



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS SOBRAL**  
**CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO**

**MATEUS PAIVA ALVES**

**DESENVOLVIMENTO DE PLATAFORMA WEB PARA MONITORAMENTO DE  
DADOS PARA AGRICULTURA DE PRECISÃO**

**SOBRAL**

**2023**

MATEUS PAIVA ALVES

DESENVOLVIMENTO DE PLATAFORMA WEB PARA MONITORAMENTO DE DADOS  
PARA AGRICULTURA DE PRECISÃO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia De Computação do Campus Sobral da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia De Computação.

Orientador: Prof. Dr. Reuber Régis de Melo.

SOBRAL

2023

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

- A48d Alves, Mateus Paiva.  
DESENVOLVIMENTO DE PLATAFORMA WEB PARA MONITORAMENTO DE DADOS PARA AGRICULTURA DE PRECISÃO / Mateus Paiva Alves. – 2023.  
41 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Sobral, Curso de Engenharia da Computação, Sobral, 2023.  
Orientação: Prof. Dr. Reuber Régis de Melo.
1. Agricultura de precisão. 2. IoT. 3. Node-Red. 4. Desenvolvimento Web. 5. Vue.js. I. Título.  
CDD 621.39
-

MATEUS PAIVA ALVES

DESENVOLVIMENTO DE PLATAFORMA WEB PARA MONITORAMENTO DE DADOS  
PARA AGRICULTURA DE PRECISÃO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia De Computação do Campus Sobral da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia De Computação.

Aprovada em: 12 de Dezembro de 2023

BANCA EXAMINADORA

---

Prof. Dr. Reuber Régis de Melo (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Iális Cavalcante de Paula Júnior  
Universidade Federal do Ceará (UFC)

---

Prof. Me. David Nascimento Coelho  
Universidade Federal do Ceará (UFC)

À minha avó, Maria. A mulher mais forte que  
conheço.

## AGRADECIMENTOS

Queridos pais, Maria do Livramento e Francisco, gostaria de expressar minha mais profunda gratidão a vocês por serem a fonte de apoio e crença em mim ao longo desta jornada acadêmica. Suas palavras de incentivo, amor e confiança foram fundamentais para minha perseverança e sucesso. Agradeço de coração por serem os pilares que sustentaram cada desafio e triunfo, moldando-me não apenas como acadêmico, mas também como pessoa.

Aos dedicados professores da Universidade Federal do Ceará, do curso de Engenharia da Computação, e em particular ao meu orientador, Prof. Dr. Reuber Régis de Melo, agradeço pela orientação sábia, pelo apoio constante e pela inspiração que me proporcionaram ao longo desta jornada acadêmica. Suas contribuições foram cruciais para meu crescimento profissional e intelectual, e sou grato pela oportunidade de aprender com os melhores.

Agradeço aos amigos, Nicole, Jamille, Ray, Caio e Roberto, pelo valioso conhecimento compartilhado, pelos momentos leves e pela amizade que tornaram os desafios acadêmicos mais significativos. Vocês são parte essencial das memórias que carregarei dessa jornada, e sou grato por ter contado com colegas tão especiais.

Agradeço a Universidade Federal do Ceará, em especial ao campus de Sobral, pelos recursos disponíveis e pelo ambiente acadêmico que proporcionou o cenário propício para meu desenvolvimento. Cada instante vivido neste campus contribuiu significativamente para minha formação, e levo comigo uma profunda apreciação por tudo que vivenciei aqui.

Por fim, expresso minha gratidão a todos que, de alguma forma, contribuíram para que minha jornada acadêmica fosse melhor, por cada gesto, palavra de encorajamento e suporte recebido ao longo dessa jornada. Cada contribuição, por menor que seja, teve um impacto positivo em meu caminho acadêmico. Obrigado por fazerem parte dessa história e por serem parte fundamental do meu percurso até aqui.

"A vida é o que fazemos dela. As viagens são os viajantes. O que vemos não é o que vemos, senão o que somos." (Fernando Pessoa)

## RESUMO

A agricultura de precisão surgiu com o desenvolvimento tecnológico e a necessidade de aumentar a produtividade e reduzir o impacto ambiental. Nesse contexto, a Internet das coisas (IoT) tem um papel central na coleta de dados a partir de sensores para o monitoramento dos campos de cultivo. Os dados coletados são importantes para a tomada de decisão e podem ser enviados para nuvem usando diversas tecnologias. Assim, esse trabalho teve como objetivo de desenvolver uma plataforma Web para o monitoramento de dados de sensores enviados por microcontroladores inseridos no meio da agricultura de precisão. Para isso, foram utilizadas as tecnologias Node-Red como *Gateway* entre microcontrolador e aplicação em nuvem e vue.js para o desenvolvimento da parte visual do projeto. A eficiente configuração do Gateway Node-RED para processar dados de sensores destaca-se, resultando em um dashboard intuitivo na aplicação web.

**Palavras-chave:** Agricultura de precisão. IoT. Vue.js. Node-Red. Plataforma Web.

## **ABSTRACT**

Precision farming has emerged with technological development and the need to increase productivity and reduce environmental impact. In this context, Internet of Things (IoT) takes a central position in collecting data from sensors for monitoring crop fields. The collected data is important for decisions and can be sent to the cloud using various technologies. The aim of this work was to develop a web platform for monitoring sensor data sent by microcontrollers inserted in the precision agriculture environment. For this purpose, Node-Red were used as gateway between microcontroller and cloud application and vue.js for the development of the visual part of the project. The efficient configuration of the Node-RED Gateway to process sensor data is notable, resulting in an intuitive dashboard in the web application.

**Keywords:** Precision Agriculture. IoT. Vue.js. Node-Red. Web Platform.

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 1 – Arquitetura MVC . . . . .                           | 19 |
| Figura 2 – Metodologia . . . . .                               | 25 |
| Figura 3 – Visão geral do Projeto . . . . .                    | 26 |
| Figura 4 – Gateway Node-RED . . . . .                          | 27 |
| Figura 5 – Esquema do protótipo para coleta de dados . . . . . | 34 |
| Figura 6 – Pagina de Login . . . . .                           | 35 |
| Figura 7 – Dashboard . . . . .                                 | 35 |
| Figura 8 – Dashboard Dispositivos Móveis . . . . .             | 36 |

## LISTA DE ABREVIATURAS E SIGLAS

|      |                                     |
|------|-------------------------------------|
| AP   | Agricultura de Precisão             |
| APIs | Application Programming Interface   |
| GCP  | Google Cloud Platform               |
| HTTP | Hyper Text Transfer Protocol        |
| IaaS | Infrastructure as a Service         |
| IoT  | Internet of Things                  |
| JIT  | Just In Time                        |
| MQTT | Message Queuing Telemetry Transport |
| MVC  | Model - View - Controller           |
| NFC  | Near-field communication            |
| PaaS | Platform as a Service               |
| RFID | Radio-frequency Identification      |
| SaaS | Software as a Service               |
| SoC  | System On Chip                      |
| TCP  | Transmission Control Protocol       |
| TI   | Tecnologia da Informação            |
| VM   | Virtual Machine                     |
| WSN  | Wireless Sensor Networks            |

## SUMÁRIO

|                |   |           |
|----------------|---|-----------|
| <b>1</b>       | <b>INTRODUÇÃO</b>                                       | <b>12</b> |
| <b>1.1</b>     | <b>Objetivos</b>  | <b>13</b> |
| <i>1.1.1</i>   | <i>Objetivo geral</i>                                   | <i>13</i> |
| <i>1.1.2</i>   | <i>Objetivos específicos</i>                            | <i>13</i> |
| <b>1.2</b>     | <b>Justificativa</b>                                    | <b>13</b> |
| <b>1.3</b>     | <b>Trabalhos Relacionados</b>                           | <b>14</b> |
| <b>1.4</b>     | <b>Organização do Trabalho</b>                          | <b>15</b> |
| <b>2</b>       | <b>FUNDAMENTAÇÃO TEÓRICA</b>                            | <b>17</b> |
| <b>2.1</b>     | <b>Agricultura de Precisão</b>                          | <b>17</b> |
| <b>2.2</b>     | <b>NodeJS</b>   | <b>17</b> |
| <b>2.3</b>     | <b>Node-Red</b>   | <b>18</b> |
| <b>2.4</b>     | <b>JavaScript</b>                                       | <b>18</b> |
| <b>2.5</b>     | <b>Vue.js</b>   | <b>19</b> |
| <b>2.6</b>     | <b>Inertia.js</b>                                       | <b>20</b> |
| <b>2.7</b>     | <b>Laravel</b>  | <b>20</b> |
| <b>2.8</b>     | <b>Microcontroladores</b>                               | <b>20</b> |
| <b>2.9</b>     | <b>Tecnologias de sensoriamento</b>                     | <b>21</b> |
| <b>2.10</b>    | <b>Computação em Nuvem</b>                              | <b>21</b> |
| <i>2.10.1</i>  | <i>Google Cloud Platform</i>                            | <i>22</i> |
| <b>2.11</b>    | <b>GitHub</b>   | <b>24</b> |
| <b>3</b>       | <b>METODOLOGIA</b>                                      | <b>25</b> |
| <b>3.1</b>     | <b>Visão Geral</b>                                      | <b>26</b> |
| <b>3.2</b>     | <b>Gateway Node-RED</b>                                 | <b>26</b> |
| <i>3.2.1</i>   | <i>Provisionamento da Aplicação na Nuvem (GCP)</i>      | <i>28</i> |
| <b>3.3</b>     | <b>Desenvolvimento</b>                                  | <b>28</b> |
| <i>3.3.1</i>   | <i>Levantamento de Requisitos</i>                       | <i>28</i> |
| <i>3.3.2</i>   | <i>Requisitos Funcionais</i>                            | <i>29</i> |
| <i>3.3.2.1</i> | <i>Integração com a plataforma de nuvem:</i>            | <i>29</i> |
| <i>3.3.2.2</i> | <i>Disponibilidade e tolerância a falhas:</i>           | <i>29</i> |
| <i>3.3.2.3</i> | <i>Gerenciamento e monitoramento da infraestrutura:</i> | <i>29</i> |

|              |   |           |
|--------------|---|-----------|
| 3.3.2.4      | <i>Tela de Login e Cadastro:</i>        | 30        |
| 3.3.2.5      | <i>Gerenciamento de Usuários:</i>       | 30        |
| 3.3.2.6      | <i>Seleção de Sensores:</i>             | 30        |
| 3.3.2.7      | <i>Organização dos Sensores:</i>        | 30        |
| 3.3.2.8      | <i>Visualização de Dados:</i>           | 30        |
| 3.3.2.9      | <i>Personalização:</i>                  | 31        |
| 3.3.2.10     | <i>Alertas e Notificações:</i>          | 31        |
| 3.3.2.11     | <i>Exportação de Dados:</i>             | 31        |
| <b>3.3.3</b> | <b><i>Requisitos Não Funcionais</i></b> | <b>31</b> |
| 3.3.3.1      | <i>Usabilidade:</i>                     | 31        |
| 3.3.3.2      | <i>Desempenho:</i>                      | 31        |
| 3.3.3.3      | <i>Segurança:</i>                       | 32        |
| 3.3.3.4      | <i>Confiabilidade:</i>                  | 32        |
| 3.3.3.5      | <i>Escalabilidade:</i>                  | 32        |
| 3.3.3.6      | <i>Compatibilidade:</i>                 | 32        |
| 3.3.3.7      | <i>Manutenibilidade:</i>                | 33        |
| <b>4</b>     | <b>RESULTADOS</b>                       | <b>34</b> |
| <b>5</b>     | <b>CONCLUSÕES E TRABALHOS FUTUROS</b>   | <b>37</b> |
|              | <b>REFERÊNCIAS</b>                      | <b>38</b> |

## 1 INTRODUÇÃO

A agricultura moderna requer maior produção de alimentos para atender às demandas da crescente população mundial. O avanço da ciência e da tecnologia tem promovido o crescimento sustentável e a modernização do campo. Este fato resultou em métodos agrícolas inteligentes que permitem que agricultores monitorem as colheitas remotamente usando sensores e tenham sistemas de irrigação automatizados. As aplicações de Internet das Coisas (IoT, do inglês Internet of Things) baseados em sensores podem fornecer informações mais precisas sobre culturas, solos e clima do que os métodos tradicionais (GÓMEZ-CHABLA *et al.*, 2019).

Internet of Things (IoT) pode ser definida como "coisas que estão associadas através da Internet". Ou seja, a IoT ajuda na transferência de informações reunidas de vários dispositivos para lugares destinados, através da Internet. Essas coisas estão associadas à Internet geralmente por Wireless Sensor Networks (WSN), Radio-frequency Identification (RFID), Bluetooth, Near-field communication (NFC), 4G e várias outras tecnologias inteligentes de comunicação (KHANNA; KAUR, 2019).

Outro aspecto importante é a implementação de sistemas de controle e automação na agricultura, possibilitados pela IoT que pode ajudar a agricultura a aumentar a produtividade, reduzir o impacto ambiental e reduzir o desperdício. Tudo isso por meio de uma rede de sensores que monitoram constantemente o estado e a ordem dos insumos de plantas, animais e propriedades. Isso permite um ambiente ideal para o crescimento das plantas, maximizando a produtividade e garantindo condições adequadas de bem-estar. O monitoramento remoto das lavouras é tecnicamente difícil, pois requer transmissão, armazenamento e processamento de grande quantidade de dados (*Big Data*) gerados pela rede de sensores (ABDALA *et al.*, 2022).

Neste contexto, a importância de uma plataforma web dedicada ao monitoramento agrícola torna-se evidente. Este trabalho propõe o desenvolvimento de uma plataforma utilizando tecnologias como Node-Red, Vue.js, Laravel e Inertia.js, para agricultura de precisão. O propósito central é avaliar a viabilidade do monitoramento de dados, proporcionando aos agricultores uma ferramenta eficaz para gerenciar suas operações de maneira mais inteligente e sustentável. A escolha dessas tecnologias não apenas simplifica o processo, tornando-o mais acessível, mas também garante uma plataforma mais leve e eficiente para lidar com as demandas específicas da agricultura moderna.

## 1.1 Objetivos

### 1.1.1 *Objetivo geral*

O objetivo desse trabalho é desenvolver uma plataforma web baseada em Node-Red e Vue.js para agricultura de precisão, visando a viabilidade do monitoramento de dados.

### 1.1.2 *Objetivos específicos*

- Revisão bibliográfica sobre Agricultura de Precisão;
- Desenvolver uma comunicação entre um dispositivo microcontrolado (Arduino ou Node-MCU) e a aplicação Node-RED;
- Desenvolver a aplicação web de supervisão e hospedar na nuvem;
- Testar o sistema com medições reais obtidas por sensores.
- Processar as informações coletadas para tomada de decisões.

## 1.2 Justificativa

A agricultura é um setor econômico que influencia de forma muito significativa no desenvolvimento do Brasil, uma vez que ele é o quarto maior exportador mundial de grãos (QUINTAM; ASSUNÇÃO, 2023). Com a ampla aplicação da tecnologia IoT na agricultura, impulsionada pelo desenvolvimento da Internet, da tecnologia digital e da tecnologia de sensoriamento, os sensores agrícolas estão constantemente emergindo e se desenvolvendo no sentido de serem incorporados, inteligentes, integrados e miniaturizados (XU *et al.*, 2022). Esses avanços tecnológicos têm proporcionado uma diversificação das funções dos sensores agrícolas, abrangendo áreas como o monitoramento do solo, a análise meteorológica, a medição de recursos hídricos e a observação das plantas.

Os sensores de solo desempenham um papel crucial ao coletar informações sobre os nutrientes presentes no solo, sua umidade e salinidade, permitindo aos agricultores ajustar as práticas de fertilização e irrigação de forma precisa e eficiente. Já os sensores meteorológicos monitoram fatores climáticos, como temperatura, umidade do ar, velocidade do vento e radiação solar, auxiliando no planejamento de cultivos e no manejo de pragas e doenças. Os sensores de água têm como objetivo monitorar o uso dos recursos hídricos, medindo a quantidade de água disponível no solo, a taxa de evapotranspiração das plantas e o nível dos reservatórios. Com

esses dados, os agricultores podem otimizar o uso da água, adotando práticas de irrigação mais eficientes e evitando o desperdício (COELHO *et al.*, 2009).

Além disso, os sensores de plantas têm ganhado destaque ao fornecer informações sobre o crescimento, o desenvolvimento e a saúde das culturas. Por meio da detecção precoce de estresses, doenças ou deficiências nutricionais, os agricultores podem intervir de maneira oportuna, adotando medidas corretivas para garantir uma produção saudável e de alta qualidade.

Essa diversidade de sensores agrícolas e suas capacidades de detecção fornecem um suporte poderoso para a coleta de dados de produção agrícola (KASSIM, 2020). No entanto, a coleta de dados por si só não é suficiente. É essencial um manejo eficaz desses dados para tomar decisões focadas no aumento da produtividade e na diminuição dos impactos ambientais.

Por meio da análise e interpretação dos dados coletados pelos sensores agrícolas, os agricultores podem obter (insights) valiosos para otimizar suas práticas agrícolas. Essas informações permitem a adoção de medidas preventivas e corretivas de forma mais precisa, como a aplicação pontual de fertilizantes e pesticidas, reduzindo o uso excessivo desses insumos e minimizando os impactos negativos no meio ambiente.

Além disso, o manejo de dados possibilita a implementação de sistemas de monitoramento em tempo real, permitindo uma tomada de decisão ágil e adaptativa. Com acesso a informações atualizadas sobre as condições ambientais e o desenvolvimento das culturas, os agricultores podem ajustar suas práticas de manejo de acordo com as necessidades específicas de cada área, maximizando a eficiência produtiva.

Dessa forma, a aplicação da tecnologia IoT na agricultura, aliada ao uso de sensores agrícolas, representa uma oportunidade para impulsionar a produtividade agrícola de forma sustentável. Ao permitir a coleta e o manejo eficaz dos dados de produção, os agricultores podem tomar decisões embasadas em informações precisas, contribuindo para o aumento da eficiência produtiva, a redução dos impactos ambientais e o desenvolvimento sustentável do setor agrícola (MUNIZ, 2021).

### **1.3 Trabalhos Relacionados**

Nesta seção serão discutidos os principais livros e artigos utilizados para o desenvolvimento da plataforma, que serviram como referência para este trabalho.

Coelho *et al.* (2020) propõe uma aplicação agrícola que é focada no desenvolvimento de um sistema de coleta de dados de parâmetros. Para tal, o sistema inclui a monitorização

da umidade do solo e sensores atmosféricos (temperatura e umidade relativa) para apoiar os agricultores na tomada de decisão, com o objetivo de implementar um futuro sistema de irrigação automatizado de forma a minimizar o desperdício de água. A coleta de dados é feita por sensores conectados a um sistema microcontrolador, e os sinais são transmitidos por um módulo de radiofrequência utilizando o protocolo LoRaWan. Os dados são recebidos no *gateway* Node-red e disponibilizados na nuvem e podem ser monitorados em tempo real. Em Sunehra e Srinidhi (2020) é proposto um sistema usando Node-Red que visa monitorar e manter os parâmetros essenciais de crescimento, como luz, temperatura, que prevalecem em uma planta, assegurando um rendimento máximo.

Lekić e Gardašević (2018) apresenta a implementação de uma aplicação de IoT que realiza a detecção de temperatura e umidade usando o sensor DHT11 no Raspberry Pi e a transferência de dados para a nuvem. A implementação é feita utilizando o sistema de programação Raspbian Stretch Lite sobre Raspberry Pi e a plataforma IBM Internet of Things baseada na ferramenta Node-RED instalada no Raspberry Pi e na IBM Cloud.

O artigo Duguma e Bai (2023) discute como a Internet das Coisas (IoT) pode ser combinada com outras tecnologias importantes, como computação em nuvem, armazenamento de grandes volumes de dados e análise de dados, para melhorar os sistemas agrícolas. Os autores destacam a importância da IoT na agricultura e como ela pode ser usada para coletar e analisar dados em tempo real para melhorar a eficiência e a produtividade. Eles também discutem os desafios associados à implementação da IoT na agricultura, incluindo questões de segurança, privacidade e interoperabilidade.

Muniz (2021) aborda a presença da Internet das Coisas (IoT) na agricultura irrigada, com enfoque nos fruticultores do Nordeste brasileiro. São apresentadas as tecnologias da agricultura de precisão e seus benefícios para os produtores. Além disso, são mencionadas as principais tecnologias de sensoriamento e automação utilizadas nas culturas irrigadas, juntamente com os problemas que elas solucionam. O autor compartilha sua experiência com as soluções da empresa 3V3 Tecnologia, que englobam o monitoramento de solo, variáveis ambientais, processos produtivos e a automação de válvulas, bombas e reservatórios. Por fim, são destacados os benefícios das tecnologias de IoT para produtores de diferentes tamanhos.

#### **1.4 Organização do Trabalho**

O presente trabalho está estruturado em cinco capítulos:

- **Capítulo 1 - Introdução:** Foi apresentado o tema junto com uma contextualização, os objetivos gerais e específicos, a justificativa, trabalhos relacionados e a estrutura da monografia;
- **Capítulo 2 - Fundamentação teórica:** Capítulo que descreve as principais tecnologias usadas para a construção da monografia e para o desenvolvimento da plataforma;
- **Capítulo 3 - Metodologia:** Nesse capítulo são descritos os passos para a criação do *gateway* e para a criação da plataforma;
- **Capítulo 4 - Resultados:** Nesta seção, são apresentados os resultados obtidos durante a implementação do projeto. Como os sensores coletaram dados, como o Node-RED processou e encaminhou esses dados para a API e como o aplicativo vue.js exibiu os dados no (dashboard).
- **Capítulo 5 - Conclusão:** É feita uma análise dos resultados obtidos em relação aos objetivos do projeto. Os desafios encontrados, as lições aprendidas e as contribuições do trabalho. Além disso, são sugeridos trabalhos futuros, como melhorias no sistema, expansões ou novas funcionalidades a serem adicionadas.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Agricultura de Precisão

A Agricultura de Precisão (AP) é uma técnica de manejo que leva em conta a variabilidade espacial e permite a aplicação de insumos específicos do local, como fertilizantes, agentes de tratamento, pesticidas, sementes, água e outros. Além disso, dada a variação temporal, a AP permite um uso mais racional de insumos na dose certa, no lugar certo e na hora certa, com possíveis benefícios econômicos e ambientais. Como um sistema de gestão da produção que depende fortemente de dados e informações de campo (plantas, animais, solos, clima, maquinário etc.), georreferenciamento, digitalização e alta produtividade, o AP fornece a base estrutural e conceitual para conectar os sistemas de produção agrícola com o mundo digital, que abre um canal para coletar e compartilhar dados de campo (BASSOI *et al.*, 2019).

A AP inclui um conjunto de tecnologias que combinam sensores, sistemas de informação, maquinaria otimizada e gestão de informação para melhorar a produção tendo em conta a variabilidade e incerteza dos sistemas agrícolas. AP é uma abordagem de gestão agrícola que usa Tecnologia da Informação (TI) para garantir que as culturas e os solos recebam exatamente os recursos de que precisam para uma produtividade ideal. Os objetivos da AP são garantir rentabilidade, sustentabilidade e proteção ambiental (KHANNA; KAUR, 2019).

### 2.2 NodeJS

Node.js é um ambiente *Runtime* JavaScript que move a renderização e o processamento do código JavaScript para o lado do servidor e o separa completamente do navegador, permitindo o desenvolvimento de aplicativos Web rápidos e estáveis (SHAH, 2017). Desenvolvido para o motor JavaScript V8 - um motor criado pelo Google usado nos navegadores Chrome e Chromium, o Node.js é orientado a eventos (event driven) e não bloqueia o uso de solicitações de I/O (input e output), o que garante estabilidade e baixo consumo de recursos. O objetivo do desenvolvimento da tecnologia era fornecer uma maneira fácil de criar aplicativos escaláveis (PEREIRA, 2014).

### 2.3 Node-Red

Node-RED é uma ferramenta de ambiente visual de código aberto que foi originalmente desenvolvida para implantar, construir e/ou conectar dispositivos IoT, depois estendida para dispositivos, Application Programming Interface (APIs) e serviços da web. Assim é possível ler arquivos CSV, eventos Hyper Text Transfer Protocol (HTTP), Transmission Control Protocol (TCP), WebSocket, eventos Message Queuing Telemetry Transport (MQTT), dentre outros através de nós. O Node-Red possui um editor baseado em navegador que, além de ser simples e compatível com vários navegadores, permite fácil conexão de fluxos usando os nós do painel. Assim como o protocolo MQTT, que é objeto de outras publicações no espaço IoT, o Node-RED foi criado pela IBM Emerging Technology (LEKIĆ; GARDAŠEVIĆ, 2018).

### 2.4 JavaScript

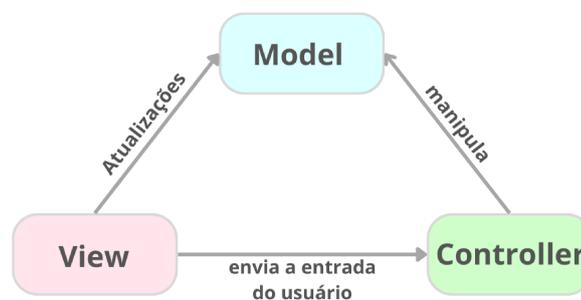
A linguagem de programação JavaScript é uma ferramenta poderosa e versátil que tem se tornado cada vez mais popular entre os desenvolvedores. Ela é usada tanto no lado do cliente quanto no lado do servidor, proporcionando uma experiência dinâmica e interativa para os usuários da web. Um dos principais benefícios do JavaScript é a sua velocidade. Como é frequentemente executado diretamente no navegador do cliente, o JavaScript tende a ser muito rápido, pois não é retardado por chamadas a um servidor de back-end (DELCEV; DRASKOVIC, 2018). Além disso, todos os principais navegadores suportam a compilação Just In Time (JIT) para JavaScript, o que significa que não há necessidade de compilar o código antes de executá-lo (JOHANSSON, 2020).

Outra vantagem do JavaScript é a sua simplicidade. Sua sintaxe foi inspirada na do Java e é relativamente fácil de aprender em comparação com outras linguagens populares como C++. Além disso, o JavaScript é onipresente na web e, com o advento do Node.js, está sendo cada vez mais usado no back-end. O JavaScript também é a melhor escolha para o desenvolvimento de sites para desktop e dispositivos móveis. Com ferramentas como JQuery, Angular e vue.js, o JavaScript oferece praticamente infinitas capacidades para a programação web (DELCEV; DRASKOVIC, 2018).

## 2.5 Vue.js

Vue.js é um *framework* JavaScript para a construção de interfaces de usuário, o que a torna o 'V' no *Model - View - Controller (MVC)*, padrão mostrado na Figura 1. Ao contrário de outros frameworks monolíticos, Vue.js foi projetado desde o início para ser adotado de forma incremental. A biblioteca principal se concentra apenas na camada de visualização, tornando fácil pegar e integrar com outras bibliotecas ou projetos existentes. Vue.js foi criado por Evan You depois de trabalhar no Google usando AngularJS em vários projetos(KAEWSANMUANG, 2019).

Figura 1 – Arquitetura MVC



Fonte: (WIERUCH, 2017)

Vue.js é usado apenas para construir sua camada de visualização. Especificamente, é uma biblioteca na qual a visualização é uma hierarquia de componentes integráveis(VUE.JS, 2023). Vue.js é uma das melhores escolhas para construir aplicações web modernas. Mais uma vez, ele só oferece a visão, mas o ecossistema abrangente constitui uma estrutura totalmente flexível e reutilizável. Vue.js tem uma APIs leve, um ecossistema robusto e uma grande comunidade em evolução(VUE SCHOOL, 2023).

Vue.js garante uma codificação mínima para os desenvolvedores através do sistema reativo que usa objetos JavaScript simples e otimiza a renderização. Isso mantém seu código (variáveis e valores de array) sincronizado em toda a aplicação. Vue.js é uma escolha popular e está ganhando muita tração no campo do desenvolvimento front-end. A simplicidade do Vue.js significa que ele pode ser implementado de forma rápida e eficiente, reduzindo o tempo e os custos de desenvolvimento(MONTERAIL, 2023).

## 2.6 Inertia.js

Inertia.js é uma ferramenta inovadora que permite aos desenvolvedores criar aplicações de página única (SPA) com renderização tanto no lado do cliente quanto no lado do servidor. Ele é chamado de “monólito moderno” porque permite criar SPAs totalmente renderizadas no lado do cliente, sem a complexidade que vem com as SPAs modernas. Ele faz isso aproveitando os padrões de servidor existentes.

Um dos principais benefícios do Inertia.js é que ele permite criar aplicações web modernas usando rotas do lado do servidor. Além disso, o Inertia.js funciona muito bem com qualquer *framework* de back-end, mas é otimizado para Laravel. Outra vantagem do Inertia.js é que ele não é um *framework*, nem é um substituto para os *frameworks* existentes no lado do servidor ou do cliente. Em vez disso, ele é projetado para trabalhar com eles (ROSAS, 2023).

## 2.7 Laravel

Laravel é um *framework* PHP que tem ganhado popularidade entre os desenvolvedores devido à sua robustez e versatilidade. Ele é amplamente utilizado para o desenvolvimento de aplicações web, especialmente para sites de comércio eletrônico. Um dos principais benefícios do Laravel é a sua velocidade. Como é executado diretamente no servidor, o Laravel tende a ser muito rápido, pois não é retardado por chamadas a um servidor de back-end. Além disso, o Laravel suporta uma variedade de bancos de dados e oferece uma sintaxe expressiva e elegante que facilita a escrita de código (YADAV *et al.*, 2019).

Outra vantagem do Laravel é a sua segurança. Ele vem com várias funcionalidades de segurança integradas, como proteção CSRF, proteção contra injeção SQL e proteção contra ataques XSS. Além disso, o Laravel é atualizado regularmente para corrigir quaisquer vulnerabilidades de segurança que possam surgir. O Laravel também é conhecido por sua escalabilidade. Ele pode facilmente lidar com grandes volumes de tráfego e pode ser escalado para atender às necessidades de crescimento de uma aplicação (VALAREZO; GUARDA, 2018).

## 2.8 Microcontroladores

Uma das camadas bastante utilizadas em IoT é a camada de sensores e atuadores que são gerenciados por microcontroladores. Os microcontroladores simplificam o uso da Internet das Coisas, estão disponíveis em diversos modelos, tamanhos e preços. Um microcontrolador System

On Chip (SoC) é um pequeno computador em um único circuito integrado que contém um núcleo de processador, memória e periféricos programáveis de entrada e saída. Os microcontroladores são projetados para aplicações embarcadas, ao contrário dos microprocessadores usados em computadores pessoais ou outros aplicativos de uso geral. Os microcontroladores são usados em produtos e dispositivos automatizados (CHÉOUR *et al.*, 2020).

Os microcontroladores têm um consumo de energia relativamente baixo, geralmente em miliwatts, e pode entrar no modo de hibernação (*Sleep* ou *Wait*) para aguardar uma interrupção ou um evento externo, como um pressionamento de tecla ou um sinal de entrada, através da interface de dados. O consumo em *standby* desses microcontroladores pode chegar a nanowatts (SILVA *et al.*, 2019), tornando-os ideais para aplicações onde a exigência de baixo consumo de energia é um fator crítico para o sucesso do projeto.

## 2.9 Tecnologias de sensoriamento

A agricultura moderna, também conhecida como Agricultura de precisão, requer informações rápidas, precisas e baseadas em fontes confiáveis. No entanto, nem sempre é fácil obter esses dados prontamente disponíveis. Por essa razão, surgem tecnologias que transformam as características do ambiente em valores mensuráveis, permitindo que sejam transmitidos como informações úteis para os agricultores. Os dispositivos responsáveis por estabelecer essa conexão entre o mundo físico e o virtual são os sensores, que têm a capacidade de detectar e registrar diversas mudanças. Essas informações são convertidas em dados que podem ser interpretados de forma ágil por pessoas ou computadores (MUNIZ, 2021).

## 2.10 Computação em Nuvem

A computação em nuvem é uma forma de fornecer recursos de computação, armazenamento, banco de dados e outros serviços através da internet (SADEEQ *et al.*, 2021). Em vez de possuir e gerenciar hardware e software em um *data center* próprio, é possível acessar recursos *on demand* em um provedor de nuvem. Existem muitos benefícios em usar computação em nuvem, incluindo:

- Escalabilidade: os recursos em nuvem podem ser aumentados ou diminuídos conforme necessário, permitindo que as empresas se ajustem facilmente às necessidades em constante mudança.

- Disponibilidade: os provedores de nuvem geralmente têm data centers em todo o mundo, o que garante alta disponibilidade e redundância para evitar interrupções no serviço.
- Segurança: os provedores de nuvem geralmente têm equipes dedicadas de segurança que monitoram e protegem os recursos em nuvem contra ameaças.
- Redução de custos: o uso de recursos em nuvem pode ser mais econômico do que investir em hardware e software localmente, pois as empresas pagam apenas pelo que usam.
- Acesso remoto: a computação em nuvem permite que os usuários acessem seus recursos a partir de qualquer lugar do mundo, desde que tenham uma conexão com a internet.

Há três tipos principais de serviços em computação em nuvem: IaaS, PaaS e SaaS. Infrastructure as a Service (IaaS) é um modelo em que o provedor de nuvem fornece acesso a recursos de infraestrutura, como servidores virtuais, armazenamento e rede. Os clientes usam esses recursos para construir e gerenciar suas próprias aplicações. Platform as a Service (PaaS) é um modelo em que o provedor de nuvem fornece uma plataforma de desenvolvimento que os clientes usam para construir suas próprias aplicações. O provedor gerencia a infraestrutura subjacente, como servidores, armazenamento e rede, enquanto os clientes desenvolvem e implantam seus aplicativos na plataforma. Software as a Service (SaaS) é um modelo em que o provedor de nuvem fornece software diretamente aos usuários finais, geralmente por meio de um navegador web. Os clientes não precisam se preocupar com a infraestrutura ou o gerenciamento do software, pois o provedor de nuvem cuida disso.

Em resumo, computação em nuvem é uma forma flexível, escalável e econômica de fornecer recursos de computação e armazenamento por meio da internet. Com o modelo de nuvem, as empresas podem ter acesso a recursos poderosos e confiáveis sem investir em infraestrutura e manutenção local. (BELLO *et al.*, 2021)

### **2.10.1 Google Cloud Platform**

A Google Cloud Platform (GCP) é uma plataforma de computação em nuvem oferecida pela Google, projetada para atender às necessidades de empresas e desenvolvedores que buscam recursos escaláveis, confiáveis e flexíveis na nuvem. A GCP oferece uma ampla gama de serviços e ferramentas que abrangem desde infraestrutura de computação até análise de dados e inteligência artificial (BISONG; BISONG, 2019).

Em termos de infraestrutura, a GCP disponibiliza o Google Compute Engine, que permite a criação e o gerenciamento de máquinas virtuais altamente escaláveis e com desempenho

rápido. Além disso, o GCP oferece serviços de armazenamento, como o Cloud Storage, que permite o armazenamento seguro e durável de objetos, e o Cloud SQL, um serviço de banco de dados relacional totalmente gerenciado. Uma das áreas em que a GCP se destaca é a de inteligência artificial e aprendizado de máquina. Ela oferece recursos avançados, como o TensorFlow, um popular framework de aprendizado de máquina desenvolvido pela Google, que permite criar e treinar modelos personalizados. Além disso, a GCP disponibiliza serviços como o AutoML, que simplifica o processo de criação de modelos de aprendizado de máquina sem a necessidade de conhecimentos avançados em programação.

A análise de dados é outra área-chave da GCP. O BigQuery é um serviço de análise de dados em larga escala que permite executar consultas SQL em conjuntos de dados massivos de forma rápida e eficiente. Além disso, o Dataflow oferece recursos para processamento e análise de dados em tempo real, enquanto o Pub/Sub permite a ingestão e o processamento de eventos em tempo real em escala global.

A segurança e a conformidade também são aspectos essenciais da GCP. A plataforma adota práticas de segurança líderes do setor e oferece recursos avançados de segurança para proteger os dados dos clientes. Além disso, a GCP é certificada por várias normas de segurança e conformidade, garantindo a conformidade com regulamentações específicas de setores, como a HIPAA (Lei de Portabilidade e Responsabilidade do Seguro de Saúde) e a GDPR (Regulamento Geral de Proteção de Dados).

No que diz respeito ao gerenciamento e monitoramento, a GCP oferece ferramentas abrangentes para facilitar o gerenciamento e a operação dos recursos na nuvem. O Cloud Monitoring permite monitorar o desempenho, a disponibilidade e a integridade dos serviços, enquanto o Cloud Deployment Manager simplifica o processo de implantação e gerenciamento de recursos.

A GCP também se destaca por seu ecossistema e integrações. Ela oferece suporte ao Kubernetes, uma plataforma de orquestração de contêineres amplamente utilizada, e possui integrações com outras tecnologias populares, como o Docker e o Hadoop. Isso permite que os usuários aproveitem suas ferramentas e plataformas favoritas na GCP, facilitando a migração e o desenvolvimento de aplicativos.

Em relação aos preços, a GCP adota um modelo flexível que permite aos usuários escolher entre preços sob demanda ou contratos de uso sustentado para obter descontos. Isso permite que as empresas ajustem os custos de acordo com as necessidades e o volume de uso

dos serviços (SHAH; DUBARIA, 2019).

## 2.11 GitHub

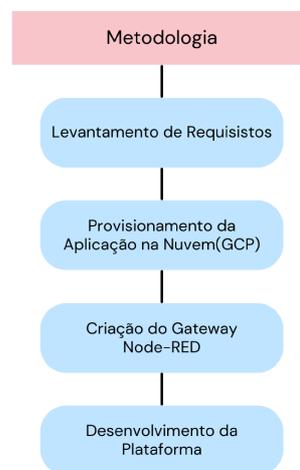
O GitHub é uma plataforma de hospedagem e gerenciamento de código fonte amplamente utilizada por desenvolvedores de software e equipes de desenvolvimento. Ele fornece um repositório centralizado para armazenar e controlar as alterações de um projeto, utilizando o sistema de controle de versão Git. Isso permite que várias pessoas trabalhem em um projeto simultaneamente, façam alterações independentes e depois as combinem de forma organizada.

Um dos recursos principais do GitHub são os repositórios, que são espaços onde os projetos são armazenados. Cada repositório contém todo o histórico de alterações, ramos (branches) para desenvolvimento paralelo e recursos para discussões e gerenciamento de problemas. Essa estrutura facilita a colaboração e o gerenciamento de conflitos entre diferentes desenvolvedores(COSENTINO *et al.*, 2017).

### 3 METODOLOGIA

Nessa seção será apresentada a metodologia utilizada para o desenvolvimento de uma aplicação baseada em Node-Red para receber dados coletados em microcontroladores no cenário da agricultura de precisão. O desenvolvimento do *dashboard* de agricultura de precisão se divide em duas partes principais: a criação do Gateway Node-RED e o desenvolvimento da aplicação web. As principais etapas são mostradas na Figura 2.

Figura 2 – Metodologia



Fonte: o autor.

A primeira etapa envolve a criação do Gateway Node-RED, que tem como objetivo receber os dados coletados pelos sensores, processá-los e enviá-los para a aplicação web. O Node-RED é uma plataforma de desenvolvimento visual que permite a criação de fluxos de dados de forma intuitiva, utilizando nós pré-configurados. Esse Gateway é responsável por receber os dados dos microcontroladores instalados nos dispositivos de coleta de dados na agricultura de precisão e encaminhá-los para a aplicação web para posterior análise e visualização.

A segunda parte consiste no desenvolvimento da aplicação web propriamente dita. Essa aplicação será responsável por exibir os dados coletados pelos sensores em um formato visualmente atraente e intuitivo para os usuários. Nessa seção, será apresentada a metodologia utilizada para o desenvolvimento dessa aplicação baseada em Node-RED, que permitirá receber, processar e apresentar os dados provenientes dos microcontroladores no cenário da agricultura de precisão.

Combinando o Gateway Node-RED e a aplicação web, os usuários terão acesso a um

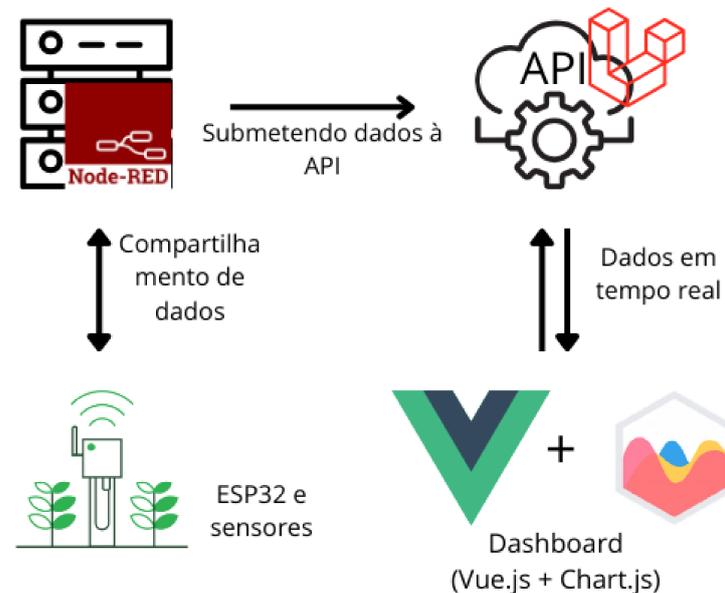
*dashboard* completo e funcional, onde podem visualizar os dados coletados pelos sensores de forma organizada e compreensível, auxiliando na tomada de decisões na agricultura de precisão.

É importante ressaltar que essa abordagem baseada em Node-RED oferece uma maneira flexível e escalável de criar e gerenciar o fluxo de dados entre os dispositivos de coleta e a aplicação web, facilitando o desenvolvimento e a manutenção do sistema como um todo.

### 3.1 Visão Geral

Neste projeto, esquematizado na Figura 3, sensores conectados ao ESP coletam dados, que são transmitidos via Wi-Fi para um gateway Node-RED. O Node-RED processa os dados, autentica os dispositivos e os envia para uma API externa. Essa API armazena os dados e fornece acesso a eles para uma aplicação vue.js, que exibe os dados em um *dashboard*. Isso permite monitorar e visualizar informações dos sensores de forma eficaz.

Figura 3 – Visão geral do Projeto



Fonte: o autor.

### 3.2 Gateway Node-RED

Para desenvolver um gateway Node-RED, mostrado na Figura 4, para uma aplicação em agricultura de precisão que utiliza sensores de temperatura e umidade e a placa ESP32 para se conectar à rede, deve-se seguir os seguintes passos.

Primeiro, a placa ESP32 é configurada. Deve-se certificar de ter o ambiente de desen-

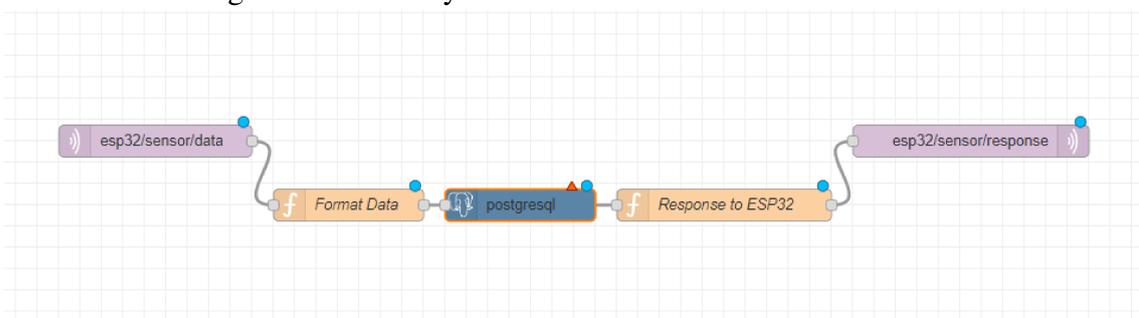
volvimento adequado, como a Arduino IDE. A placa ESP32 deve ser conectada ao computador para se comunicar com os sensores de temperatura e umidade. São utilizadas bibliotecas apropriadas para cada sensor e analisa-se a leitura dos sensores para garantir que estejam funcionando corretamente.

Em seguida, o Node-RED é instalado. Pode-se instalá-lo localmente no computador ou em um servidor. Após a instalação, é configurado o fluxo no Node-RED, Após iniciado, o editor do Node-RED é acessado pelo navegador. Um novo fluxo em branco é criado e os nós necessários são adicionados ao canvas. Para permitir a comunicação entre a placa ESP32 e o Node-RED, adiciona-se um nó MQTT configurado com as informações de conexão, como endereço do servidor MQTT e credenciais, se necessário. Conecta-se o nó MQTT ao nó de leitura dos sensores na placa ESP32. Nós de processamento são adicionados, como conversão de dados, filtragem ou cálculos específicos para a aplicação de agricultura de precisão. Por fim, os nós de saída são incluídos para exibir os dados ou enviá-los para outras plataformas ou sistemas.

Na programação da placa ESP32, a biblioteca MQTT é utilizada para se conectar ao servidor MQTT configurado no Node-RED. As informações de conexão são inseridas, como endereço IP do servidor MQTT e credenciais. Por fim, os dados dos sensores de temperatura e umidade são enviados por meio do protocolo MQTT para o Node-RED.

Após configurar o fluxo e a placa ESP32, o fluxo deve ser testado no Node-RED. O fluxo deve ser iniciado e analisado para averiguar se os dados dos sensores estão sendo recebidos corretamente e se o processamento ocorre conforme o esperado. Após isso deve-se fazer ajustes no fluxo, se necessário, para corrigir problemas.

Figura 4 – Gateway Node-RED



Fonte: o autor.

### **3.2.1 Provisionamento da Aplicação na Nuvem (GCP)**

Para o projeto, será utilizado o serviço de nuvem GCP, escolhido devido à usabilidade. Para utilizá-lo, é necessário acessar a Google Cloud Console e proceder com a criação de uma conta ou fazer login em uma já existente. Posteriormente, é preciso configurar a faturação e criar um novo projeto na plataforma Google Cloud. Essa conta servirá como base para hospedar a aplicação web e configurar o gateway Node-RED para a agricultura de precisão.

No ambiente de desenvolvimento local, é iniciada a instalação do Laravel e criada uma aplicação vue.js usando o Composer. A aplicação web deve ser desenvolvida com Laravel no backend e vue.js no frontend.

O próximo estágio envolve a realização da implantação da aplicação. O Google App Engine pode ser escolhido para hospedar o Laravel. Uma Virtual Machine (VM) no Compute Engine da GCP é criada para hospedar o Node-RED. O Node-RED é configurado para receber dados provenientes dos sensores de agricultura de precisão e processá-los.

Para proteger os sistemas, a autenticação é configurada tanto para a aplicação web quanto para o gateway Node-RED. Além disso, é importante estabelecer regras de autorização para controlar o acesso a recursos e informações sensíveis. Pode-se definir alertas que possam notificar sobre problemas em tempo real. É fundamental manter backups regulares dos dados e garantir que o código e as bibliotecas sejam mantidos atualizados, visando a manutenção da segurança e eficiência contínua da aplicação na GCP.

## **3.3 Desenvolvimento**

Os tópicos a seguir apresentam maiores detalhes dos passos de cada etapa para o desenvolvimento desta aplicação web.

### **3.3.1 Levantamento de Requisitos**

O levantamento de requisitos é um processo crucial na engenharia de software. Ele envolve a descoberta e documentação das necessidades e requisitos de tal forma que possam servir como base para todas as outras atividades de desenvolvimento do sistema. A engenharia de requisitos é a fase mais importante do ciclo de vida do desenvolvimento de software (SDLC). Envolve a coleta, análise, documentação, validação e gerenciamento dos requisitos do sistema de software proposto. São divididos em requisitos funcionais e requisitos não funcionais (TUKUR

*et al.*, 2021).

### **3.3.2 Requisitos Funcionais**

Requisitos funcionais são as especificações das funcionalidades ou do comportamento esperado do sistema. Eles descrevem o que o sistema deve fazer e quais ações deve executar em resposta a determinados eventos. Esses requisitos estão relacionados diretamente às funcionalidades do software e são geralmente expressos em termos de entradas, saídas e comportamentos esperados (WIEGERS; BEATTY, 2013).

#### **3.3.2.1 Integração com a plataforma de nuvem:**

A aplicação web do *dashboard* de agricultura de precisão deve ser capaz de se integrar e utilizar os serviços e recursos oferecidos pela plataforma de nuvem selecionada, como a Google Cloud Computing. Isso pode envolver a comunicação com bancos de dados, o uso de serviços de armazenamento de arquivos, gerenciamento de autenticação e autorização, entre outros.

#### **3.3.2.2 Disponibilidade e tolerância a falhas:**

É importante que a aplicação seja projetada para aproveitar os recursos de alta disponibilidade e tolerância a falhas oferecidos pela plataforma de nuvem. Isso inclui a utilização de mecanismos de balanceamento de carga para distribuir as solicitações de forma equilibrada, a redundância de servidores para garantir a continuidade do serviço em caso de falhas e estratégias de recuperação para minimizar o impacto de possíveis interrupções.

#### **3.3.2.3 Gerenciamento e monitoramento da infraestrutura:**

A aplicação deve ser capaz de aproveitar os recursos de gerenciamento e monitoramento da infraestrutura de nuvem. Isso pode incluir recursos como *logs* para rastreamento de eventos, métricas de desempenho para análise e otimização, alarmes para detecção de problemas e painéis de controle para visualização e gerenciamento da infraestrutura.

#### 3.3.2.4 *Tela de Login e Cadastro:*

A aplicação web deve incluir uma tela de login e cadastro para os usuários. Os usuários devem ter a opção de criar uma conta fornecendo informações básicas, como nome, email e senha. Além disso, os usuários registrados devem poder fazer login na aplicação utilizando suas credenciais.

#### 3.3.2.5 *Gerenciamento de Usuários:*

A aplicação deve oferecer recursos de gerenciamento de usuários para os administradores. Isso inclui a capacidade de criar, editar e excluir contas de usuário. Os administradores devem ter a opção de atribuir diferentes papéis e permissões aos usuários, como administrador, usuário regular, etc.

#### 3.3.2.6 *Seleção de Sensores:*

Os usuários devem ter a possibilidade de selecionar quais sensores desejam visualizar no *dashboard*. A aplicação deve exibir uma lista dos sensores disponíveis para seleção. Além disso, os usuários devem poder salvar suas seleções de sensores para utilização em sessões futuras.

#### 3.3.2.7 *Organização dos Sensores:*

Para facilitar a navegação e visualização, os sensores devem ser agrupados em seções ou categorias. Dessa forma, os usuários podem selecionar uma seção específica para visualizar os sensores associados a ela.

#### 3.3.2.8 *Visualização de Dados:*

O *dashboard* da aplicação deve apresentar de forma clara e organizada os dados coletados pelos sensores. Os usuários devem poder visualizar os dados em tempo real ou selecionar intervalos de tempo específicos. Dependendo do tipo de dado e sensor, os dados podem ser exibidos em formatos apropriados, como tabelas, gráficos ou mapas.

### 3.3.2.9 *Personalização:*

Para atender às preferências individuais dos usuários, a aplicação deve permitir personalizar o layout e as opções de exibição do *dashboard*. Os usuários devem poder reorganizar os elementos do *dashboard* de acordo com suas preferências pessoais.

### 3.3.2.10 *Alertas e Notificações:*

A aplicação deve fornecer recursos para configurar alertas e notificações com base em condições predefinidas nos dados dos sensores. Os usuários devem poder definir limites ou critérios específicos que, quando atingidos, acionem alertas.

### 3.3.2.11 *Exportação de Dados:*

É importante que os usuários tenham a opção de exportar os dados coletados pelos sensores. A aplicação deve permitir exportar os dados em formatos comuns, como CSV ou Excel. Além disso, os usuários devem poder selecionar um intervalo de tempo específico para a exportação dos dados.

## 3.3.3 *Requisitos Não Funcionais*

Requisitos não funcionais são os critérios de qualidade que especificam como o sistema deve se comportar em termos de desempenho, segurança, usabilidade, confiabilidade, escalabilidade, entre outros aspectos. Eles não estão diretamente relacionados às funcionalidades específicas, mas sim às características gerais do sistema (WIEGERS; BEATTY, 2013).

### 3.3.3.1 *Usabilidade:*

A aplicação web do *dashboard* de agricultura de precisão deve ser projetada com foco na usabilidade. É importante que seja intuitiva e de fácil utilização, mesmo para usuários sem conhecimentos avançados em tecnologia.

### 3.3.3.2 *Desempenho:*

O desempenho da aplicação é essencial para proporcionar uma experiência fluida aos usuários. Portanto, é fundamental que a plataforma seja responsiva e tenha tempos de

carregamento rápidos, permitindo que os usuários acessem e visualizem os dados de forma eficiente. Além disso, é necessário que a aplicação seja capaz de lidar com grandes volumes de dados e garantir um desempenho estável, mesmo durante períodos de pico de uso.

#### 3.3.3.3 *Segurança:*

A segurança dos dados é uma preocupação crucial em qualquer aplicação. No caso do *dashboard* de agricultura de precisão, é necessário garantir a proteção das informações confidenciais, como credenciais de login e dados coletados pelos sensores. Portanto, é fundamental implementar medidas de segurança adequadas, como autenticação e autorização robustas, para controlar o acesso aos dados e funcionalidades da aplicação.

#### 3.3.3.4 *Confiabilidade:*

A confiabilidade da aplicação é fundamental para garantir a coleta precisa e consistente dos dados dos sensores. É necessário que a aplicação seja altamente confiável e tenha uma disponibilidade elevada, permitindo que os dados sejam coletados de forma contínua e sem interrupções. Caso ocorram falhas ou interrupções, a aplicação deve ser capaz de se recuperar e retomar seu funcionamento normal de maneira adequada.

#### 3.3.3.5 *Escalabilidade:*

A aplicação do *dashboard* de agricultura de precisão deve ser dimensionada para lidar com o crescimento futuro. Isso inclui a capacidade de lidar com um aumento no número de usuários e no volume de dados. Além disso, é importante que a aplicação possa incorporar novos sensores e expandir sua capacidade conforme necessário, sem comprometer o desempenho e a funcionalidade. Além disso, a aplicação deve ser projetada para aproveitar os recursos de escalabilidade automática fornecidos pela plataforma de nuvem. Isso permite que a aplicação ajuste automaticamente sua capacidade de acordo com as demandas de tráfego e processamento, garantindo um desempenho consistente mesmo durante picos de uso.

#### 3.3.3.6 *Compatibilidade:*

Considerando a diversidade de dispositivos e navegadores web, é importante garantir a compatibilidade da aplicação. Ela deve ser capaz de funcionar adequadamente em diferentes

navegadores e dispositivos, proporcionando uma experiência consistente para os usuários, independentemente do ambiente em que estão utilizando a aplicação. Isso inclui a compatibilidade com diferentes sistemas operacionais e tamanhos de tela, incluindo dispositivos móveis.

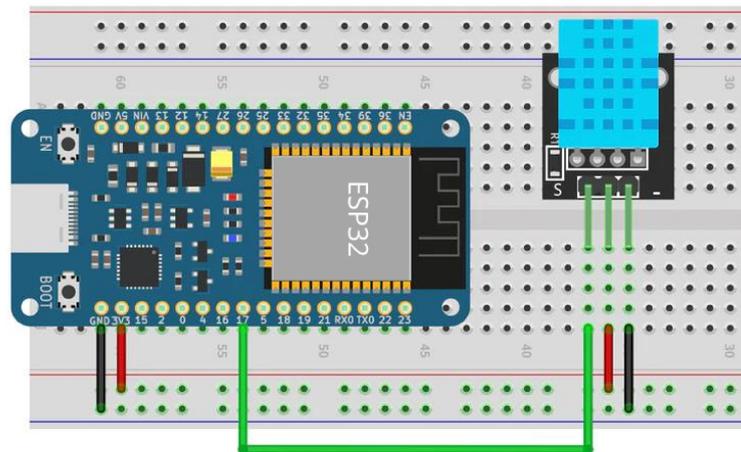
#### *3.3.3.7 Manutenibilidade:*

Para facilitar a manutenção e a incorporação de futuras atualizações e melhorias, é importante que a aplicação seja projetada e desenvolvida de forma modular e organizada. Isso permite realizar tarefas de manutenção, como atualizações de software e correções de falhas, de maneira mais eficiente e sem causar interrupções significativas no funcionamento da aplicação.

## 4 RESULTADOS

Durante a configuração do Gateway Node-RED, foram seguidos os passos recomendados na documentação oficial. Isso incluiu a instalação do Node-RED, a configuração inicial do servidor MQTT para comunicação com os microcontroladores e a criação de um fluxo que facilita o recebimento, processamento e encaminhamento dos dados para a aplicação web. O Gateway Node-RED mostrou-se eficiente no recebimento e processamento dos dados provenientes dos sensores conectados ao ESP32, o esquema do protótipo para coleta de dados é mostrado na Figura 5. A integração de nós MQTT simplificou a comunicação, enquanto os nós de processamento permitiram realizar transformações específicas nos dados antes de enviá-los para a aplicação web. A abordagem visual do Node-RED proporcionou uma visão clara do fluxo de dados. A utilização de nós pré-configurados facilitou significativamente o desenvolvimento, permitindo uma rápida prototipagem e ajustes conforme necessário.

Figura 5 – Esquema do protótipo para coleta de dados

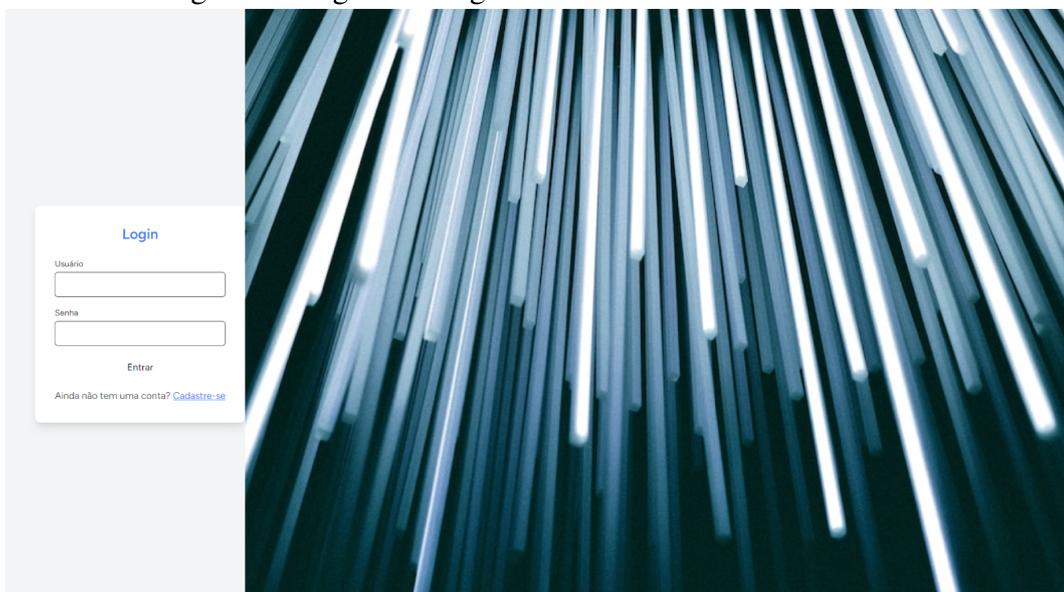


Fonte: o autor.

A aplicação web, mostrada nas Figuras 6 e 7, foi desenvolvida utilizando o Node-RED como base para criar um *dashboard* intuitivo. No processo de desenvolvimento, foram incorporadas tecnologias adicionais para aprimorar a experiência do usuário. O framework Inertia.js foi utilizado para facilitar a comunicação entre o Laravel e vue.js. A escolha do Inertia.js para facilitar a comunicação entre o Laravel e o Vue.js na aplicação web além de simplificar o desenvolvimento também proporcionou uma experiência de usuário mais ágil, eficiente e coesa. Essa abordagem moderna de comunicação entre back-end e front-end é fundamental para a construção de aplicações web robustas e de alto desempenho.

A estrutura do *dashboard* foi cuidadosamente elaborada para garantir uma apresentação intuitiva e organizada das informações. Foram empregadas técnicas de design para garantir que os elementos fossem dispostos de maneira lógica, facilitando a compreensão dos dados pelos usuários. Os dados coletados pelos sensores foram visualmente representados por meio de gráficos interativos fornecidos pelo Chart.js, uma biblioteca JavaScript de código aberto que permite aos desenvolvedores criar gráficos interativos para a web. Essa escolha permitiu uma visualização clara e personalizável dos dados. Os usuários podem selecionar diferentes intervalos de tempo e personalizar a exibição dos gráficos de acordo com suas preferências.

Figura 6 – Pagina de Login



Fonte: o autor.

Figura 7 – Dashboard



Fonte: o autor.

Para garantir uma experiência consistente em dispositivos móveis, o layout do *dash-*

*board* foi adaptado responsivamente, como ilustrado na Figura 8. Utilizando técnicas de design responsivo, a aplicação ajusta dinamicamente a apresentação dos elementos para se adequar a diferentes tamanhos de tela, proporcionando uma experiência otimizada, independentemente do dispositivo utilizado. Essas melhorias no processo de desenvolvimento e na escolha de tecnologias adicionais visam não apenas aprimorar a estética do *dashboard*, mas também a experiência geral do usuário, tornando a interação mais eficiente e agradável.

Figura 8 – Dashboard Dispositivos Móveis



Fonte: o autor.

A integração da aplicação web com o Google Cloud Platform (GCP) foi realizada de maneira fluida. A aplicação foi hospedada no Google App Engine, Serviços como Google Cloud Storage foram configurados para armazenar dados. As configurações foram ajustadas para garantir uma comunicação segura entre a aplicação e os serviços da GCP. O ambiente de nuvem foi configurado para escalabilidade automática, permitindo que a aplicação ajuste dinamicamente sua capacidade de acordo com as demandas de tráfego. A autenticação e autorização foram rigorosamente configuradas para garantir a segurança dos dados.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

A realização deste trabalho proporcionou avanços significativos na implementação de um sistema de agricultura de precisão, integrando dispositivos IoT, Node-RED e uma aplicação web baseada em vue.js. A organização do trabalho em cinco capítulos permitiu uma abordagem estruturada, abrangendo desde a introdução do tema até a apresentação dos resultados obtidos.

No capítulo de metodologia, foram detalhados os passos para o desenvolvimento do *gateway* Node-RED e da aplicação web, destacando a importância da integração eficiente entre os dispositivos de coleta e a plataforma de nuvem. A visão geral do projeto forneceu uma compreensão visual do fluxo de dados, destacando a interconexão entre os sensores, o *gateway* e a aplicação.

Ao abordar os resultados, observamos que o *gateway* Node-RED mostrou-se eficaz na recepção, processamento e encaminhamento dos dados coletados pelos sensores. A utilização de nós MQTT simplificou a comunicação, enquanto os nós de processamento proporcionaram flexibilidade na manipulação dos dados antes de sua apresentação na aplicação web.

A aplicação web, por sua vez, foi desenvolvida de forma a atender a requisitos funcionais e não funcionais específicos. A usabilidade foi priorizada, garantindo uma experiência intuitiva aos usuários. A escolha do Inertia.js facilitou a comunicação entre o Laravel e vue.js, contribuindo para um desenvolvimento eficiente e uma experiência de usuário ágil.

A apresentação dos resultados no *dashboard* demonstrou a efetividade da visualização dos dados coletados, utilizando gráficos interativos e uma organização lógica dos elementos. A adaptação responsiva do layout permitiu uma experiência consistente em dispositivos móveis.

A integração bem-sucedida com o Google Cloud Platform (GCP) foi um passo crucial, garantindo escalabilidade automática, segurança na comunicação e armazenamento eficiente dos dados.

Em conclusão, este trabalho proporcionou avanços significativos na implementação de um sistema completo de agricultura de precisão, abordando desde a coleta de dados até a apresentação eficiente desses dados em um *dashboard* acessível e intuitivo. As melhorias na usabilidade, desempenho, segurança e escalabilidade contribuem para a construção de um sistema robusto e eficaz. Considerando os resultados obtidos, sugere-se a continuidade do trabalho com possíveis melhorias no sistema, expansões e adição de novas funcionalidades, consolidando assim a contribuição deste trabalho para o campo da agricultura de precisão baseada em IoT.

## REFERÊNCIAS

- ABDALA, M. C. *et al.* Desenvolvimento de uma plataforma iot sob a arquitetura de computação em névoa para agricultura de precisão. Universidade Estadual do Oeste do Paraná, 2022.
- BASSOI, L. H.; INAMASU, R. Y.; BERNARDI, A. d. C.; VAZ, C. M. P.; SPERANZA, E. A.; CRUVINEL, P. E. Agricultura de precisão e agricultura digital. *TECCOGS*, n. 20, jul./dez., 2019., 2019.
- BELLO, S. A.; OYEDELE, L. O.; AKINADE, O. O.; BILAL, M.; DELGADO, J. M. D.; AKANBI, L. A.; AJAYI, A. O.; OWOLABI, H. A. Cloud computing in construction industry: Use cases, benefits and challenges. *Automation in Construction*, Elsevier, v. 122, p. 103441, 2021.
- BISONG, E.; BISONG, E. An overview of google cloud platform services. **Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners**, Springer, p. 7–10, 2019.
- CHÉOUR, R.; KHRIJI, S.; KANOUN, O. *et al.* Microcontrollers for iot: optimizations, computing paradigms, and future directions. In: IEEE. **2020 IEEE 6th World Forum on Internet of Things (WF-IoT)**. [S. l.], 2020. p. 1–7.
- COELHO, A. D.; DIAS, B. G.; ASSIS, W. de O.; MARTINS, F. de A.; PIRES, R. C. Monitoring of soil moisture and atmospheric sensors with internet of things (iot) applied in precision agriculture. In: IEEE. **2020 XIV Technologies Applied to Electronics Teaching Conference (TAEE)**. [S. l.], 2020. p. 1–8.
- COELHO, J. P. C.; SILVA, L. M. d.; PINHEIRO, A. C.; TRISTANY, M.; NETO, M. d. C. **Agricultura de precisão**. [S. l.]: Associação dos Jovens Agricultores de Portugal, 2009.
- COSENTINO, V.; IZQUIERDO, J. L. C.; CABOT, J. A systematic mapping study of software development with github. *IEEE Access*, IEEE, v. 5, p. 7173–7192, 2017.
- DELCEV, S.; DRASKOVIC, D. Modern javascript frameworks: A survey study. In: **2018 Zooming Innovation in Consumer Technologies Conference (ZINC)**. [S. l.: s. n.], 2018. p. 106–109.
- DUGUMA, A. L.; BAI, X. Contribution of internet of things (iot) in improving agricultural systems. *International Journal of Environmental Science and Technology*, Springer, 2023. Disponível em: <https://link.springer.com/article/10.1007/s13762-023-05162-7>.
- GÓMEZ-CHABLA, R.; REAL-AVILÉS, K.; MORÁN, C.; GRIJALVA, P.; RECALDE, T. Iot applications in agriculture: A systematic literature review. In: SPRINGER. **2nd International conference on ICTs in agronomy and environment**. [S. l.], 2019. p. 68–76.
- JOHANSSON, A. Why javascript is better than python. IEEE Computer Society, 2020. Disponível em: <https://www.computer.org/publications/tech-news/build-your-career/5-reasons-javascript-is-still-better-than-python/>.
- KAEWSANMUANG, K. Learn vue.js from the top 50 vue.js articles on medium. 2019. Disponível em: <https://medium.com/vue-mastery/the-top-50-vue-js-articles-of-2019-c8b6e2e1812e>.

KASSIM, M. R. M. Iot applications in smart agriculture: Issues and challenges. In: IEEE. **2020 IEEE conference on open systems (ICOS)**. [S. l.], 2020. p. 19–24.

KHANNA, A.; KAUR, S. Evolution of internet of things (iot) and its significant impact in the field of precision agriculture. **Computers and electronics in agriculture**, Elsevier, v. 157, p. 218–231, 2019.

LEKIĆ, M.; GARDAŠEVIĆ, G. Iot sensor integration to node-red platform. In: IEEE. **2018 17th International Symposium Infoteh-Jahorina (Infoteh)**. [S. l.], 2018. p. 1–5.

MONTERAIL. **Why Vue.js is the Best Choice for Your Next Web Project**. 2023. Disponível em: <https://www.monterail.com/vuejs-development-guide>.

MUNIZ, L. R. Internet das coisas na agricultura moderna: estudo da integração entre automação e sensoriamento no cultivo de frutos no nordeste brasileiro. 2021.

PEREIRA, C. R. **Aplicações web real-time com Node. js**. [S. l.]: Editora Casa do Código, 2014.

QUINTAM, C. P. R.; ASSUNÇÃO, G. M. de. Panorama do agronegócio exportador brasileiro. **RECIMA21-Revista Científica Multidisciplinar-ISSN 2675-6218**, v. 4, n. 7, p. e473642–e473642, 2023.

ROSAS, A. Enhancing laravel and inertia.js with typescript and vue 3's composition api to build a powerful spa. Laravel.io, 2023. Disponível em: <https://laravel.io/articles/enhancing-laravel-and-inertiajs-with-typescript-and-vue-3s-composition-api-to-build-a-powerful-spa>.

SADEEQ, M. M.; ABDULKAREEM, N. M.; ZEEBAREE, S. R.; AHMED, D. M.; SAMI, A. S.; ZEBARI, R. R. Iot and cloud computing issues, challenges and opportunities: A review. **Qubahan Academic Journal**, v. 1, n. 2, p. 1–7, 2021.

SHAH, H. Node. js challenges in implementation. **Global Journal of Computer Science and Technology**, 2017.

SHAH, J.; DUBARIA, D. Building modern clouds: Using docker, kubernetes google cloud platform. In: **2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)**. [S. l.: s. n.], 2019. p. 0184–0189.

SILVA, R. O. da; ARAUJO, W. M.; CAVALCANTE, M. M. Visão geral sobre microcontroladores e prototipagem com arduino. **Tecnologias Em Projeção**, v. 10, n. 1, p. 36–46, 2019.

SUNEHRA, D.; SRINIDHI, M. Implementation of smart urban farming using raspberry pi, arduino and node-red platform. In: IEEE. **2020 IEEE International Conference for Innovation in Technology (INOCON)**. [S. l.], 2020. p. 1–6.

TUKUR, M.; UMAR, S.; HASSINE, J. Requirement engineering challenges: A systematic mapping study on the academic and the industrial perspective. **Arabian Journal for Science and Engineering**, Springer, v. 46, p. 3723–3748, 2021. Disponível em: <https://link.springer.com/article/10.1007/s13369-020-05159-1>.

VALAREZO, R.; GUARDA, T. Comparative analysis of the laravel and codeigniter frameworks: For the implementation of the management system of merit and opposition competitions in the state university península de santa elena. In: **2018 13th Iberian Conference on Information Systems and Technologies (CISTI)**. [S. l.: s. n.], 2018. p. 1–6.

- VUE SCHOOL. **Vue.js Articles and Tutorials**. 2023. Disponível em: <https://vueschool.io/articles/>.
- VUE.JS. **Vue.js Community**. 2023. Disponível em: <https://vue-community.org/>.
- WIEGERS, K.; BEATTY, J. **Software requirements**. [S. l.]: Pearson Education, 2013.
- WIERUCH, R. **The road to react: Your journey to master plain yet pragmatic react. js**. [S. l.]: Robin Wieruch, 2017.
- XU, J.; GU, B.; TIAN, G. Review of agricultural iot technology. **Artificial Intelligence in Agriculture**, Elsevier, 2022.
- YADAV, N.; RAJPOOT, D. S.; DHAKAD, S. K. Laravel: A php framework for e-commerce website. In: **2019 Fifth International Conference on Image Information Processing (ICIIP)**. [S. l.: s. n.], 2019. p. 503–508.