

UM PROTÓTIPO DE SISTEMA DE MONITORAMENTO AMBIENTAL PARA AGRICULTURA

Fábio da Silva Souza¹, Paulo Henrique Lopes Silva²

RESUMO

A presença de conceitos e tecnologias disseminadas através da Internet das Coisas (IoT) tem se tornado cada vez mais comuns atualmente, impulsionada pela evolução das Tecnologias da Informação e Comunicação (TICs). Com isso, é perceptível a crescente disseminação e adoção de sistemas embarcados em várias aplicações. A utilização de dispositivos computacionais compactos capazes de se conectarem à internet, conferem mais versatilidade na manipulação de dados e no controle descentralizado. Tais avanços tecnológicos também ganharam espaço no meio agrícola, possibilitando uma gestão mais precisa e eficiente dos recursos agrícolas. Esse trabalho apresenta um protótipo de sistema que coleta, armazena e analisa dados cruciais para melhorar a gestão de dados agrícolas e seus resultados mediante sensores que forneçam essas informações em tempo real para prevenir perdas. O hardware do sistema é composto por um microcontrolador ESP32 que coleta dados de temperatura e umidade por meio de um sensor DHT11 gerenciados por um banco de dados em tempo real da plataforma *Firebase*.

Palavras-chave: Internet das coisas; Sistemas embarcados; Microcontroladores; Tempo real.

ABSTRACT

The presence of concepts and technologies disseminated through the Internet of Things (IoT) has become increasingly common today, driven by the evolution of Information and Communication Technologies (ICTs). As a result, the increasing dissemination and adoption of embedded systems in various applications is noticeable. The use of compact computing devices capable of connecting to the internet provides more versatility in data manipulation and decentralized control. Such technological advances have also gained ground in agriculture, enabling more precise and efficient management of agricultural resources. This work presents a prototype system that collects, stores and analyzes crucial data to improve the management of agricultural data and its results through sensors that provide this information in real time to prevent losses. The system hardware consists of an ESP32 microcontroller that collects temperature and humidity data through a DHT11 sensor managed by a real-time database on the *Firebase* platform.

Key words: Internet of Things; Embedded systems; Microcontrollers; Real time.

1 INTRODUÇÃO

A evolução de novas tecnologias computacionais e de automação, junto a consolidação de modelos de sistemas IoT (*Internet of Things*) e WSN (*Wireless Sensor Network*), motiva tanto a indústria como a academia a investir tempo e mão de obra na criação de mecanismos de integração dessas áreas visando desenvolver aplicações para monitoramento de ambientes e controle de dados de forma remota. Dessa forma, percebe-se um avanço significativo no projeto e desenvolvimento de microcontroladores e microprocessadores, tornando-os mais eficientes, menores e mais acessíveis, ligados a sensores para monitorar ambientes onde as condições climáticas são parâmetros importantes e exercem influência direta em determinadas atividades, sejam em ambientes industriais ou agrícolas (QUEIROZ *et al.*, 2016).

Manter o controle sobre tais condições proporciona redução de custos, elevação da produtividade, diminuição de impactos ambientais, aumento da segurança e eficiência, além de uma maior assertividade, causando impactos na qualidade e na quantidade dos produtos, melhorando assim a qualidade de vida do produtor rural, mantendo-o e sua família no meio rural, e fortalecendo outras atividades também, como o comércio e indústria, pois a renda gera prosperidade para outros ramos da economia (BASSO, 2021).

As constantes mudanças climáticas estão ameaçando diversos setores na agricultura, que é, particularmente, vulnerável devido à sua dependência dos fatores climáticos. Isso pode resultar em perdas significativas de safras e mudanças geográficas nas áreas de cultivo. No Brasil, onde a maioria das áreas

¹ Graduação em Ciência da Computação, Universidade Federal Rural do Semiárido (UFERSA).

² Doutorado em Engenharia Elétrica e de Computação, Universidade Federal do Rio Grande do Norte (UFRN).

agrícolas depende das variações naturais da chuva, o aumento da temperatura e as mudanças climáticas podem reduzir a disponibilidade de água, prejudicando ainda mais a capacidade de produção. A crescente frequência de secas, alagamentos, temperaturas extremas e a alteração de padrões de pragas e doenças são preocupações adicionais para o setor agrícola. Como resultado, os produtores rurais precisam se adaptar e buscar a melhor forma de novas tecnologias para manter um monitoramento e controle sobre tais mudanças climáticas (SILVA, 2023).

Basso (2021) discute a importância da tecnologia de geração distribuída no contexto da atividade rural, especificamente na região oeste do Paraná, Brasil. A atividade rural desempenha um papel crucial na economia dessas áreas, afetando tanto o meio rural quanto o urbano. Os desafios enfrentados pelos produtores rurais incluem questões como mudanças climáticas, aumento nos custos de produção e a necessidade de utilizar recursos de forma consciente.

Silva *et al.* (2017), descreve o desenvolvimento de um sistema de Internet das Coisas (IoT) para aquisição e monitoramento de dados de sensores. Os dados de sensores são lidos pelo microcontrolador (PIC18F2550) e enviados para um Raspberry Pi, que os envia para a plataforma *ThingSpeak* na internet. O projeto oferece a vantagem de suportar uma variedade de sensores e permite a expansão fácil. Além disso, os autores destacam o potencial de extensão do projeto para tomada de decisões automatizadas ou adaptação dinâmica dos sensores.

Mutha *et al.* (2016), apresenta um sistema de aquisição de dados em tempo real para monitoramento ambiental, utilizando um microcontrolador baseado em *Arduino-UNO* para implementar a estratégia de aquisição de dados e conectar sensores analógicos e digitais para monitorar temperatura, umidade etc.

Portanto, esse trabalho apresenta o desenvolvimento de um protótipo de sistema que visa proporcionar versatilidade na aquisição e armazenamento de dados de umidade e temperatura, que vão servir como base para monitoramento remoto em tempo real com foco para plantas e hortas. Para isso, a aplicação combina o uso de plataformas livres, como a placa NodeMCU e o ambiente de desenvolvimento integrado Arduino IDE (ARDUINO, 2023), com os recursos e a segurança da plataforma *Firebase* (FIREBASE, 2023) com banco de dados NoSQL, armazenado em tempo real usando serviços de computação em nuvem. O sistema captura as informações de temperatura e umidade por meio de um sensor DHT11 (MOUSER ELETRONICS, 2015) e o monitoramento será realizado via uma aplicação *web* desenvolvida em PHP (PHP, 2023), onde as informações estarão disponíveis na forma de gráficos.

Por se tratar de um sistema universal para sensores, supõe-se que o projeto possa ser estendido para outras linhas de pesquisa relacionadas, com a adição de outros sensores dependendo da necessidade. Foram encontrados outros trabalhos com objetivos similares, envolvendo a aquisição e monitoramento de dados de sensores para propósitos específicos ou genéricos. O artigo destaca a importância do monitoramento em tempo real do ambiente, especialmente em setores como agricultura e indústria, onde desvios dos parâmetros ambientais ideais podem resultar em perdas significativas de produtividade.

2 METODOLOGIA

Esta seção descreve os passos para o desenvolvimento do sistema proposto. O objetivo é fornecer uma compreensão clara de como foi desenvolvido e implementado o sistema. Ela contém uma breve revisão dos sensores e softwares utilizados e como estes encontram-se organizados para se comunicarem. Para tanto, o desenvolvimento desse sistema foi faseado em etapas distintas para garantir uma implementação precisa e eficiente do sistema, que são descritas brevemente no quadro 1.

Quadro 1. Etapas para implementação do sistema. (Autoria própria)

Etapas	Apresentação	Descrição dos passos
Primeira etapa	Seleção de uma arquitetura para o desenvolvimento do sistema, bem como seus componentes principais.	1 A escolha adequada dos sensores que podem desempenhar um papel fundamental na precisão e eficácia no monitoramento. Optamos por selecionar sensores específicos para medir as seguintes variáveis ambientais relevantes para a agricultura: 2 Temperatura e umidade do ar. 3 Umidade do solo. 4 A escolha do microcontrolador para controlar e processar os dados dos sensores. 5 A escolha de um banco de dados em tempo real para podermos armazenar os dados coletados de maneira segura.
Segunda etapa	Configuração Inicial do microcontrolador com os sensores.	6 Inicialmente, configuramos o microcontrolador para coletar dados dos sensores. Utilizamos as bibliotecas para garantir leituras precisas dos sensores. Os dados obtidos foram exibidos no <i>dashboard</i> da plataforma Arduino IDE para validação e análise inicial.
Terceira etapa	Sincronização de Tempo via NTP	7 Implementamos um serviço de sincronização de tempo via NTP para organizar os dados por data e hora. O microcontrolador, conectado à internet

		via <i>Wi-Fi</i> , fez requisições ao servidor NTP ("pool.ntp.org") para obter informações precisas de data e hora em tempo real.
Quarta etapa	Integração com o banco de dados.	8 Inicialmente, criamos um projeto no console do banco de dados. Obtivemos as informações de integração necessárias com a biblioteca para o envio de dados ao <i>real time database</i> criado e estabelecemos uma conexão à internet via <i>Wi-Fi</i> estável. 9 Na plataforma Arduino IDE, implementamos a funcionalidade de organização e envio dos dados para o banco de dados usando a biblioteca necessária para enviar os dados no formato correto, garantindo simplicidade, flexibilidade e eficiência no armazenamento.
Quinta etapa	Validação de Precisão e Confiabilidade	12 Foram realizados testes para verificar a precisão e confiabilidade das leituras dos sensores, bem como a estabilidade da conexão com o banco de dados e o tempo de comunicação entre os componentes, validando a aplicação por meio desses testes.
Sexta etapa	Desenvolvimento de uma aplicação web.	13 Configuração do Ambiente: definimos um ambiente de desenvolvimento local para web para executar scripts. 14 definimos um ambiente de desenvolvimento local para web para criar e testar o site antes de colocá-lo em um servidor de produção na internet. 15 Conexão com o Banco de Dados: estabelecemos conexão com o banco de dados a partir do servidor usando as SDKs (<i>Software Development Kit</i>) fornecidas no projeto do banco de dados. 16 Recuperação e apresentação dos dados: para a recuperação dos dados do consultamos a documentação do banco de dados, para criarmos funções que recuperam e manipulam os dados utilizando a partir de scripts. 17 Para o desenvolvimento da apresentação da página, criamos um layout bem simples para a página. Deixando com uma interface mais simples possível.

A metodologia adotada para desenvolver este sistema foi cuidadosamente planejada e executada em várias etapas distintas. A seleção criteriosa de sensores, o uso do microcontrolador e a integração com o banco de dados garantiram uma coleta de dados confiável. O desenvolvimento da aplicação web visa disponibilizar uma interface simples e intuitiva para os usuários acessarem e interpretarem os dados coletados. Essas etapas formaram uma base sólida para a implementação do sistema, demonstrando a eficácia da abordagem adotada.

3 DESENVOLVIMENTO

Esta seção aborda a apresentação e descrição da arquitetura do sistema, tal como seus principais componentes, desde o *hardware* até a parte de *software* e serviços em nuvem. Cada subseção aborda detalhadamente o funcionamento das partes do sistema, proporcionando uma compreensão abrangente do seu funcionamento.

3.1. Arquitetura e Organização do Sistema

O sistema é composto por partes em *hardware*, em *software* e por serviços em nuvem. Na parte de *hardware*, o DHT11 e o Sensor de Umidade do Solo Higrômetro (1) fazem leituras precisas de temperatura e umidade do ambiente. Estes sensores realizam a conversão de dados analógicos para digitais sobre estes dados e enviam para o ESP32 (2), que por sua vez, conectará à internet via Wi-Fi e fazendo solicitações de data e hora em tempo real a um servidor NTP (3), que é um protocolo de rede utilizado para sincronizar os relógios de computadores em uma rede a partir de servidores de tempo na Internet. Neste sistema optamos pelo servidor "pool.ntp.org", ele responde à requisição do ESP32 com os dados solicitados, garantindo que o dispositivo tenha uma referência de tempo precisa e confiável. Na figura 1, é possível observar o esquema do projeto.

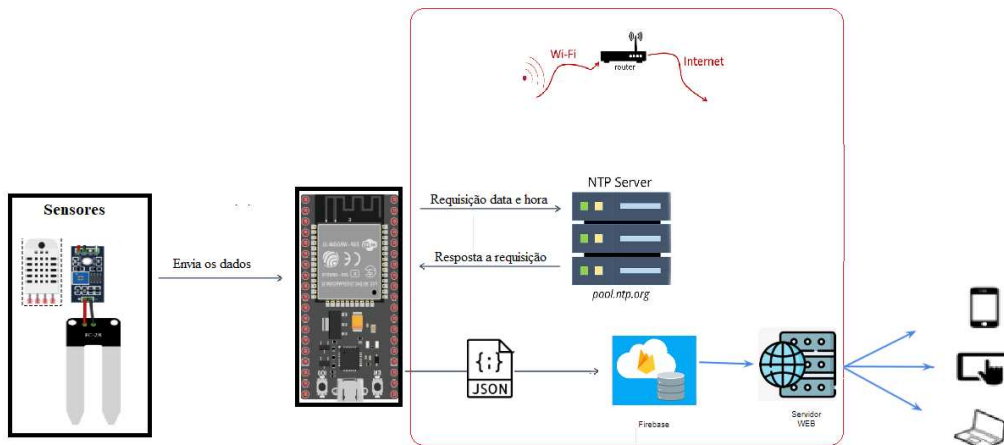


Figura 1. Esquema do projeto. (Autoria própria)

Na parte de *software* a biblioteca *ArduinoJSON* (BLANCHON, 2014) é a responsável por converter todas as leituras e dados recebidos em um formato padrão JSON. O ESP32 através de seu software embarcado faz o envio dos dados já formatados no padrão JSON (4) para o *Firebase* (5) através da conexão à internet via *Wi-fi*. O *Firebase* (5) atua como uma plataforma de armazenamento em tempo real hospedada em nuvem, permitindo que esses dados possam ser imediatamente acessados simultaneamente por diversas aplicações que tenham as permissões necessárias.

A plataforma *Firebase* conta com um *dashboard* onde são exibidos os dados armazenados. No entanto, a interpretação desses dados pode ser um desafio para usuários comuns devido à complexidade desses dados em estado bruto. Para melhorar a interação com os usuários, implementamos um servidor PHP (6) conectado ao *Firebase*, que acessa os dados armazenados e os processa para apresentação. Utilizando um navegador de preferência (7), os usuários podem visualizar os dados de forma clara e compreensível.

A Figura 2, mostra a parte de *hardware* do sistema em funcionamento. A alimentação dos circuitos dos sensores é fornecida pelo conector USB do ESP32 (A), que fornece uma tensão estável de 5,0V. Este fluxo de energia é essencial para o funcionamento adequado dos sensores e outros componentes conectados.

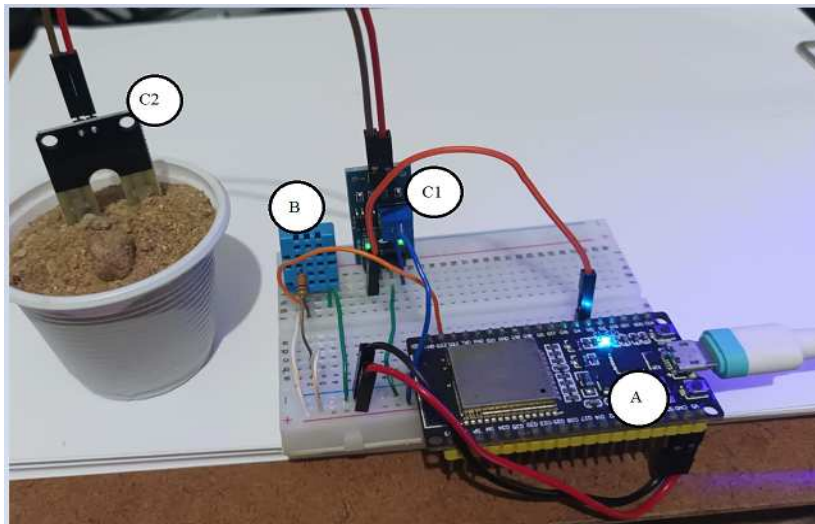


Figura 2. Sistema completo montado com protoboard. (Autoria Própria).

Com o auxílio de uma protoboard conectamos e organizamos o sensor DHT11 (B) e o sensor de umidade do solo Higrômetro (C1). O sensor de umidade do solo é composto por uma sonda (C2) que está inserida em um recipiente contendo areia, simulando as condições do solo. Para estabilizar a comunicação entre o sensor DHT11 e o microcontrolador utilizamos um resistor de 10k Ω (ohms) garantindo a precisão e confiabilidade das leituras. Esses dados, representativos das condições ambientais, são então capturados pelos

sensores e transmitidos ao ESP32.

3.2. Módulo ESP32

O ESP32 é um microcontrolador desenvolvido pela empresa chinesa *Espressif Systems* (FIREBASE, 2023). Ele é baseado no processador *Xtensa Dual-Core* de 32 bits e pode ser usado em diversos projetos de IoT (Internet das Coisas), robótica, automação residencial e outros projetos que envolvem conexão com a internet. É considerado um dos microcontroladores mais poderosos e versáteis disponíveis no mercado, especialmente em termos de conectividade, pois possui suporte para *Wi-Fi*, *Bluetooth 4.2* e *Bluetooth Low Energy* (BLE). Além disso, ele tem uma ampla variedade de interfaces periféricas, como SPI (*Serial Peripheral Interface*), I2C (*Inter-Integrated Circuit*), UART (*Universal Asynchronous Receiver/Transmitter*), PWM (*Pulse Width Modulation*), ADC (*Analog-to-Digital Converter*), entre outras, o que o torna muito versátil para diversos tipos de aplicações (USINAINFO, 2018).

As suas principais características são a presença de um microprocessador duplo onde um núcleo é destinado à comunicação e o outro é responsável pelo controle de portas de entrada e saída, evitando que alguns problemas comuns no ESP8266 se repetissem no ESP32, além de permitir assim a execução de várias tarefas simultaneamente, com maior eficiência e um baixo consumo de energia. Outras características relevantes são: alto desempenho devido ao processador com *clock* de até 240 MHz e uma GPU integrada, suporte a diversos protocolos como HTTP (RFC 9110), MQTT (RFC 9431), TCP/IP (RFC 1180) e outros (DATATRACKER, 2023). O ESP32 também se destaca por possuir memória flash integrada de até 4MB, o que permite o armazenamento de programas e dados, além de ter um preço bastante acessível (USINAINFO, 2018).

A programação do ESP32 é feita em linguagem de programação de alto nível, como C ou C++, usando o ambiente de desenvolvimento Arduino e suas várias bibliotecas e exemplos disponíveis para facilitar o desenvolvimento de projetos. Outro ambiente de desenvolvimento é a plataforma *Espressif IoT Development Framework* (ESP-IDF), que é o SDK oficial da *Espressif* para o ESP32. O ESP8266 e o ESP32 são microcontroladores com recursos de conectividade *Wi-Fi* e *Bluetooth* integrados. Embora possuam semelhanças, o ESP32 possui algumas vantagens em relação ao ESP8266. A Tabela 1, mostra uma comparação entre algumas das principais diferenças entre esses microcontroladores:

Tabela 1. Comparativo ESP8266 e ESP32. (Organizado pelo autor; adaptado de Usinainfo, 2018)

Especificação	ESP8266	ESP32
MCU	Xtensa® Single-Core 32-bit L106	Xtensa® Dual-Core 32-bit LX6 600 DMIPS
802.11 b/g/n Wi-Fi	Sim, HT20	Sim, HT40
Bluetooth	Não	Bluetooth 4.2 ou inferior
Frequência	80MHz	160MHz
SRAM	160 kBytes	512 kBytes
Flash	SPI Flash, até 16 mBytes	SPI Flash, até 16 mBytes
GPIO	17	36
Hardware/ Software PWM	Não / 8 canais	1 / 16 canais
SPI/ I2C/ I2S/ UART	2/ 1/ 2/ 2	4/ 2/ 2/ 2
ADC	10-bit	12-bit
CAN	Não	1
Interface Ethernet MAC	Não	1
Sensor Touch	Não	Sim
Sensor de Temperatura	Não	Sim
Temperatura de Trabalho	-40° C - 125° C	-40° C - 125° C

Essas características do ESP32 em relação ao ESP8266, demonstram seu maior poder computacional, tendo uma importante influência na sua escolha como microcontrolador para o sistema.

3.3. DHT11

O Sensor de Temperatura e Umidade DHT11 é um sensor similar ao DHT22, com a diferença de ter a faixa de medição de 0° a 50° Celsius e a umidade entre 20% e 90%. É um sensor integrado em um módulo de baixo custo que utiliza um termistor para medir a temperatura e um sensor capacitivo para medir a umidade do ambiente (MOUSER ELETRONICS, 2015).

O DHT11 funciona por meio de um sensor de umidade e um termistor para medir o ar circundante, todos enviando informações para um microcontrolador de 8 bits que responde com um sinal serial para outro microcontrolador. Ele trabalha com tensão de 3 a 5,5V e possui um baixo consumo de corrente, por volta de 0,5mA a 2,5mA, sem contar que sua precisão para medição de umidade chega a aproximadamente 5% RH e a precisão para temperatura chega por volta de 2°C, tendo um excelente custo-benefício (MOUSER ELETRONICS, 2015).

3.4. Sensor de Umidade do Solo Higrômetro

Este sensor detecta as variações de umidade no solo. Quando o solo ao redor dele está seco, ele emite um sinal de saída em estado alto e quando o solo está úmido, a saída do sensor muda para estado baixo. O limite entre seco e úmido pode ser ajustado através do potenciômetro presente no sensor, permitindo uma adaptação precisa às necessidades do seu projeto. O sensor também possibilita aproveitar a saída analógica A0 e conectá-la a um conversor AD, como o disponível em placas Arduino e outros microcontroladores, proporcionando um controle ainda mais preciso das condições do solo (ELETROGATE, 2018).

3.5. Google Firebase

O *Firebase* é uma plataforma digital utilizada para facilitar o desenvolvimento de aplicativos web ou *mobile*, de uma forma efetiva, rápida e simples, oferecida gratuitamente pela empresa Google. Ela disponibiliza serviços de armazenamento em nuvem, autenticação, banco de dados em tempo real, hospedagem, dentre outros (FIREBASE, 2023).

Este projeto utiliza o *Firebase Realtime Database* que é um banco de dados hospedado na nuvem que armazena dados em formato JSON (*JavaScript Object Notation*) (JSON, 2023). A Figura 3 mostra um exemplo de um modelo de um arquivo organizado no formato JSON, onde os dados organizados de forma estruturada usando pares de chave-valor. Assim, os dados são organizados em uma estrutura semelhante a uma árvore onde cada dado é armazenado como um par de chave-valor e cada nó (ou chave) possui um valor associado a ele, permitindo uma estrutura de dados flexível e dinâmica.

```
{
  "2023": {
    "Outubro": {
      "05": {
        "05:22:00": {
          "Temperatura": "20.99",
          "Umidade": "77"
        },
        "05:22:01": {
          "Temperatura": "20.99",
          "Umidade": "77"
        },
        "05:22:02": {
          "Temperatura": "20.99",
          "Umidade": "77"
        },
        "05:22:03": {
          "Temperatura": "20.99",
          "Umidade": "77"
        }
      }
    }
  }
}
```

Figura 3. Modelo de dados no formato JSON. (Autoria própria)

A plataforma também conta com conexão e sincronização em tempo real, fazendo assim com que todos os clientes conectados recebem automaticamente essas atualizações em tempo real, permitindo uma comunicação constante e instantânea com o banco de dados, além de oferecer segurança e controle de acesso que podem ser definidas no servidor para garantir que apenas usuários autorizados possam ler e escrever dados no banco de dados. Isso ajuda a manter a integridade dos dados e a privacidade dos usuários.

Para acessar os serviços do *Firebase* é necessário ter uma conta do Google (por exemplo, uma conta do Gmail). O *Firebase* possibilita a criação de múltiplos bancos de dados em tempo real, possibilitando que o desenvolvedor, quando logado, possa acessar múltiplos projetos distintos. Cada projeto é identificado por um ID exclusivo e conta com um painel que mostra ao usuário os dados armazenados em tempo real em estado bruto onde podemos adicionar, modificar ou remover os dados manualmente. Esse painel também fornece uma visão geral do projeto, incluindo estatísticas de uso, informações sobre usuários ativos, definições de regras de acesso e extensões de serviços que podem ser adicionados ao projeto. Na Figura 4, é possível visualizarmos como é apresentado o painel com algumas de suas principais funcionalidades.

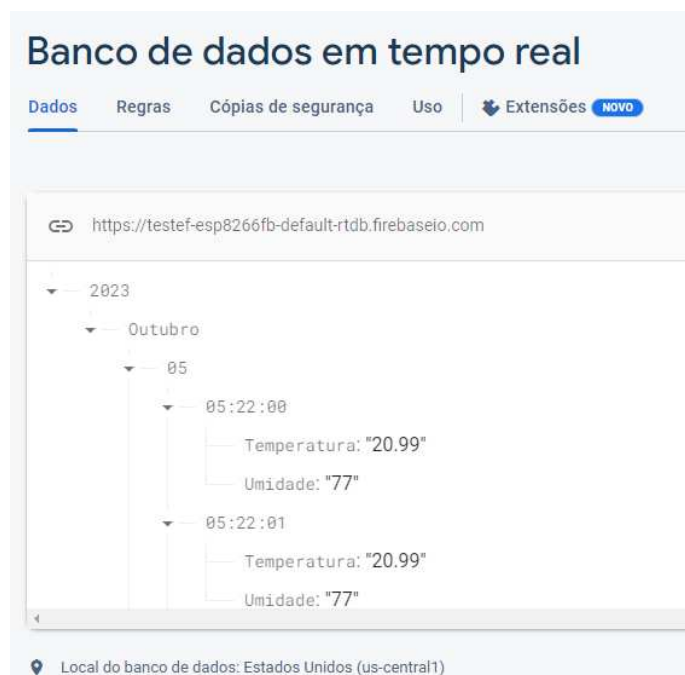


Figura 4. Painel de dados *Realtime Database*. (Dados de autoria própria armazenados no *Google Firebase*)

Um problema do uso do painel para o tratamento dos dados é como os dados são mostrados, que pode trazer ao usuário uma experiência muito desgastante quando houver muitos dados armazenados. Esse problema pode ser resolvido com o desenvolvimento de um aplicativo ou página *web* que possam melhorar a visualização dos dados.

A facilidade de integração também é uma característica do *Firebase*. Ele fornece SDKs (*Software Development Kit*) em várias linguagens de programação, como Java (JAVA, 2023), JavaScript, Swift, Objective-C, Node.js, entre outras, tornando a integração com aplicações bem acessível. A documentação do *Firebase* é bem simplificada e completa.

Na Figura 5, podemos visualizar o script de autenticação da SDK usado para integrar o projeto *Firebase* à aplicação *web*. É importante ressaltar que essas informações devem ser tratadas com confidencialidade, sendo acessíveis apenas às pessoas autorizadas a manipular o banco de dados, caso contrário qualquer pessoa que tenha esses dados poderá ter acesso às informações contidas no banco de dados.

```

<script type="module">
  // Import the functions you need from the SDKs you need
  import { initializeApp } from "https://www.gstatic.com/firebasejs/9.12.0/firebase-app.js";
  // TODO: Add SDKs for Firebase products that you want to use
  // https://firebase.google.com/docs/web/setup#available-libraries

  // Your web app's Firebase configuration
  const firebaseConfig = {
    apiKey: "AIzaSyB...",
    authDomain: "test-esp3201.firebaseio.com",
    databaseURL: "https://test-esp3201.firebaseio.com",
    projectId: "test-esp3201",
    storageBucket: "test-esp3201.appspot.com",
    messagingSenderId: "1010101010101",
    appId: "1:1010101010101:web:1010101010101"
  };

  // Initialize Firebase
  const app = initializeApp(firebaseConfig);
</script>

```

Figura 5. Código de autenticação da SDK do *Firebase*. (fonte *Firebase*)

O *Firebase*, por ser um produto da Google, se integra muito bem com outros produtos da empresa, como *Google Ads*, *Google Play*, *Google Marketing Platform*, entre outros, aprimorando ainda mais as funcionalidades que podem ser adicionadas em projetos ou aplicações (FIREBASE, 2023).

3.6 Arduino IDE e Bibliotecas utilizadas

A *Arduino IDE (Integrated Development Environment)* (ARDUINO, 2023) é um ambiente de desenvolvimento integrado utilizado para facilitar o desenvolvimento e a gravação de códigos diretamente no microcontrolador. A IDE do *Arduino* pode ser utilizada com as linguagens C/C++, além de uma compatibilidade com sistemas como *Windows*, *macOS* e *Linux*, tornando-a acessível a uma ampla variedade de usuários.

Arduino IDE fornece todas as ferramentas necessárias para escrever, compilar e fazer o upload de códigos, além de diversas bibliotecas padrão que fornecem funcionalidades prontas para uso. Além disso, os usuários podem criar suas próprias bibliotecas para reutilização de código (STRAUB, 2019). Para este projeto foram utilizadas as seguintes bibliotecas organizadas no quadro 2:

Quadro 2. Bibliotecas utilizadas no projeto (organizado pelo autor)

<i>IOXhop_FirebaseESP32</i> : que é uma extensão para a plataforma ESP32 que permite a integração fácil e eficaz entre projetos baseados em ESP32 e o serviço <i>Firebase Realtime Database</i> , oferecido pela Google. Essa biblioteca é a responsável por fazer a comunicação entre o circuito e o banco de dados em tempo real do <i>Firebase</i> (ARTRONSHOP, 2018).
<i>ArduinoJson</i> .5.13.3: é essencial para a comunicação eficaz entre dispositivos <i>Arduino</i> , como o ESP32, e o <i>Firebase</i> . O <i>Firebase</i> geralmente retorna dados em formato JSON (<i>JavaScript Object Notation</i>), que é leve e fácil de entender para troca de dados. Ela simplifica a manipulação de dados JSON, formatando assim os dados para serem enviados para o <i>Firebase</i> (BLANCHON, 2014).
<i>DHT-sensor-library</i> : simplifica a leitura de dados de sensores de temperatura e umidade como o DHT11 no <i>Arduino</i> . Ela tem funções que facilitam a comunicação com o sensor, permitindo aos desenvolvedores lerem facilmente os valores de temperatura e umidade (ADAFRUIT, 2011).
A biblioteca <i>Time</i> é uma ferramenta fundamental para trabalhar com temporização e manipulação de tempo. O código é derivado da biblioteca <i>Playground DateTime</i> , mas é atualizado para fornecer uma API mais flexível e fácil de usar. essa biblioteca pode ser usada para projetos como um relógio em tempo real, serviço de horário NTP da Internet, dados de horário GPS e mensagens de horário serial de um computador para sincronização de horário. Ela permite aos desenvolvedores obter informações precisas sobre a hora, minuto, segundo, dia, mês e ano. Além disso, facilita a conversão entre formatos de tempo (STOFFREGEN, 2014).

O uso dessas bibliotecas não apenas proporcionou uma economia de tempo no desenvolvimento, mas

também garantiu a confiabilidade e a eficiência do sistema, contribuindo significativamente para o sucesso do projeto.

3.7. Aplicação WEB

A aplicação web do sistema foi cuidadosamente desenvolvida com o objetivo principal de proporcionar uma visualização clara e compreensível dos dados coletados pelos sensores, que são armazenados de forma segura no *Firebase*. Isso permite que o usuário final, especialmente agricultores, monitore suas plantações de forma fácil e rápida. Optamos por utilizar PHP como linguagem do lado do servidor. Ao integrar PHP com HTML e CSS, criamos um *layout* simples e responsivo, garantindo que os dados sejam exibidos de maneira clara e acessível. Estabelecemos uma conexão segura com o banco de dados *Firebase* por meio da SDK, simplificando a integração da aplicação com a plataforma e assegurando a integridade e segurança dos dados.

Para lidar com os dados armazenados no *Firebase*, desenvolvemos um script utilizando diversas técnicas e métodos de programação. Dado que os dados do *Firebase* são organizados no formato JSON, utilizamos a recursividade para lidar com dados aninhados. Isso significa que, quando um objeto (ou chave) contém um ou mais objetos aninhados, nossa função é chamada novamente para processar esses objetos, garantindo uma manipulação eficaz e precisa dos dados. A Figura 6, mostra a visualização da página principal, que foi inicialmente executada localmente, onde será postada em um domínio para que o agricultor consiga o acesso pelo navegador da web em seu dispositivo (computador, tablet ou smartphone).

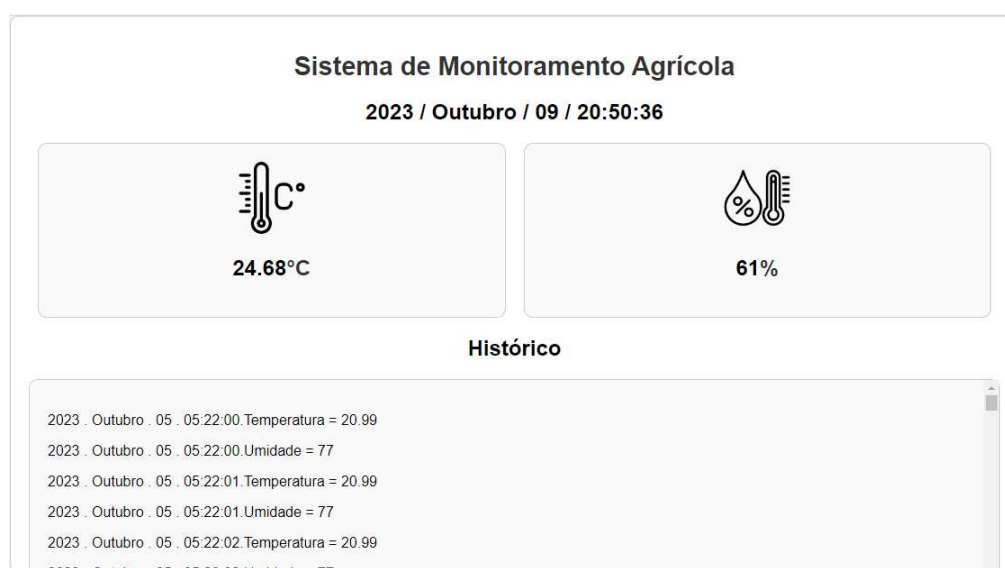


Figura 6. Front-end da aplicação web (Imagem criada pelo autor usando Apache Netbeans IDE)

Após acessar o site, o agricultor é direcionado para a página principal da aplicação, essa é a página onde ele pode visualizar os dados em tempo real provenientes dos sensores de temperatura e umidade em tempo real, como mostrado na Figura 6, permitindo uma análise instantânea das condições de sua plantação. Esses dados são atualizados automaticamente à medida que são coletados pelos sensores, proporcionando uma visão precisa e em tempo real do ambiente agrícola. Além disso, o agricultor pode visualizar um histórico de registros passados, possibilitando uma análise comparativa ao longo do tempo, crucial para compreender padrões climáticos e tendências que podem afetar o crescimento das plantas. Ao seguir os passos mencionados acima, o agricultor pode facilmente acessar informações vitais sobre suas plantações, permitindo uma melhor tomada de decisões e uma gestão agrícola mais eficiente e produtiva.

4 CONCLUSÕES

Diante dos desafios enfrentados pelos produtores rurais, como as mudanças climáticas e a necessidade de utilizar recursos de forma consciente, visando a eficiência e a redução de custos, desenvolvemos um sistema de monitoramento de dados agrícolas em tempo real. Durante o desenvolvimento deste projeto, enfrentamos diversos desafios técnicos, desde a escolha dos sensores apropriados, a integração eficaz com a plataforma *Firebase* em nuvem, até a aplicação web que exibe os dados armazenados. Problemas com componentes da parte de hardware também aumentaram a dificuldade deste desafio, como o rompimento de um dos pinos do Sensor de Umidade do Solo Higrômetro causado por problemas externos.

Já na final do projeto nos deparamos com um erro na memória flash da placa ESP32, onde ao tentarmos enviar as instruções de códigos para a placa era nos apresentado o seguinte erro: “*md5 of file does not match data in flash!*”, buscamos de várias maneiras resolver o problema o quanto antes possível, tentamos trocar o cabo

de alimentação da placa, executar o código em outro computador, executar outros códigos mais simples, resetar a placa para padrão de fábrica, entre outras tentativas frustradas, tivemos que substituir o ESP32 por um outro.

Durante os testes de sistemas realizados para verificar a estabilidade da conexão com o *Firebase*, a precisão e confiabilidade das leituras dos sensores e a validação completa do sistema para garantir que atenda aos requisitos especificados, foram obtidos resultados significativos. O sistema desenvolvido mostrou-se capaz de coletar, armazenar e apresentar dados de temperatura e umidade do ar, bem como umidade do solo, em tempo real. Utilizando o poder computacional do ESP32 e a plataforma *Firebase*, conseguimos fornecer aos dados precisos e atualizados sobre as condições ambientais do ambiente de testes. Estes testes foram cruciais para validar a eficácia do sistema proposto.

Em relação às motivações iniciais, os resultados demonstraram a viabilidade e a eficácia do sistema proposto. As funcionalidades de monitoramento em tempo real atenderam às expectativas, proporcionando uma visão detalhada e imediata das condições ambientais. Vale ressaltar a versatilidade do sistema, permitindo sua expansão para outras áreas de pesquisa relacionadas com o apoio de outros sensores, mostrando potencial para futuras inovações.

Diante desses resultados, nossos objetivos neste estudo foram alcançados, destacando-se a criação de um sistema robusto e acessível que permite aos agricultores monitorarem suas plantações de forma eficiente. As novidades apresentadas neste estudo consistem na integração eficaz de tecnologias diversas, como sensores, microcontroladores e computação em nuvem, para criar uma solução prática e útil para o setor agrícola.

Como trabalhos futuros, vislumbra-se a implementação de funcionalidades adicionais, como alertas automáticos baseados nos dados coletados, permitindo que os agricultores sejam notificados sobre condições adversas, mais funcionalidades no serviço web, como filtro de pesquisa, utilização de tabelas e gráficos interativos. Além disso, a evolução do sistema para incluir mais sensores, sensores mais potentes e mais variáveis agrícolas a serem monitoradas poderia fornecer uma visão ainda mais abrangente das condições da plantação.

5 REFERÊNCIAS BIBLIOGRÁFICAS

[1] ADAFRUIT. **DHT Sensor Library**. 2011. Disponível em: <https://github.com/adafruit/DHT-sensor-library>. Acesso em: 27 de jul. 2023.

[2] APACHE XAMPP. 2023. Disponível em: https://www.apachefriends.org/pt_br/index.html. Acesso em: 7 de out. de 2023.

[3] APACHE NETBEANS. 2023. Disponível em: <https://netbeans.apache.org/>. Acesso em: 7 de out. de 2023.

[4] ARDUINO. Arduino. **IDE**. Disponível em: <https://www.arduino.cc/en/software>. Acesso em: 3 de outubro de 2023.

[5] ARTRONSHOP. 2018. Disponível em: https://github.com/ArtronShop/IOXhop_Firebase_ESP32. Acesso em: 8 de jun 2023.

[6] BASSO, Ângela Meier. **Geração Distribuída na Atividade Rural**. 2021. 13f. Trabalho de conclusão de curso (Especialização em Integração Paraguai-Brasil) - Universidade Federal da Integração Latino - Americana (UNILA). p.1-13, 2021.

[7] BLANCHON, Benoît. **ArduinoJson**. 2014. Disponível em: <https://github.com/bblanchon/ArduinoJson>. Acesso em: 8 de jun 2023.

[8] DATATRACKER. **Internet Engineering Task Force Data Tracker**. Disponível em: <https://datatracker.ietf.org/doc/>. Acesso em: 5 de out. de 2023.

[9] ELETROGATE. **Módulo Sensor de Umidade de Solo**. Eletrogate, 2018. Disponível em: <https://www.eletrogate.com/modulo-sensor-de-umidade-de-solo>. Acesso em: 12 de jul. 2023.

[10] FIREBASE. **Firestore Documentation**. **Firestore 2023**. Disponível em: <https://firebase.google.com/docs?hl=pt-br>. Acesso em: 13 de jun. 2023.

[11] JAVA. Java. **A Linguagem de Programação Número Um**. Disponível em: <https://www.java.com/pt-BR/>. Acesso em: 1 de out. de 2023.

[12] JSON. **The JSON Data Interchange Standard**. Disponível em: <https://www.json.org/json-en.html>. Acesso em: 1 de out. de 2023.

- [13] MOUSER ELETRONICS. **DHT11 Technical Data Sheet**. 2015. Disponível em: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>. Acesso em: 3 de out. de 2023.
- [14] MUTHA, Vasant Raj; KUMAR, Narendra; PAREEK, Parikshit. **Real time standalone data acquisition system for environmental data**. International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), p. 1–4, 2016.
- [15] PHP. PHP: **Hypertext Preprocessor**. Disponível em: <https://www.php.net/>. Acesso em: 3 de out. de 2023.
- [16] QUEIROZ, T. A.; DIAS, D. L.; ARAUJO, P. H.; FIGUEIREDO, R. P.; JUCA, S. C. **Sistema embarcado linux para análise de sensores de temperatura dht11 e lm35**. 2016. Disponível em: <https://docplayer.com.br/65840334-Sistema-embarcado-linux-para-analise-de-sensores-de-temperatura-dht11-e-lm35.html>. Acesso em: 9 ago. 2023.
- [17] SILVA, A. A. M.; COSTA, L. S.; JUCA, S. C. S.; SILVA, P. M. M. **Sistema IoT versátil para monitoramento em nuvem de sensores**. Disponível em: https://www.researchgate.net/publication/335590433_Sistema_IoT_versatil_para_monitoramento_em_nuvem_de_sensores. Acesso em: 13 jun. 2023.
- [18] SILVA, João Lucas. **Como as mudanças climáticas afetam a agricultura no Brasil?** 2023. Disponível em: <https://www.gecal.com.br/como-as-mudancas-climaticas-afetam-a-agricultura-no-brasil>. Acesso em: 18 set. 2023.
- [19] STOFFREGEN, Paul. **Time**. GitHub. 2014. Disponível em: <https://github.com/PaulStoffregen/Time>. Acesso em: 7 ago 2023.
- [20] STRAUB, Matheus Gebert. **Arduino IDE**: o software para gravação de códigos no Arduino. Usinainfo, 2019. Disponível em: <https://www.usinainfo.com.br/blog/arduino-ide-o-software-para-gravacao-de-codigos-no-arduino/>. Acesso em: 25 set. 2023.
- [21] USINAINFO. **ESP32**: o que é, características e aplicações. 2018. Disponível em: <https://www.usinainfo.com.br/esp32-611>. Acesso em: 13 jun. 2023.