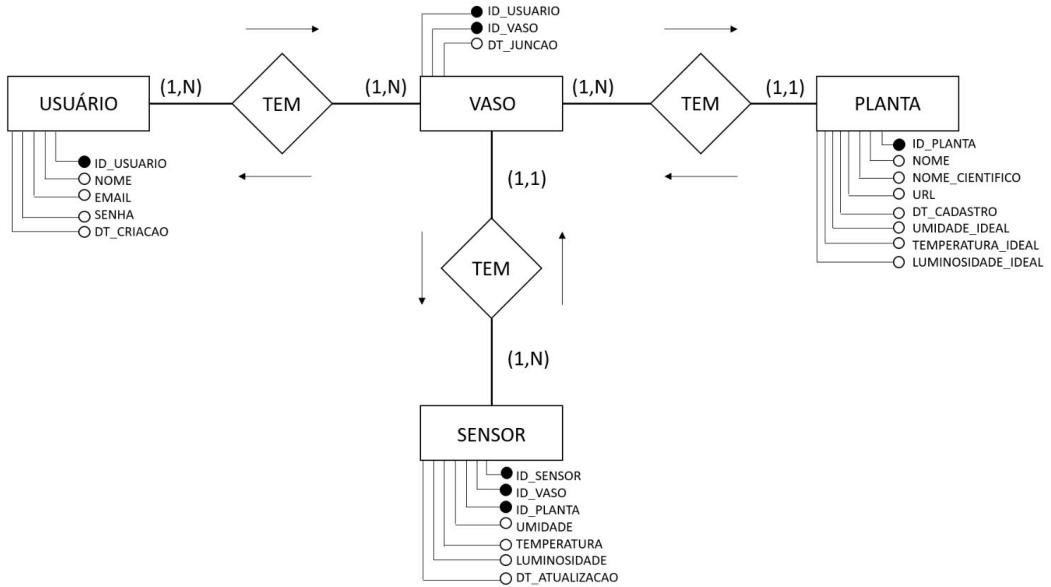


Figura 27: Modelo Conceitual

Fonte: Autor

O Modelo Lógico inclui as colunas e os relacionamentos entre elas, com ênfase na integridade referencial e na organização dos dados. Conforme a figura 28 é especificado a estrutura detalhada das tabelas.

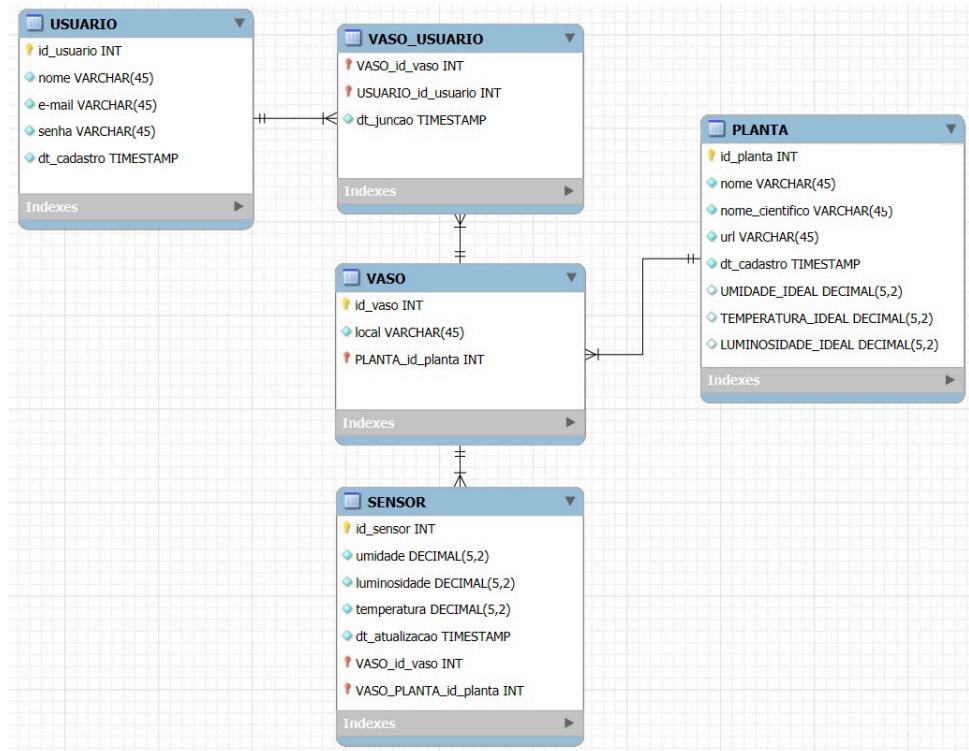


Figura 28: Modelo Lógico

Fonte: Autor

Por fim, no Modelo Físico definimos os tipos de dados, os índices para otimização e os esquemas das tabelas para assegurar um desempenho eficiente. Com isso, conforme a figura 29 transformamos em uma estrutura lógica para a implementação prática no MySQL.

```

1   -- Tabela USUARIO
2 • ○ CREATE TABLE USUARIO (
3     id_usuario INT AUTO_INCREMENT PRIMARY KEY,
4     nome VARCHAR(45) NOT NULL,
5     email VARCHAR(45) NOT NULL UNIQUE, -- E-mails devem ser únicos
6     senha VARCHAR(45) NOT NULL,
7     dt_cadastro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
8 );
9   -- Tabela PLANTA
10 • ○ CREATE TABLE PLANTA (
11    id_planta INT AUTO_INCREMENT PRIMARY KEY,
12    nome VARCHAR(45) NOT NULL,
13    nome_cientifico VARCHAR(45),
14    url VARCHAR(255), -- URL pode ser maior que 45 caracteres
15    umidade_ideal DECIMAL(5,2), -- Umidade ideal para a planta
16    temperatura_ideal DECIMAL(5,2), -- Temperatura ideal para a planta
17    luminosidade_ideal DECIMAL(5,2), -- Luminosidade ideal para a planta
18    dt_cadastro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
19 );
20   -- Tabela VASO
21 • ○ CREATE TABLE VASO (
22    id_vaso INT AUTO_INCREMENT PRIMARY KEY,
23    local VARCHAR(45),
24    PLANTA_id_planta INT,
25    FOREIGN KEY (PLANTA_id_planta) REFERENCES PLANTA(id_planta)
26    ON DELETE SET NULL -- Para evitar a exclusão de plantas caso seja excluída da tabela PLANTA
27 );
28   -- Tabela de relação VASO_USUARIO
29 • ○ CREATE TABLE VASO_USUARIO (
30    VASO_id_vaso INT,
31    USUARIO_id_usuario INT,
32    dt_juncao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
33    PRIMARY KEY (VASO_id_vaso, USUARIO_id_usuario),
34    FOREIGN KEY (VASO_id_vaso) REFERENCES VASO(id_vaso)
35    ON DELETE CASCADE, -- Ao deletar o vaso, remove a relação
36    FOREIGN KEY (USUARIO_id_usuario) REFERENCES USUARIO(id_usuario)
37    ON DELETE CASCADE -- Ao deletar o usuário, remove a relação
38 );
39   -- Tabela SENSOR
40 • ○ CREATE TABLE SENSOR (
41    id_sensor INT AUTO_INCREMENT PRIMARY KEY,
42    umidade DECIMAL(5,2),
43    luminosidade DECIMAL(5,2),
44    temperatura DECIMAL(5,2),
45    dt_atualizacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
46    VASO_id_vaso INT,
47    FOREIGN KEY (VASO_id_vaso) REFERENCES VASO(id_vaso)
48    ON DELETE CASCADE -- Ao deletar o vaso, remove os sensores
49 );

```

Figura 29: Modelo Físico

Fonte: Autor

### 3.2.3.2 Otimização

Para melhorar o desempenho das consultas, foram criados índices nas colunas frequentemente utilizadas em operações de busca e filtragem, conforme mostrado na Figura. Esses índices incluem o idx\_usuario\_email na tabela USUARIO, que otimiza o processo de autenticação; o idx\_planta\_nome na tabela PLANTA, que acelera a busca por nomes de plantas; o idx\_vaso\_id\_planta na tabela VASO, que aprimora as junções com a tabela PLANTA; o idx\_sensor\_id\_vaso na tabela SENSOR, que facilita o acesso aos dados dos sensores; e os índices idx\_usuario\_vaso\_id\_usuario e idx\_usuario\_vaso\_id\_vaso na tabela VASO\_USUARIO, que agilizam as relações entre usuários e vasos. Esses índices foram implementados para garantir operações de consulta mais eficientes, contribuindo para a agilidade geral do sistema.

```
-- Índices para otimizar o desempenho
CREATE INDEX idx_usuario_email ON USUARIO(email); -- índice no email para otimizar autenticação
CREATE INDEX idx_planta_nome ON PLANTA(nome); -- índice para busca por nome da planta
CREATE INDEX idx_vaso_id_planta ON VASO(id_planta); -- índice para junção com a tabela Planta
CREATE INDEX idx_sensor_id_vaso ON SENSOR(id_vaso); -- índice para junção com a tabela Vaso
CREATE INDEX idx_usuario_vaso_id_usuario ON VASO_USUARIO(id_usuario); -- índice para junção com Usuário
CREATE INDEX idx_usuario_vaso_id_vaso ON VASO_USUARIO(id_vaso); -- índice para junção com Vaso
```

Figura 30: Índices

Fonte: Autor

## 4. TESTES E RESULTADOS

Neste capítulo, apresentamos a metodologia de teste utilizada para avaliar o sistema desenvolvido no projeto Samanta, bem como os resultados obtidos através desses testes. A fase de testes é essencial para garantir que o sistema funcione conforme o esperado e atenda aos requisitos estabelecidos, além de identificar áreas que possam necessitar de ajustes ou melhorias.

### 4.1 Testes Funcionais

Os testes funcionais do Vaso Samanta validaram o desempenho integrado dos sensores de umidade do solo, temperatura e luminosidade, essenciais para o monitoramento e cultivo saudável das plantas. Com base no estudo de caso do tomate-cereja, foram estabelecidos parâmetros ideais para o cultivo, que serviram como referência para avaliar o desempenho do sistema.

Após calibragem cuidadosa, como destacado na seção de Cálculo, os sensores forneceram medições consistentes e precisas. Os testes também avaliaram a aplicação web, garantindo informações detalhadas e dados em tempo real.

#### 4.1.1 Hardware

Os testes de hardware do protótipo Vaso Samanta foram realizados durante um período diurno de três horas. O sistema operou de forma contínua, coletando amostras de temperatura, luminosidade e umidade do solo para validar o desempenho e a precisão dos sensores, além de verificar o funcionamento dos componentes atuadores.

##### 4.1.1.1 Sensor de Temperatura

As leituras de temperatura coletadas ao longo do teste foram:

26.17, 26.66, 26.66, 27.15, 27.15, 27.15, 27.64, 25.68, 25.68, 25.68, 24.71, 28.13, 28.13, 28.13, 28.61, 28.61, 28.61, 28.61, 29.10, 29.59, 29.59, 29.59, 29.59, 30.57, 32.03, 23.24, 23.24, 25.68, 27.15, 27.15, 27.15, 27.15, 27.15, 27.15, 26.17, 28.13, 25.68, 28.61 e 29.10.

A média de temperatura foi de 27.49°C, valor que se encontra dentro da faixa ideal de 15°C a 30°C para o crescimento do pé de tomate-cereja. Esse resultado indica que o

ambiente térmico mantido pelo sistema está adequado e que os sensores de temperatura foram capazes de monitorar as condições sem desvios significativos.

#### 4.1.1.2 Sensor de Luminosidade

As leituras de luminosidade registradas foram as seguintes:

89.93, 90.13, 91.01, 92.08, 91.69, 91.69, 92.57, 92.18, 92.28, 93.26, 93.55, 92.38, 93.06, 92.67, 91.50, 90.71, 93.45, 93.74, 93.26, 93.84, 93.55, 92.67, 91.40, 88.95, 92.18, 90.42, 92.77, 89.54, 91.30, 91.59, 91.89, 92.16, 92.40, 92.60, 91.95, 91.77, 92.39, 92.50, 92.07 e 92.33.

Esse nível de luminosidade é adequado para a fotossíntese e crescimento das plantas, garantindo que o sistema está proporcionando uma exposição adequada à luz durante o período diurno. A média de luminosidade obtida foi de 92.02%, valor que excede a faixa mínima de 70% necessária para o desenvolvimento do pé de tomate-cereja.

#### 4.1.1.3 Sensor de Umidade de Solo

As leituras de umidade do solo foram registradas em 77% para todas as amostras:

O valor de 77% está acima da faixa mínima de 20% a 60% para a umidade do solo, o que é benéfico para a saúde da planta. Isso indica que o sistema de irrigação está operando de forma eficaz, mantendo o solo consistentemente úmido e garantindo o crescimento do pé de tomate-cereja.

#### 4.1.1.4 Funcionamento dos Componentes

O Display LCD 20x4 I2C operou de maneira clara e estável ao longo dos testes, enquanto a Bomba Peristáltica respondeu de forma precisa, ativando-se conforme as leituras de umidade do solo. As leituras de temperatura, luminosidade e umidade foram transmitidas pelo módulo ESP01 para o servidor em intervalos regulares, e os dados foram armazenados de forma confiável, garantindo a integridade das informações para o monitoramento contínuo.

Os resultados dos testes indicam que o protótipo Vaso Samanta está devidamente calibrado e funcional, proporcionando condições ideais para o cultivo de plantas.

#### 4.1.2 Software

As primeiras telas da aplicação permitem que o usuário crie uma conta preenchendo um formulário com nome, e-mail e senha. Após submeter os dados, o sistema realiza a validação, assegurando que o e-mail seja único e que a senha atenda aos requisitos de segurança. A Figura 31 mostra a interface de cadastro, onde os usuários podem se cadastrar.

Após o cadastro, o usuário pode acessar a página de login, figura 32, onde deve fornecer suas credenciais (e-mail e senha). Quando os dados estão corretos, um token JWT é gerado e retornado ao cliente para ser utilizado nas próximas requisições, permitindo a autorização na aplicação.

O fluxo de cadastro e login funciona conforme esperado, proporcionando uma experiência de usuário consistente. A autenticação baseada em token foi validada, garantindo que usuários autenticados possam acessar funcionalidades. Em caso de falhas, como dados incorretos ou tentativas de cadastro com e-mail duplicado, o sistema retorna mensagens de erro.

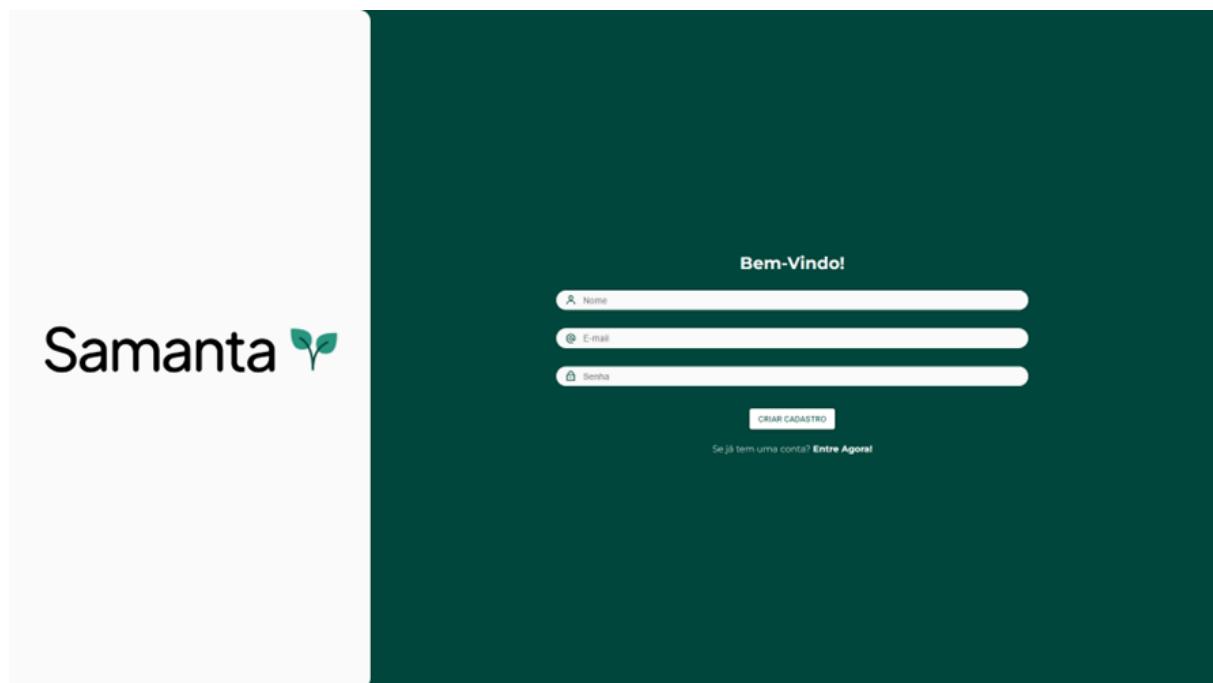


Figura 31: Tela de cadastro do usuário na aplicação.

Fonte: Autor

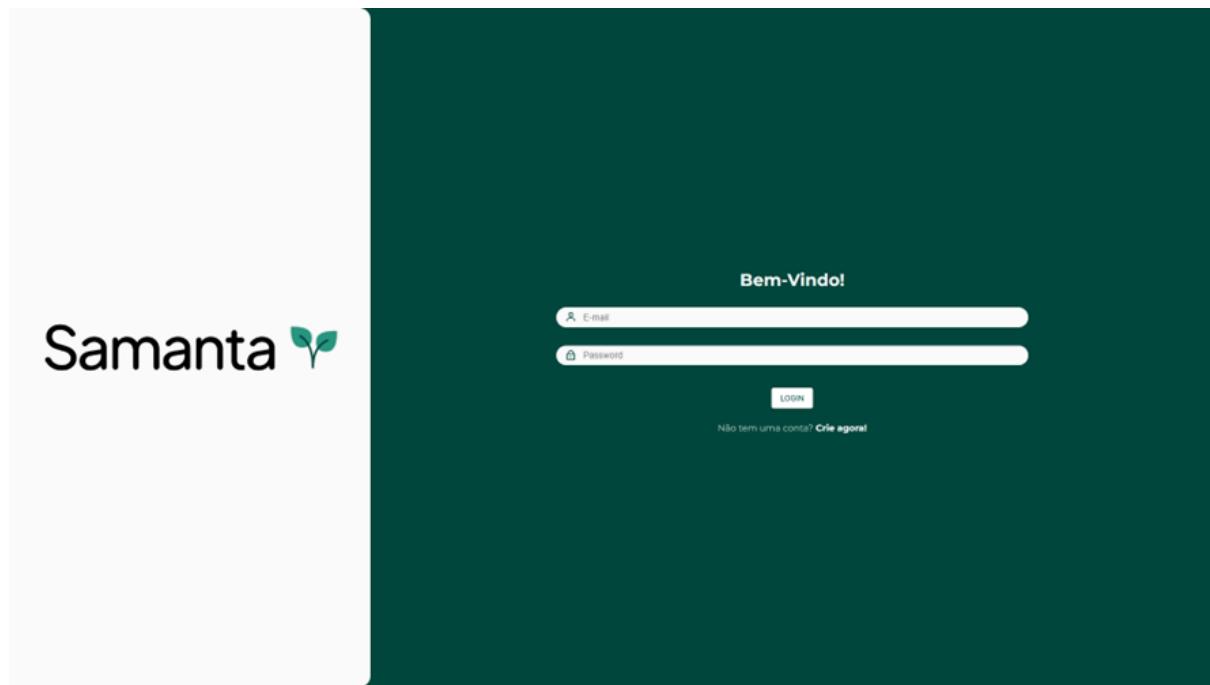


Figura 32: Tela de login do usuário na aplicação.

Fonte: Autor

Após o login bem-sucedido, o usuário será redirecionado para a tela inicial. Para o primeiro acesso, terá que cadastrar seu vaso. Ao clicar no botão “Adicione mais uma plantinha” (Figura 33), o usuário é direcionado para a página de cadastro (Figura 34).



Figura 33: Tela de home da aplicação.

Fonte: Autor

O usuário deve inserir o código do vaso, que é fornecido fisicamente no vaso, e escolher a planta que deseja monitorar. Ao clicar no botão "Cadastrar Vaso" (Figura 34), as informações são enviadas e armazenadas no banco de dados.

A aplicação garante que o código seja inserido corretamente e ele seja validado. E tem uma lista de plantas que está sendo disponibilizada pela aplicação. (Figura 35).

Ao retornar para a tela inicial será mostrado todos os vasos cadastrados ao seu respectivo usuário.

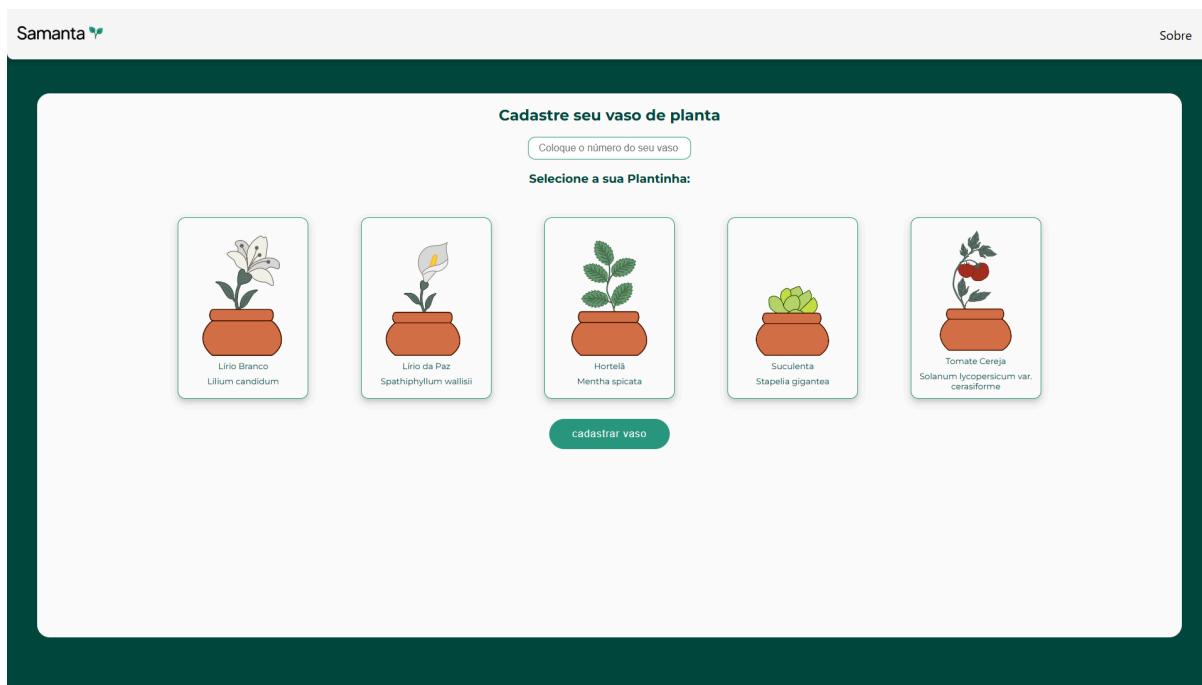


Figura 34: Tela de cadastro do vaso.

Fonte: Autor

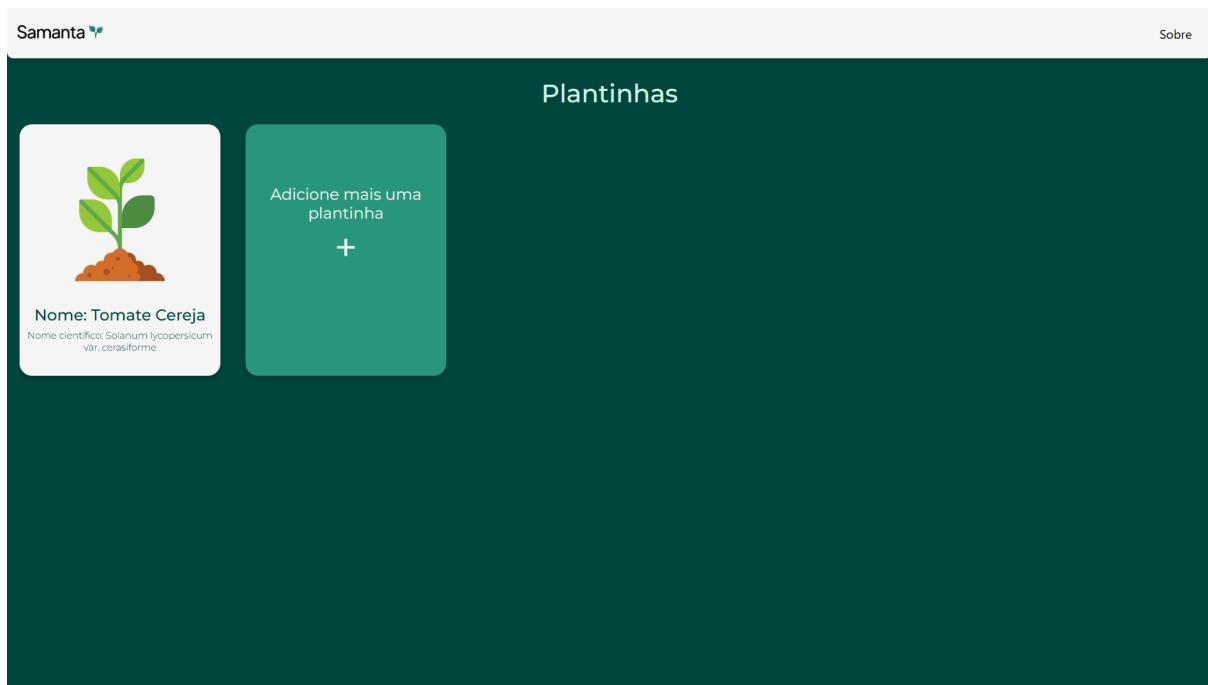


Figura 35: Tela de home da aplicação com os vasos mapeados.

Fonte: Autor

Ao clicar em um vaso específico na tela inicial, o usuário é redirecionado para a página de monitoramento daquela planta. Optamos por exibir nesta página os dados mais recentes coletados pelo hardware, como umidade, temperatura e luminosidade, tratados e formatados para fácil leitura na aplicação. Essa página, (Figura 36), permite ao usuário visualizar rapidamente as condições atuais do vaso monitorado e tem acesso aos parâmetros ideais que foram coletados de acordo com (AUTORES). Além de mostrar os parâmetros de umidade, temperatura e luminosidade ideais para aquela planta, no caso tomate-cereja.

Os dados mais recentes estão exibidos corretamente e são atualizados em tempo real conforme as novas coletas, e além de exibir a animação do estado da planta.

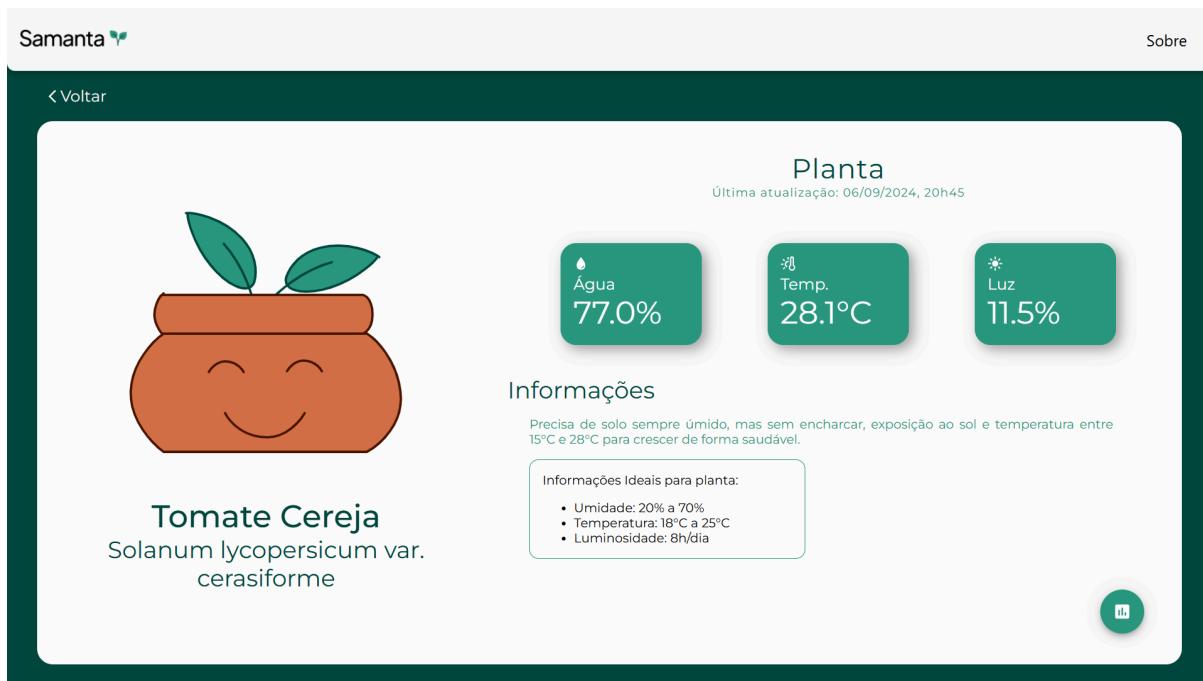


Figura 36: Tela de especificações da planta.

Fonte: Autor

E como parte final da aplicação web, o sistema gera um relatório completo com todos os dados coletados pelo hardware, permitindo que o usuário visualize as condições monitoradas e as variações. Este relatório também segue constantemente sendo atualizado conforme as coletas de dados (Figura 37).

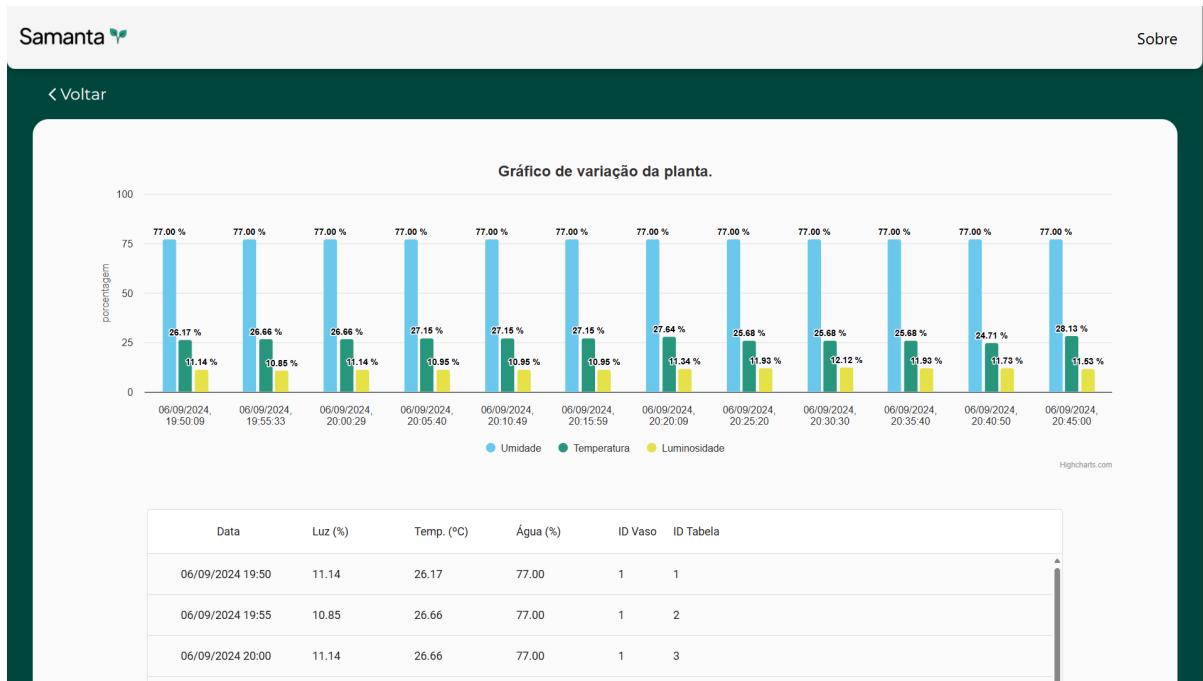


Figura 37: Tela de relatório da aplicação

Fonte: Autor

Os resultados da aplicação web seguem um fluxo de uso do sistema, desde o cadastro e login até o monitoramento e o relatório, funcionando conforme o esperado. A exibição dos dados proporciona uma experiência clara para os usuários das plantas monitoradas. A navegação das telas é fluida, com validações adequadas nas etapas para garantir que os dados serão inseridos e recebidos.

## 4.2 Resultados Obtidos

Os resultados dos testes confirmaram que o sistema Samanta atende aos objetivos propostos. A precisão dos sensores foi validada conforme os parâmetros estabelecidos para o tomate-cereja, e a interface web demonstrou ser eficaz na apresentação dos dados. O sistema conseguiu manter um bom desempenho sob diversas condições e garantiu a segurança dos dados dos usuários. A análise dos resultados indica que o sistema está pronto para ser utilizado conforme o planejado e que as funcionalidades atendem às necessidades dos usuários.

## 4.3 Conclusão dos Testes

Os testes realizados confirmaram que o sistema Samanta é eficaz na monitorização e no cuidado das plantas de forma automatizada. As funcionalidades principais estão operando conforme o esperado, e o sistema demonstrou ser robusto e seguro. As áreas de melhoria identificadas foram abordadas, e o sistema está em funcionamento para oferecer uma solução eficiente e confiável para o cultivo doméstico de tomates-cereja.

## 5. CONSIDERAÇÕES FINAIS

O projeto Samanta foi desenvolvido com o objetivo de criar uma solução tecnológica para o cultivo doméstico de plantas, integrando diversas tecnologias modernas para oferecer um sistema inteligente e eficiente. Ao longo do desenvolvimento, foram alcançados os objetivos propostos, e o projeto demonstrou o potencial das tecnologias aplicadas para resolver problemas práticos na jardinagem.

### 5.1 Avaliação dos Objetivos Específicos

Neste segmento, avaliaremos como os objetivos específicos do projeto Samanta foram alcançados, analisando cada um com base em critérios e métricas definidos no estudo de caso. A avaliação considera a eficácia e a realização de cada meta proposta para medir o sucesso e identificar áreas de melhoria.

#### 5.1.1 Integração de Sensores

O primeiro objetivo específico do projeto foi a integração de sensores de umidade, temperatura e luminosidade ao sistema. Este objetivo foi plenamente alcançado, com a implementação de sensores que permitem a coleta contínua e precisa dos dados ambientais essenciais para o cuidado das plantas. A integração desses sensores possibilitou a automação dos cuidados com as plantas, reduzindo a necessidade de intervenção manual e melhorando a eficiência do sistema.

#### 5.1.2 Implementação de Banco de Dados

O segundo objetivo específico envolveu a implementação de um banco de dados para o armazenamento dos dados coletados. O banco de dados MySQL foi configurado e otimizado para garantir a eficiência e segurança no armazenamento das informações. A normalização e as técnicas de otimização aplicadas asseguraram a integridade e a rápida recuperação dos dados, facilitando a análise e o monitoramento das condições das plantas ao longo do tempo.

### 5.1.3 Desenvolvimento de Interface Web

A criação de uma interface web intuitiva foi o foco do terceiro objetivo específico. Utilizando ReactJS, foi desenvolvida uma interface que permite aos usuários visualizar e interagir com os dados das plantas de forma prática e acessível. A interface facilita o monitoramento remoto e a análise dos dados coletados, melhorando a experiência do usuário e a usabilidade do sistema.

### 5.1.4 Criação de Paineis Interativos

O quarto objetivo específico foi a criação de painéis interativos que exibem gráficos de umidade, temperatura e luminosidade. Estes painéis, desenvolvidos com ferramentas de visualização de dados, proporcionam uma representação visual clara das condições ambientais. A capacidade de visualizar dados em tempo real e de forma gráfica permite uma análise mais eficaz e uma tomada de decisão mais informada sobre o cuidado das plantas.

### 5.1.5 Implementação de Segurança de Dados

O quinto objetivo específico envolveu a implementação de medidas de segurança para proteger os dados dos usuários. Foram adotadas práticas como autenticação JWT e criptografia de senhas com Bcrypt para garantir a segurança e a privacidade das informações. Essas medidas protegem os dados contra acessos não autorizados e asseguram a integridade das informações armazenadas.

## 5.2 Contribuições e Impactos

O projeto Samanta contribui significativamente para a área de jardinagem doméstica ao introduzir um sistema inteligente e automatizado que simplifica o cuidado com as plantas. As tecnologias aplicadas não só melhoraram a eficiência do monitoramento e do cuidado das plantas, mas também promoveram práticas mais sustentáveis e um estilo de vida mais ecológico.

## 5.3 Trabalhos Futuros

Para futuras melhorias, recomenda-se a exploração de funcionalidades adicionais, como a integração com assistentes virtuais e a expansão das capacidades de análise de dados. A adição de novas características pode aprimorar ainda mais a usabilidade e a eficácia do

sistema, oferecendo aos usuários uma ferramenta ainda mais completa e versátil para a jardinagem doméstica.

Em resumo, o projeto Samanta não apenas atingiu os objetivos propostos, mas também estabeleceu um novo modelo para a integração de tecnologias no cultivo de plantas, promovendo práticas sustentáveis, melhorando a experiência dos usuários e fornecendo uma base sólida para futuros projetos na área.

## REFERÊNCIAS

ALURA. O que é front-end e back-end. Disponível em: [https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end?srsltid=AfmBOoq0loiqfIMtxTPqaA0rgSwwGTMUktdR5Hf6n2vdyQ\\_sAGV2zIkS](https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end?srsltid=AfmBOoq0loiqfIMtxTPqaA0rgSwwGTMUktdR5Hf6n2vdyQ_sAGV2zIkS). Acesso em: 14 set. 2024.

ARDUINO. Arduino UNO. Disponível em: <https://www.arduino.cc/en/Guide/ArduinoUNO>. Acesso em: 17 set. 2024.

AUTOCORE ROBÓTICA. Display LCD 20x4 com Adaptador I2C e Backlight Azul. Disponível em: <https://www.autocorerobotica.com.br/display-lcd-20x4-com-adaptador-i2c-backlight-azul>. Acesso em: 3 set. 2024.

AUTOCORE ROBÓTICA. Sensor de Temperatura TMP36. Disponível em: <https://www.autocorerobotica.com.br/sensor-de-temperatura-tmp36>. Acesso em: 3 set. 2024.

AWS. The difference between front-end and back-end. Disponível em: <https://aws.amazon.com/pt/compare/the-difference-between-frontend-and-backend/>. Acesso em: 14 set. 2024.

AWARI. Construindo uma arquitetura de backend com Node.js: guia completo. 23 ago. 2023. Disponível em: <https://awari.com.br/construindo-uma-arquitetura-de-backend-com-node-js-guia-completo/>. Acesso em: 16 set. 2024.

AXIOS. Introdução ao Axios. Disponível em: <https://axios-http.com/ptbr/docs/intro>. Acesso em: 31 ago. 2024.

BALBINO DA COSTA, F.; GOMES, G. do N. V.; STRAUSS, E.; VILLAS BÔAS JÚNIOR, M. Automação residencial com IoT para idosos e pessoas com deficiência física. Projectus, Rio de Janeiro, v. 6, n. 4, p. 16-31, 2021.

BANDAI. Tamagotchi Original. Disponível em:  
[https://tamagotchi-official.com/br/series/original/item/01\\_265/](https://tamagotchi-official.com/br/series/original/item/01_265/). Acesso em: 14 set. 2024.

BAYER, Fernando Mariano; ECKHARDT, Moacir; MACHADO, Renato. Automação de sistemas. Santa Maria - RS: Colégio Técnico Industrial de Santa Maria, 2011.

BRASIL. Secretaria de Estado de Meio Ambiente, Desenvolvimento Econômico e Agricultura (SEMADE) de Mato Grosso do Sul. 7 dicas para ter sucesso na plantação de tomate cereja. Disponível em:  
<https://www.semadesr.ms.gov.br/7-dicas-para-ter-sucesso-na-plantacao-de-tomate-cereja/>. Acesso em: 18 de set. de 2024.

CLICK AND GROW. The Smart Garden 3. Disponível em:  
<https://www.clickandgrow.com/products/the-smart-garden-3?srsltid=AfmBOopnOJhEyMu3r4u-DV3HeIbOBPaoqS8UUfSwVLdOmasXt6LpQikf>. Acesso em: 28 fev. 2023.

CURTO CIRCUITO. Sensor de Luminosidade LDR 5mm. Disponível em:  
<https://curtocircuito.com.br/sensor-de-luminosidade-ldr-5mm.html>. Acesso em: 3 set. 2024.

DESIGN MINIMALISTA: o que é e seus tipos. Blog Adobe Brasil, 26 de mai. de 2022. Disponível em:  
<https://blog.adobe.com/br/publish/2022/05/26/design-minimalista-tudo-o-que-voce-precisa-saber-para-construir-o-seu>. Acesso em: 18 ago. 2024.

DESIGNERVERSO. O que é design? a melhor definição de todas!. YouTube, 8 de mar. de 2021. Disponível em: [https://youtu.be/3LnXgbXVg7Y?si=cb5\\_nlaekOQPTo6W](https://youtu.be/3LnXgbXVg7Y?si=cb5_nlaekOQPTo6W). Acesso em: 18 de set. de 2024.

EOS. Umidade do solo. Disponível em:  
<https://eos.com/pt/blog/umidade-do-solo/#:~:text=A%20gama%20ideal%20de%20teor,entre%2020%25%20e%2060%25>. Acesso em: 18 de set. de 2024.

FIGMA. Figma. Sobre nós. Disponível em: <https://www.figma.com/pt-br/about/>. Acesso em: 17 de set. de 2024.

GOOGLE CLOUD. O que é o MySQL?. Disponível em: <https://cloud.google.com/mysql?hl=pt-BR>. Acesso em: 1 set. 2024.

HEUSER, Carlos Alberto. Projeto de Banco de Dados V4. Porto Alegre: UFRGS, 2009. Disponível em: <https://integrada.minhabiblioteca.com.br>. Acesso em: 1 set. 2024.

IBM. Indexação de Banco de Dados. Última atualização: 3 mar. 2021. Disponível em: <https://www.ibm.com/docs/pt-br/mfc1/7.6.2?topic=databases-database-indexing>. Acesso em: 1 set. 2024.

INOVADORTECH. Tecnologia IoT: Qual é o impacto em Jardins Verticais? Disponível em: <https://inovadortech.com/impacto-da-tecnologia-iot-em-jardins-verticais/> Acesso em: 20 de set. 2024.

KERLYN. Uma breve introdução sobre Bcrypt. Medium, 10 dez. 2019. Disponível em: <https://medium.com/reprogramabr/uma-breve-introdu%C3%A7%C3%A3o-sobre-bcrypt-f2fad91a7420>. Acesso em: 10 set. 2024.

LIMA, Cleyson. O que é JWT. TreinaWeb, 2021. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-jwt>. Acesso em: 3 set. 2024.

MATTENE, Henrique. O que são sensores e quais as suas aplicações. Mundo da Elétrica. Disponível em: <https://www.mundodaeletrica.com.br/o-que-sao-sensores-e-quais-as-suas-aplicacoes/>. Acesso em: 15 set. 2024.

MDN WEB DOCS. Introdução ao Express/Node.js. Disponível em: [https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express\\_Nodejs/Introduction](https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs/Introduction). Acesso em: 2 set. 2024.

MERCADO LIVRE. Imagem de módulo sensor de umidade de solo. Bazar Bom, 2024. Disponível em: <https://produto.mercadolivre.com.br/MLB-1259941546-modulo-sensor-de-umidade-de-solo->

higrmetro-arduino-pic-rasp-\_JM?matt\_tool=40343894&matt\_word=&matt\_source=google&matt\_campaign\_id=14303413655&matt\_ad\_group\_id=133855953276&matt\_match\_type=&matt\_network=g&matt\_device=c&matt\_creative=584156655519&matt\_keyword=&matt\_ad\_position=&matt\_ad\_type=pla&matt\_merchant\_id=119740810&matt\_product\_id=MLB1259941546&matt\_product\_partition\_id=2268053647590&matt\_target\_id=pla-2268053647590&cq\_src=google\_ads&cq\_cmp=14303413655&cq\_net=g&cq\_plt=gp&cq\_med=pla&gad\_source=1&gclid=CjwKCAiAxKy5BhBbEiwAYiW--8mBvVR8NcmMw0Fwb7Ae9\_DZl4Ig7i3PxDbL3K5O5o2Rb7u8Q53DuBoCo4sQAvD\_BwE. Acesso em: 3 set. 2024.

MICROSOFT. Descrição das noções básicas de normalização do banco de dados. Microsoft Learn, 6 jun. 2024. Disponível em: <https://learn.microsoft.com/pt-br/office/troubleshoot/access/database-normalization-description>. Acesso em: 1 set. 2024.

MULLER, Vivien. Lua. Disponível em: <https://vivien-muller.fr/lua>. Acesso em: 28 fev. 2023.

NEVES, Vinicius. React JS. Alura, 2024. Disponível em: [https://www.alura.com.br/artigos/react-js?srsltid=AfmBOoqGrLlx7kkDkxom-9fr\\_W7uecweI8tGgZusv7hhMNIDjP412fCI](https://www.alura.com.br/artigos/react-js?srsltid=AfmBOoqGrLlx7kkDkxom-9fr_W7uecweI8tGgZusv7hhMNIDjP412fCI). Acesso em: 2 set. 2024.

OLIVEIRA, Euler. Como usar com Arduino: módulo Wi-Fi ESP8266 ESP-01. Blog Master Walker Shop. Disponível em: <https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-modulo-wifi-esp8266-esp-01>. Acesso em: 17 set. 2024.

ORACLE. O que é JSON?. Disponível em: [https://www.oracle.com/br/database/what-is-json/#:~:text=JavaScript%20Object%20Notation%20\(JSON\)%20%C3%A9,vez%20que%20n%C3%A3o%20requer%20deserializa%C3%A7%C3%A3o](https://www.oracle.com/br/database/what-is-json/#:~:text=JavaScript%20Object%20Notation%20(JSON)%20%C3%A9,vez%20que%20n%C3%A3o%20requer%20deserializa%C3%A7%C3%A3o). Acesso em: 2 set. 2024.

PEREIRA, Caio Ribeiro. Aplicações web real-time com Node.js. Brasil: Casa do Código, 2014.

PESSÔA, Camila. Node.js: definição, características, vantagens e usos. Alura, 08 mar. 2022. Disponível em:  
[https://www.alura.com.br/artigos/node-js-definicao-caracteristicas-vantagens-usos?utm\\_term=&utm\\_campaign=%5BSearch%5D+%5BPerformance%5D+-+Dynamic+Search+Ads+-+Artigos+e+Conte%C3%BAdos&utm\\_source=adwords&utm\\_medium=ppc&hsa\\_acc=7964138385&hsa\\_cam=11384329873&hsa\\_grp=164240702375&hsa\\_ad=703853654617&hsa\\_src=g&hsa\\_tgt=aud-396128415587:dsa-2276348409543&hsa\\_kw=&hsa\\_mt=&hsa\\_net=adwords&hsa\\_ver=3&gad\\_source=1&gclid=CjwKCAjwxNW2BhAkEiwA24Cm9N7C2EQYHUbkmhW0kAuyPQT\\_h4IiLNreeEUY1mZNXBZkIenJbwLlQxoCWk0QAvD\\_BwE](https://www.alura.com.br/artigos/node-js-definicao-caracteristicas-vantagens-usos?utm_term=&utm_campaign=%5BSearch%5D+%5BPerformance%5D+-+Dynamic+Search+Ads+-+Artigos+e+Conte%C3%BAdos&utm_source=adwords&utm_medium=ppc&hsa_acc=7964138385&hsa_cam=11384329873&hsa_grp=164240702375&hsa_ad=703853654617&hsa_src=g&hsa_tgt=aud-396128415587:dsa-2276348409543&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gad_source=1&gclid=CjwKCAjwxNW2BhAkEiwA24Cm9N7C2EQYHUbkmhW0kAuyPQT_h4IiLNreeEUY1mZNXBZkIenJbwLlQxoCWk0QAvD_BwE). Acesso em: 2 set. 2024.

PICHETTI, Roni Francisco; VIDA, Ednilson da Silva; PAIXÃO CORTES, Vanessa Stangherlin Machado. Banco de dados. Disponível em:  
<https://integrada.minhabiblioteca.com.br/>. Acesso em: 1 set. 2024.

PINTO, Ulisses Reis Correia. Características produtivas de tomate cereja em função da aplicação de fósforo via solo e fertirrigação em cultivo protegido. 2017. Trabalho de Conclusão de Curso (Programa de Pós-Graduação em Irrigação no Cerrado) – Instituto Federal de Educação, Ciência e Tecnologia Goiano, Campus Ceres. Ceres, 2017.

PRICE, Jason. Oracle Database 11g SQL: domine SQL e PL/SQL no banco de dados Oracle. Disponível em:  
<https://books.google.com.br/books?id=n3euJtdEM0UC&printsec=frontcover&hl=pt-BR#v=o nepage&q&f=false>. Acesso em: 1 set. 2024.

QUVVATOV, Behruz Ulugbek son. Web front-end and back-end technologies in programming. In: THEORETICAL ASPECTS IN THE FORMATION OF PEDAGOGICAL SCIENCES, 3., 2024. Anais... [S.1.]: Zenodo, 2024. p. 208–215. Disponível em:  
<https://doi.org/10.5281/zenodo.10518360>. Acesso em: 14 set. 2024.

ROBOCORE. Adaptador para ESP8266. Disponível em:  
<https://www.robocore.net/placa-robocore/adaptador-para-esp8266>. Acesso em: 3 set. 2024.

ROBOCORE. Bomba Peristáltica. Disponível em:  
<https://www.robocore.net/atuador/bomba-peristaltica>. Acesso em: 3 set. 2024.

ROBOCORE. Fonte Chaveada 12V 5A. Disponível em:  
<https://www.robocore.net/fonte/fonte-chaveada-12v-5a>. Acesso em: 3 set. 2024.

ROBOCORE. Irrigador Automático com Arduino. Disponível em:  
<https://www.robocore.net/tutoriais/irrigador-automatico-arduino>. Acesso em: 28 ago. 2024.

ROBOCORE. Módulo Relé. Disponível em:  
<https://www.robocore.net/atuador-rele/modulo-rele>. Acesso em: 3 set. 2024.

ROBOCORE. Placa Arduino Uno R3 com Cabo USB. Disponível em:  
<https://www.robocore.net/placa-arduino/placa-uno-r3-com-cabo-usb-para-arduino>. Acesso em: 3 set. 2024.

RUBIO, Wilmer. A agricultura 5.0: uma introdução ao uso de novas tecnologias. YouTube, 25 ago. 2024. Disponível em: [https://www.youtube.com/watch?v=fdsKJo\\_NVs0](https://www.youtube.com/watch?v=fdsKJo_NVs0). Acesso em: 15 set. 2024.

SANTOS, Jussara. Usando tecnologia de jardim inteligente em casa. Disponível em:  
<https://automacaoemcasa.com.br/automatizar-um-jardim/>. Acesso em: 08 out. 2024.

SILVA, Gabriela. O que é MongoDB e para que serve. Alura, 2024. Disponível em:  
[https://www.alura.com.br/artigos/o-que-e-mongodb?utm\\_source=ActiveCampaign&utm\\_medium=email&utm\\_content=Com+certeza+voc%C3%AA+vai+querer+participar+da+imers%C3%A3o+que+vai+mudar+os+rumos+do+seu+conhecimento+em+tecnologia&utm\\_campaign=fechado-plug+cobre+email&srsltid=AfmBOoqXYW5mSkek7f\\_SGlR-WVVtdTR-WT5btS2K8BcvDXfTzWls5RTf87E](https://www.alura.com.br/artigos/o-que-e-mongodb?utm_source=ActiveCampaign&utm_medium=email&utm_content=Com+certeza+voc%C3%AA+vai+querer+participar+da+imers%C3%A3o+que+vai+mudar+os+rumos+do+seu+conhecimento+em+tecnologia&utm_campaign=fechado-plug+cobre+email&srsltid=AfmBOoqXYW5mSkek7f_SGlR-WVVtdTR-WT5btS2K8BcvDXfTzWls5RTf87E).

SMART PLANT IVY. Ivy. Disponível em: <https://smartplantivy.com/products/ivy>. Acesso em: 28 fev. 2023.

SMITH, Adam. Building a REST API with Node.js. Medium, 25 ago. 2022. Disponível em: <https://medium.com/@adamsmith/building-a-rest-api-with-nodejs>. Acesso em: 15 set. 2024.

W3SCHOOLS. O que é JSON?. Disponível em: [https://www.w3schools.com/whatis/whatis\\_json.asp](https://www.w3schools.com/whatis/whatis_json.asp). Acesso em: 2 set. 2024.

W3SCHOOLS. O que é MongoDB?. Disponível em: [https://www.w3schools.com/whatis/whatis\\_mongodb.asp](https://www.w3schools.com/whatis/whatis_mongodb.asp). Acesso em: 3 set. 2024.

WLADIMIR. *React SPA: Entendendo o conceito de uma Single Page Application*. DevMedia, 2022. Disponível em: <https://www.devmedia.com.br/react-spa-entendendo-o-conceito-de-uma-single-page-application/43490>. Acesso em: 07 nov. 2024.

## APÊNDICE A - Código Fonte do Software

Neste apêndice, será fornecido o link para o repositório contendo o código fonte completo do projeto na parte de software, no caso de Frontend e Backend.

1 - Aplicação Web: <https://github.com/CamilaPrado27/Samanta>

Os repositórios contêm todas as funcionalidades e arquivos mencionados ao longo deste trabalho.

## APÊNDICE B - Código Fonte do Hardware

Neste apêndice, será fornecido o link para o repositório contendo o código fonte completo para a configuração do Arduino.

1 - Código Fonte: [https://github.com/WenderG/Samanta\\_v5/blob/main/Samanta\\_v5](https://github.com/WenderG/Samanta_v5/blob/main/Samanta_v5)

Os repositórios contêm todas as funcionalidades e arquivos mencionados ao longo deste trabalho.

## APÊNDICE C - Código Fonte do Banco de Dados

Neste apêndice, será fornecido o link para o repositório contendo o código fonte completo do projeto na parte do Banco de Dados.

1 - Código Fonte: [https://github.com/lucasvsrosa/SAMANTA\\_VASO\\_INTELIGENTE](https://github.com/lucasvsrosa/SAMANTA_VASO_INTELIGENTE)

Os repositórios contêm todas as funcionalidades e arquivos mencionados ao longo deste trabalho.