

Automação e Monitoramento Remoto de Silos Agrícolas utilizando ESP32 e Sensor Ultrassônico

Atrius Santos Luz, 061220007@faculdade.cefsa.edu.br, FESA
Gustavo Pereira de Oliveira Gonçalves, 061220023@faculdade.cefsa.edu.br, FESA
Pedro de Paula Valentinuci, 061220048@faculdade.cefsa.edu.br, FESA
Victor Inácio de Oliveira, pro@14723.cefsa.edu.br, FESA

Resumo

Este trabalho apresenta o desenvolvimento e a implementação de um sistema de monitoramento remoto para silos de grão, visando otimizar o gerenciamento de recursos agrícolas. O crescente avanço da tecnologia e da inovação no setor agrícola impulsiona a busca por soluções que melhorem a precisão e a eficiência na gestão de insumos. O estudo foi fundamentado em conceitos de Internet das Coisas (IoT), sensores e microcontroladores, com foco na aplicação prática da Engenharia de Controle e Automação para resolver desafios reais. O usuário final deve contar com visualização dos dados em um Dashboard para melhor interação. Os resultados demonstram a capacidade do sistema em prover monitoramento preciso, reforça como soluções tecnológicas podem transformar o gerenciamento de estoques agrícolas, confere maior praticidade, controle e baixo custo.

Palavras-chave: Monitoramento. Agroindustrial. IoT

Introdução

O agronegócio, um setor de vital importância econômica e social globalmente, demanda cada vez mais a incorporação de soluções tecnológicas para otimizar suas operações e garantir a segurança alimentar. A gestão eficaz de estoques de grãos em silos é crucial para preservar a qualidade do produto e mitigar perdas significativas causadas por fatores como umidade, temperatura e pragas. Nesse cenário de crescente digitalização e automação industrial, a Indústria 4.0, que se caracteriza pela integração e controle da produção por meio de sistemas inteligentes e interconectados, tem a Internet das Coisas (IoT) como um de seus pilares fundamentais. A aplicação da IoT na agricultura, por sua vez, permite a integração de sensores em rede para coletar e analisar dados em tempo real, otimizando o manejo e aumentando a eficiência operacional.

Nesse contexto, a Fundação Engenheiro Salvador Arena, em sua propriedade rural em Santa Rita do Passa Quatro, enfrenta desafios na gestão de seus silos agrícolas. Atualmente, os sistemas de estocagem de insumos e produtos operam de forma puramente analógica, desprovida de qualquer automação, o que acarreta significativa perda de tempo e redução da eficiência produtiva. A ausência de um sistema de monitoramento adequado pode comprometer a logística e o planejamento, destacando a importância de desenvolver uma solução que realize medições precisas e em tempo real para evitar perdas ou desclassificação de produtos, como abordado por Ferreira (2024).

Diante dessa problemática, o presente trabalho tem como objetivo principal desenvolver e implementar um sistema de automação e monitoramento remoto para silos agrícolas, visando otimizar a gestão de recursos, melhorar a precisão e a eficiência na administração de insumos. O sistema proposto

busca fornecer ao usuário final uma visualização clara dos dados em um *dashboard* para melhor interação e controle, objetivo este também evidenciado por Ferreira (2024).

Figura 1 – Demonstração visual do sistema atual do silo e proposta de implementação
Fonte: Autoria própria (2024)



Para alcançar esse objetivo, o estudo foi fundamentado em pilares tecnológicos essenciais para a Engenharia de Controle e Automação e para o desenvolvimento de sistemas inteligentes:

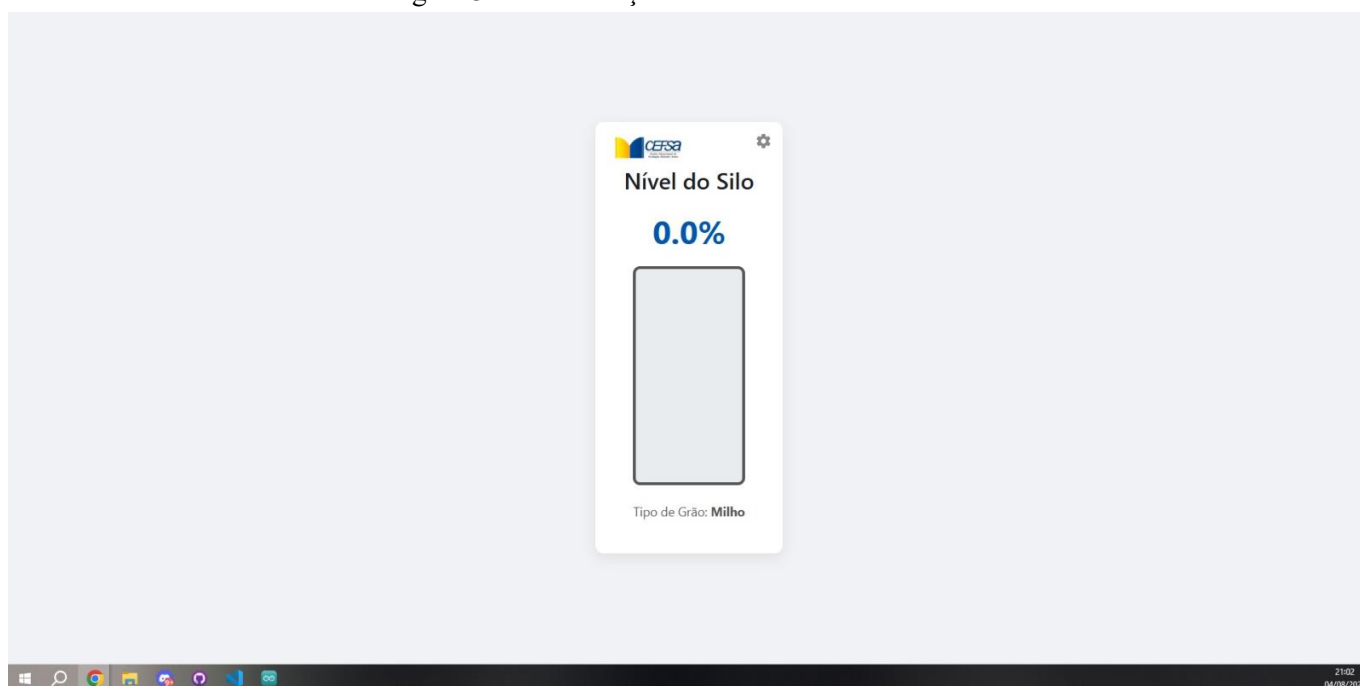
- **Internet das Coisas (IoT):** A IoT é conceituada como um sistema de dispositivos de computação inter-relacionados, que recebem identificadores únicos e a capacidade de transferir dados através de uma rede sem exigir interação entre homem-homem ou homem-máquina. O conceito foi cunhado por Kevin Ashton em 1999, e o sistema foi posteriormente denominado "Internet das Coisas" por Santos et al. (2017). Essa conectividade permite o monitoramento remoto e o controle de variáveis em tempo real, mesmo em ambientes remotos, facilitando o acesso e a transferência de dados entre máquinas. A instrumentação, por meio da alta conectividade e rápida comunicação da IoT, possibilitou avanços significativos, como na Indústria 4.0.

- **Microcontroladores (ESP32):** Para a aquisição e processamento de dados, o microcontrolador ESP32 foi escolhido devido ao seu baixo custo, versatilidade e capacidades integradas de Wi-Fi e Bluetooth, sendo amplamente utilizado para o desenvolvimento de dispositivos IoT, conforme relatado por Ichsanuddin (2022), Fathoni (2023), Rombekila (2022) e Alves Costa (2019). Oliveira (2017) destaca o ESP32 como parte da família de microcontroladores ARM, e a Espressif (2019, 2022) detalha suas especificações. O ESP32 pode ser programado através da Arduino IDE utilizando linguagens como C/C++, e oferece funcionalidades como o modo "deep sleep" para otimização do consumo de energia em aplicações de baixo consumo, conforme detalhado por Silva et al. (2014) e Alves Costa (2019).

- **Sistemas Supervisórios e Interfaces:** A visualização e interação com os dados coletados são cruciais para o usuário final. Sistemas supervisórios, como o Blynk ou ScadaBR, e interfaces web baseadas

em conceitos de *back-end* e *front-end*, utilizando APIs RESTful e o formato JSON para o tráfego de informações, permitem o acesso e a interação com os dados do silo em tempo real, facilitando a tomada de decisões estratégicas e o planejamento de ações de manutenção. Santos et al. (2021) definem sistemas supervisórios como SCADA, que rastreiam e manipulam dados de processos, enquanto Coelho (2009) destaca a agilidade na detecção de problemas. A implementação de interfaces web com *back-end* e *front-end* e o uso de APIs RESTful e JSON para troca de informações é amplamente discutida por Alves Costa (2019) e Sucierto et al. (2024).

Figura 3 – Visualização do nível do silo em dashboard



Fonte: Autoria própria (2025)

Este trabalho busca, portanto, demonstrar a viabilidade de transformar o gerenciamento de estoques agrícolas através de soluções tecnológicas, conferindo maior praticidade, controle e baixo custo.

Metodologia

O desenvolvimento deste sistema de automação e monitoramento remoto de silos agrícolas foi conduzido sob a abordagem de pesquisa aplicada, com foco no desenvolvimento de um protótipo funcional para resolver um problema real. A arquitetura do sistema compreendeu módulos de hardware e software interconectados, conforme detalhado nas subseções seguintes.

O sistema de hardware foi projetado para assegurar a medição precisa do nível do silo e a comunicação dos dados. O principal componente de sensoriamento é um sensor de nível ultrassônico de haste, selecionado por sua precisão e adequação para operação em ambientes industriais, incluindo aqueles

com potencial de gases ou poeiras inflamáveis. Este sensor gera um sinal de saída no padrão industrial de 4 a 20 mA, que representa linearmente o nível medido dentro do silo.

Para a interface entre o sensor e o microcontrolador, foi desenvolvida uma Placa de Circuito Impresso (PCI) customizada. Esta PCI foi dimensionada especificamente para converter o sinal de corrente de 4 a 20 mA para um sinal de tensão compatível com as entradas analógicas do microcontrolador. O design da placa otimizou o uso de espaço, permitindo que a PCI fosse perfeitamente acoplada ao microcontrolador e acondicionada em uma caixa protetora.

Figura 4 – ESP32 utilizado na placa



Fonte: AZ-Delivery (2025)

O ESP32 foi escolhido como o microcontrolador central do sistema devido à sua capacidade de processamento, conectividade Wi-Fi embarcada e múltiplas portas de entrada/saída (GPIOs) com funcionalidade de Conversor Analógico-Digital (ADC). O conjunto eletrônico, composto pela PCI e o ESP32, foi integrado em uma caixa protetora robusta. Esta caixa foi equipada com um fusível de proteção para salvaguardar o sistema contra sobrecargas elétricas. Para monitoramento visual do status operacional, dois indicadores luminosos (LEDs) foram incorporados: um LED verde, para indicar que o sistema está ligado e conectado à rede Wi-Fi, e um LED vermelho, para sinalizar eventuais falhas ou perda de conexão. A alimentação do módulo é provida por uma entrada de energia de dois pinos, conectada a um transformador externo que fornece 5V à PCI e ao ESP32. Adicionalmente, dois terminais dedicados foram previstos na caixa para a conexão dos fios do sensor de nível, garantindo uma interface elétrica segura e padronizada.

A arquitetura do software é centrada em uma estrutura de servidor web assíncrono que gerencia tanto a interface do usuário quanto a lógica de aquisição e armazenamento de dados. O fluxo de operação é dividido em duas fases principais: a inicialização (setup, linha 121) e o ciclo de operação contínua (loop, linha 169).

O servidor web do ESP32 opera com base em um conjunto de rotas (endpoints) que respondem a requisições específicas do navegador. A comunicação é dividida entre servir os arquivos estáticos da

interface e fornecer dados dinâmicos através de uma API. As rotas são definidas na função setup (a partir da linha 154).

As rotas DE Arquivos Estáticos são responsáveis por entregar os arquivos que compõem a interface visual. A função `handleFileRead` (linha 23) gerencia todas elas, identificando o tipo de arquivo pela sua extensão.

- **GET /:** Rota principal. Serve o arquivo da página inicial, `index.html`.
- **GET /style.css:** Serve o arquivo de folha de estilos `style.css`.
- **GET /script.js:** Serve o arquivo JavaScript principal da página inicial.
- **GET /history:** Serve a página de histórico, `history.html`.
- **GET /history.js:** Serve o script da página de histórico, responsável por renderizar o gráfico.
- **GET /logo.png:** Serve a imagem do logo da instituição.

Já as APIs de dados dinâmicos permitem a comunicação de dados entre a interface e o ESP32.

- **GET /nivel:** Fornece o nível atual do silo. A função `handleNivel` (linha 41) lê o sensor, converte o valor para porcentagem e responde com texto puro (ex: "75.3"). É chamado a cada 2 segundos pelo `script.js`.
- **GET /get-config:** Fornece a configuração atual. A função `handleGetConfig` (linha 52) lê o "Tipo de Grão" da memória e responde com um objeto JSON (ex: `{"tipoGrao":"Milho"}`). É chamado no carregamento da página.
- **POST /save-config:** Salva uma nova configuração. A função `handleSaveConfig` (linha 63) recebe o novo "Tipo de Grão" do formulário, salva-o na memória flash usando `Preferences` e confirma a operação.
- **POST /reset-wifi:** Inicia o processo de reconfiguração do Wi-Fi. A função `handleWifiReset` (linha 78) apaga as credenciais salvas e reinicia o ESP32, forçando a entrada no portal do `WiFiManager`.
- **GET /get-history:** Fornece os dados históricos. A rota utiliza a função `handleFileRead` (linha 23) para ler e enviar o conteúdo completo do arquivo `history.json`.

Figura 5 – Void Setup

```
void setup() {  
  Serial.begin(115200);  
  
  pinMode(pinoLedSensor, OUTPUT);  
  pinMode(pinoLedWifi, OUTPUT);  
  digitalWrite(pinoLedSensor, LOW);  
  digitalWrite(pinoLedWifi, LOW);  
  
  if(!SPIFFS.begin(true)){ Serial.println("Erro ao montar o SPIFFS"); return; }  
  
  preferences.begin("silo-config", false);  
  tipoGraoAtual = preferences.getString("tipoGrao", "Grão não definido");  
  preferences.end();  
  
  wm.setHostname(hostName);  
  wm.setConfigPortalTimeout(180);  
  
  if (!wm.autoConnect("Silo-Config-AP")) {  
    ESP.restart();  
  }  
  
  digitalWrite(pinoLedWifi, HIGH);  
  
  configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);  
  
  if (!MDNS.begin(hostName)) { Serial.println("Erro ao configurar o mDNS!"); }  
  
  Serial.println("\n-----");  
  Serial.println("Wifi Conectado!");  
  Serial.print("IP: "); Serial.println(WiFi.localIP());  
  Serial.print("Acesse pelo nome: http://"); Serial.print(hostName); Serial.println(".local");  
  Serial.println("-----");  
  
  ws.onEvent(onWsEvent);  
  server.addHandler(&ws);  
  
  // Rotas do servidor  
  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){ request->send(SPIFFS, "/index.html", "text/html"); });  
  server.on("/style.css", HTTP_GET, [](AsyncWebServerRequest *request){ request->send(SPIFFS, "/style.css", "text/css"); });  
  server.on("/script.js", HTTP_GET, [](AsyncWebServerRequest *request){ request->send(SPIFFS, "/script.js", "application/javascript"); });  
  server.on("/history", HTTP_GET, [](AsyncWebServerRequest *request){ request->send(SPIFFS, "/history.html", "text/html"); });  
  server.on("/history.js", HTTP_GET, [](AsyncWebServerRequest *request){ request->send(SPIFFS, "/history.js", "application/javascript"); });  
  server.on("/get-history", HTTP_GET, [](AsyncWebServerRequest *request){ request->send(SPIFFS, "/history.json", "application/json"); });  
  server.on("/logo.png", HTTP_GET, [](AsyncWebServerRequest *request){ request->send(SPIFFS, "/logo.png", "image/png"); });  
}
```

Fonte: Autoria própria (2025)

Resultados e discussão

Após os testes realizados com a fonte de bancada para simular o sinal elétrico enviado pelo sensor ultrassônico, foi possível confirmar a eficácia da leitura dos dados e a conversão do sinal analógico em digital. O sinal representou com precisão a medição de 0% a 100%, conforme esperado, e a integração com o servidor permitiu o monitoramento em tempo real do nível atual por meio de diferentes dispositivos, tanto em plataformas mobile quanto desktop. Os testes foram conduzidos em até quatro dispositivos simultaneamente, demonstrando que, em um contexto como o de um silo de grãos em uma

fazenda, diversos técnicos podem acessar e analisar os dados medidos, desde que estejam conectados à mesma rede.

A aplicação da solução em um ambiente real promete tornar o processo de manutenção e controle dos dados relacionados ao armazenamento de grãos significativamente mais seguro e ágil. Isso é especialmente relevante em ambientes rurais, onde a automação e o monitoramento remoto podem reduzir custos operacionais e aumentar a eficiência da gestão dos silos. Para aprimorar ainda mais os resultados, futuras melhorias serão consideradas, como o aperfeiçoamento do armazenamento da placa e a otimização do método de prototipagem, visando reduzir o ruído de sinal e garantir maior segurança na instalação final.

Além dos resultados técnicos, destaca-se a importância dos conhecimentos adquiridos ao longo do projeto, especialmente no que diz respeito à prototipagem e impressão da placa, bem como aos aspectos de programação envolvidos na construção do servidor, conversão de dados e apresentação visual das informações ao usuário final. Esses aprendizados contribuem diretamente para a evolução da solução e sua aplicabilidade em cenários agrícolas reais.

Referências

ALVES COSTA, Rafael. **Sistema de monitoramento de reservatórios de água com abordagem e conceito de Internet das Coisas**. 2019. 50 p. Trabalho de Conclusão de Curso (Bacharel em Engenharia Elétrica) – Instituto Federal Campus Formiga, Formiga, MG, 2019.

COELHO, M. S. **Sistemas Supervisórios**. 6. ed. Apostila. 3. rev. Centro Federal de Educação Tecnológica de São Paulo, 2009. 142 p.

ESPRESSIF. **ESP 32 a different IoT power and performance**. 2019. Disponível em: <https://www.espressif.com/en/products/hardware/esp32/overview>. Acesso em: 12 abr. 2025.

ESPRESSIF. **ESP32 Series. Datasheet**. 2022. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Acesso em: 03 ago. 2025.

FATHONI, A. N. **Rancang Bangun Smart Home berbasis IoT Menggunakan Telegram Messenger Bot e NodeMCU ESP 32**. TELKA, v. 9, p. 34-43, 2023.

FERREIRA, Rafael Florenciano. **Sistema de monitoramento interno para silos elevados de armazenagem de grãos**. 2024. 58 p. Monografia (Graduação em Sistemas de Informação) – Universidade Estadual de Mato Grosso do Sul, Dourados, MS, 2024.

ICHSANUDDIN, F. Maullana. **Rancangan Prototype Smart Home Untuk Kontrol Jarak Jauh Pada Perangkat Rumah Dengan Mikrokontroler ESP32**. Seminar Nasional Mahasiswa Fakultas Tecnologia da Informação (SENAFTI), p. 1287, 2022.

OLIVEIRA, Sérgio de. **Internet das coisas com ESP8266, Arduino e Raspberry Pi**. 1. ed. São Paulo: Novatec, 2017. 240 p.

ROMBEKILA, A. **Prototype Smart Home Berbasis IoT com Handphone Android Usando NodeMCU ESP32**. Jurnal Teknik AMATA, v. 3, 2022.

SANTOS, B. P. et al. **Internet das Coisas: da Teoria à Prática**. 2017. 50 p. (Departamento de Ciências da Computação) – UFMG, Belo Horizonte, Minas Gerais, 2017.

SANTOS, F. T. DOS.; MARINATO, M. M.; SANTOS, W. C. DOS. **Sistema Supervisório: Utilização de um sistema supervisório na usina fotovoltaica**. 2021.

SILVA, João Lucas de S. et al. **Plataforma Arduino integrado ao PLX-DAQ: Análise e aprimoramento de sensores com ênfase no LM35**. In: INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA BAHIA (IFBA). 2014.

SUCIPTO; MUZAKILLOH, Muhammad Najibullo; KARAMAN, Jamilah; ZAKUR, Yahya. **Enhancing RESTful API Authentication with Cryptography in Student Information Systems**. ILKOM Jurnal Ilmiah, v. 16, n. 2, p. 125-133, ago. 2024.

AZ-DELIVERY. **ESP32 Dev Kit C V4 NodeMCU WiFi CP2102 Development Board**. 2025. Disponível em: <https://www.az-delivery.de/en/products/esp32-developmentboard>. Acesso em: 23 ago. 2025.