



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO DE CIÊNCIA, TECNOLOGIA E SAÚDE  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Deividhy Junior Tonetti

**Sistema de Controle e Monitoramento para um Silo de Grãos**

Araranguá  
2022

Deividhy Junior Tonetti

**Sistema de Controle e Monitoramento para um Silo de Grãos**

Trabalho de Conclusão de Curso do Curso de Graduação em Engenharia de Computação do Centro de Ciência, Tecnologia e Saúde da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia de Computação.  
Orientador: Prof. Jim Lau, Dr.

Araranguá  
2022

Deividhy Junior Tonetti

**Sistema de Controle e Monitoramento para um Silo de Grãos**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Engenharia de Computação” e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Computação.

Araranguá, 15 de dezembro de 2022.

---

Analúcia Schiaffino Morales, Dra.  
Coordenadora do Curso

**Banca Examinadora:**

---

Prof. Jim Lau, Dr.  
Orientador

---

Prof. César Cataldo Scharlau, Dr.  
Universidade Federal de Santa Catarina

---

Prof. Fábio Rodrigues de la Rocha, Dr.  
Universidade Federal de Santa Catarina

# Sistema de Controle e Monitoramento para um Silo de Grãos

Deividhy Junior Tonetti

15 de dezembro de 2022

## RESUMO

O Brasil é um dos maiores produtores de arroz do mundo, sendo que a maior parte do plantio ocorre na região sul do país. Os anos de 2021/2022, foram anos atípicos, com incidências de secas e inundações em algumas regiões do país, que ocasionaram diversas perdas no setor da agricultura e pecuária. Com tais ocorrências, houve a diminuição da produção de arroz no Brasil, porém, o potencial de produção de arroz ainda é muito grande, tanto ele irrigado quanto sequeiro. Empresas públicas de pesquisa agropecuária como a Empresa de Pesquisa Agropecuária e Extensão Rural de Santa Catarina (EPAGRI), vêm desenvolvendo novas formas de aumentar a produção de arroz através de manipulação genética de sementes e melhoria de manejos. No entanto, o aumento da produção de arroz deve ser acompanhado por um aumento na capacidade de armazenamento do mesmo e com máxima redução nas perdas por temperatura e umidade. Um dos métodos utilizados para isso é a aeração dos grãos, que força uma passagem de ar utilizando uma turbina de secagem. Essa turbina de secagem uniformiza a temperatura garantindo um grão de qualidade em um longo período de tempo. Neste contexto, a turbina fica ligada cerca de 15 a 20 dias consecutivos, que poderiam ser reduzidos através de um controle inteligente de temperatura e umidade, além de, um acompanhamento no processo de secagem e redução no consumo de energia elétrica. Neste trabalho, propõe o desenvolvimento de um sistema de controle e monitoramento para o processo de secagem de grãos, aplicando em um protótipo de um silo de pequena escala. Este sistema é composto por sensores, que realizam a medição de temperatura e umidade, e também por dispositivos microcontroladores. Além disso, serão utilizados conceitos de *IoT* (*Internet of Things*) como uma solução para o monitoramento e gerenciamento remoto de tarefas.

**Palavras-chaves:** Internet das Coisas. Armazenamento de Grãos. Sistemas Embarcados. ESP32. Home Assistant.

## 1 INTRODUÇÃO

Nos últimos anos, Santa Catarina têm apresentado índices positivos de produtividade agropecuária, sendo um dos destaques nacionais na produção de alimentos, embora a sua área represente apenas 1,1% do território nacional (EPAGRI, 2022). Segundo (EPAGRI, 2021), a agricultura catarinense se caracteriza pelo modo de produção familiar, pela pequena propriedade e pela diversidade produtiva, com expressão na produção de grãos, frutas, mel, leite, carnes e produtos florestais.

Embora se tenha uma alta produção, observam-se elevados índices de perdas dos produtos agrícolas entre a colheita e o armazenamento, principalmente devido à falta de tecnologias específicas de conservação e estruturas de secagem deficientes. O armazenamento de grãos em propriedades rurais comumente acontece em locais tecnicamente deficientes, sem secagem eficiente e sujeitos a ataques de insetos, roedores, fungos e pragas. Como consequência, ocorre o estímulo do metabolismo do grão, provocando deteriorações, formação de micotoxinas, e reduzindo a qualidade do produto agrícola (CREPALDI, 2022).

Por este motivo, os produtores reduzem os investimentos necessários e acabam escolhendo investir nas etapas de produção, e diminuindo os recursos ou eliminando os investimentos no pós-colheita. Com a diminuição dos cuidados nesta última etapa, os produtores recorrem às indústrias e cooperativas para armazenar ou vender sua produção nas piores condições para comercialização, sujeitando-se a descontos de impureza, secagem, pagamento de taxas de armazenamento e diminuindo seus ganhos com a colheita (ELIAS; RUBIN, 2020). Com isso, grande parte do que poderia ser o lucro da atividade acaba indo para intermediários (ELIAS; OLIVEIRA; VANIER, 2017).

Deste modo, o armazenamento e condicionamento de grãos é uma etapa importante no desenvolvimento agrícola. Segundo IBGE (IBGE, 2021), a capacidade disponível para armazenamento no Brasil atingiu 183,3 milhões toneladas no segundo semestre de 2021. O volume é 1,5% maior que o registrado no semestre anterior. Por região, os silos predominam na Região Sul, responsáveis por 63,1% da capacidade armazenamento da região e 49,8% da capacidade total de silos do país.

No entanto, dados da CONAB demonstram que apenas 15% de todo o volume de grãos produzidos no país encontram-se armazenados junto à unidade produtora, o restante (85%) depositados em locais relativamente distantes de onde foram produzidos, gerando graves problemas relacionados ao transporte, dificuldade para conservar o produto e a sua consequente perda de quantidade e qualidade (CONAB, 2021). Estas perdas superam 10% do volume colhido, segundo dados do Ministério da Agricultura, Pecuária e Abastecimento. A carência de infraestrutura adequada ao armazenamento dos grãos é evidenciada em pequenas propriedades (JÚNIOR; REIS NETO, 2021).

Para diminuir os problemas descritos, a EMATER/RS desenvolveu um modelo de silo de alvenaria armada com baixo custo de construção, de fácil operação e com

bons resultados (MARTINS *et al.*, 2013). A infraestrutura de armazenamento proposta utilizar ar natural para a secagem, permitindo, assim, uma boa conservação dos grãos. No sistema atual, o ventilador do tipo turbina é ligado e desligado manualmente pelo operador, impactando nos custos com energia elétrica e nem sempre está adequado às condições ambientais favoráveis ao funcionamento do sistema. Além disso, a falta do monitoramento de algumas variáveis climáticas, principalmente a umidade e temperatura, compromete a qualidade dos grãos em silos de armazenamento, podendo levar o surgimento de pragas, na deterioração dos grãos e, por consequência, diminuindo o valor comercial do produto ou sua inutilização e descarte.

Atualmente, existem publicações relatando resultados obtidos em pesquisas envolvendo monitoramento, automação e controle do processo de secagem e armazenamento de grãos. Uma revisão sobre os sensores utilizados para monitoramento da qualidade dos grãos pode ser vista em (NEETHIRAJAN; JAYAS, 2007). Pesquisas brasileiras apresentam contribuições nas áreas de monitoramento (CITOLIN, 2012), (CEZARO STANGA, 2017) e (GARCIA; SCHARLAU, 2021), automação (CARBONERA, 2015), (SANTOS FURTADO *et al.*, 2022) e controle da umidade e temperatura (LEMOS, 2017). Também é possível destacar os resultados de controle obtidos internacionalmente, por exemplo, (ZHANG *et al.*, 2013) e (GUHA; MISHRA, 2014).

Dada a necessidade de um melhor monitoramento e controle nos silos, inclusive de modo remoto, o conceito de Internet das Coisas (*Internet of Things – IoT*), é uma solução promissora, possibilitando o gerenciamento e monitoramento de atividades à distância remotamente. Essa alternativa também pode fornecer uma interface de maneira centralizada para um conjunto de informações como, por exemplo, temperatura e umidade interna e externa, a partir das informações obtidas pelos sensores. A *IoT* redefine a maneira como se interage com o mundo físico e viabiliza formas através do uso de técnicas computacionais, de modo a permitir, através de um sistema central, o controle, o monitoramento e a automação de um silo, por exemplo. Nesse contexto, este trabalho propõe o desenvolvimento de um sistema de controle e monitoramento para o processo de secagem de grãos, aplicando em um protótipo de um silo de pequena escala. Este sistema é composto por sensores, que realizam a medição de temperatura e umidade, e também por dispositivos microcontroladores. Além disso, serão utilizados conceitos de *IoT* como uma solução para o monitoramento e gerenciamento remoto de tarefas.

O presente trabalho está estruturado da seguinte forma. Na segunda seção, será apresentada a fundamentação teórica, no qual, serão descritos conceitos relacionados ao projeto. Em seguida, serão apresentados os trabalhos correlatos relacionados a alguns projetos envolvendo o controle e monitoramento de sistemas de automação de silos de armazenamento. Posteriormente, na quarta seção, é apresentada a metodologia de desenvolvimento, no qual, estão relacionadas as etapas que constituíram a elaboração do presente trabalho. A quinta seção mostra os resultados obtidos. Já na sexta seção, são

apresentados os destaques finais e as perspectivas para continuação desta pesquisa. Por fim, são apresentados os agradecimentos e as referências utilizadas neste trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentados alguns conceitos que embasaram este projeto, fornecendo uma visão geral dos principais aspectos que serão abordados, além de, auxiliar na interpretação e discussão do trabalho.

Na sessão 2.1, será apresentado os conceitos básicos dos controladores que foram utilizados, no qual, tendo em vista a necessidade da utilização de um controlador PID para o sistema do controle, diferenciando de um sistema *On-Of*, tem-se como objetivo introduzir os conceitos que será aplicado na sessão 4, no microcontrolador ESP32 (2).

Nas sessões 2.2, 2.3 e 2.4, tem por objetivo apresentar alguns conceitos utilizados para a comunicação do sistema, sendo que, utilizou-se a arquitetura *IoT* em conjunto com os protocolos de comunicação ESP-NOW, para a comunicação entre os dois dispositivos ESP32, e o protocolo *MQTT*, responsável pela troca de informações do microcontrolador ESP32 (1) para o *Home Assistant*.

Finalizando, na sessão 2.5, será apresentado o *Home Assistant* e os respectivos complementos, entre eles estão: o InfluxDB, o Grafana e o *Hass.io Access Point*. O *Home Assistant* é o sistema operacional que auxilia a parte de comunicação e armazenamento dos dados através dos módulos instalados. Também é responsável pelo gerenciamento dos usuários e visualizações, como, por exemplo, gráficos gerados a partir do Grafana.

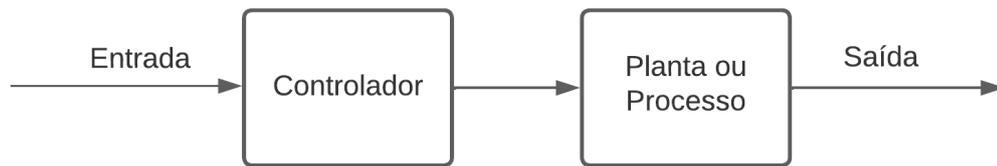
### 2.1 Sistema de Controle

Um sistema de controle é uma combinação de componentes que atuam em conjunto gerando um sistema. Segundo (TORGA, 2016), este sistema pode ser descrito como um bloco capaz de receber entradas e produzir saídas, que pode ser denominado como lei de controle (CLEBER, 2004). Deste modo, um sistema pode ser definido através de duas maneiras, sendo em um controle em malha fechada ou de malha aberta (BAZANELLA; SILVA JUNIOR, 2005).

Observando a Figura 1, tem-se um sistema de malha aberta que realiza um controle para obter a saída desejada sem a realimentação do sistema, ou seja, a saída não exerce nenhuma influência na ação do controle (TORGA, 2016).

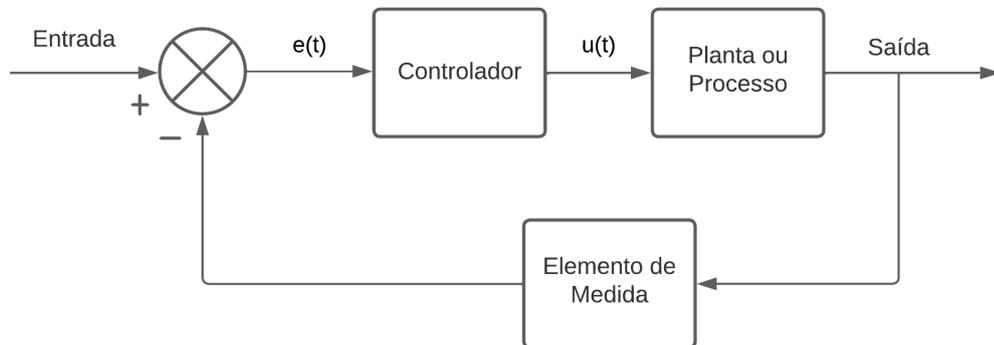
Já na Figura 2 tem-se, um sistema de malha fechada, que possui como característica principal a manipulação da entrada através da influência direta da saída. Esses sistemas que possuem retroalimentação devem encontrar o erro de processo e tentar minimizar ao máximo o erro entre a entrada e a saída. Usualmente são utilizados os controladores, proporcional (P), integral (I), derivativo (D) e suas combinações (BAZANELLA; SILVA JUNIOR, 2005).

Figura 1 – Controle de Malha Aberta



Fonte: (TORGA, 2016)

Figura 2 – Controle de Malha Fechada



Fonte: (TORGA, 2016)

A equação 1 nos remete ao controlador proporcional, que influencia diretamente a variável manipulada  $u(t)$  que é proporcional ao erro  $e(t)$ . O erro pode ser descrito como a diferença entre o valor desejado, denominado de *SetPoint* (SP), e a variável de processo (PV), mostrada na equação 2. Por fim, tem-se o ganho do controlador determinado por  $K_p$ , em algumas situações, um valor elevado de  $K_p$  pode tornar o sistema instável (TORGA, 2016).

$$u(t) = K_p \cdot e(t) \quad (1)$$

$$e(t) = r(t) - y(t) = SP - PV \quad (2)$$

Observando a equação 3 tem-se a junção de um controlador Proporcional com Integral (PI). O intuito da utilização do termo integral é mover a variável manipulada (MV) através da saída, então, tem-se a função do erro e integral do erro como mostrado na equação 3.

$$u(t) = K_p \left( e(t) + \frac{1}{T_i} \int_0^t e(t) dt \right) \quad (3)$$

A partir deste momento pode-se aplicar a transformada de Laplace, considerando as condições iniciais nulas, com o intuito de expressar a equação em domínio da variável complexa. Com isso, é possível obter a equação 4, cujo valor de  $T_i$ , refere-se ao tempo integral necessário para que ação de controle seja atualizada.

$$u(t) = K_p \left( 1 + \frac{1}{T_i s} \right) E(s) \quad (4)$$

Segundo (LOURENÇO, 1997) o controlador PID tem como principal característica eliminar o erro em regime permanente quando adicionado um polo na origem da função de transferência do controlador.

## 2.2 Internet das Coisas (IoT)

A definição de Internet das Coisas deriva-se do inglês *Internet of Things (IoT)*, o qual foi criado pelo britânico Kevin Ashton em 1999. A *IoT* pode ser descrita como uma rede de objetos físicos, como sensores e atuadores, capazes de comunicar-se entre si e transmitir dados (SANTOS *et al.*, 2016). Segundo (ROCHA, 2021), a internet das coisas nos permite conectar dispositivos e objetos do nosso dia a dia, através da internet. Os dispositivos conseguem processar e comunicar com os sensores, os quais podem ser denominados como objetos inteligentes.

Uma das grandes vantagens da utilização da tecnologia *IoT* é a possibilidade de implantação em lugares de difícil acesso, pois utiliza a rede sem fio para o envio de dados coletados, que posteriormente, devem ser enviados para um *software*. Outro aspecto positivo é que a maioria dos sensores são de baixo custo (PINTO, 2021), sendo acessíveis e confiáveis, permitindo que diversos projetos na área sejam empregados.

A tecnologia da Internet das Coisas já está na nossa rotina a anos com a utilização de *smartphones*, *smartwatches*, dispositivos inteligentes como Alexa, Google Assistente, lâmpadas e interruptores inteligentes. A ascensão da rede 5G no Brasil, possibilitará aplicar o conceito de *smart cities*, ou melhor, cidades inteligentes, com monitoramento de ruas e vias públicas de uma forma mais inteligente e eficaz, além da modificação dos meios de transporte, telemedicina, entre outros (CUNHA *et al.*, 2021).

## 2.3 Protocolo de comunicação (MQTT)

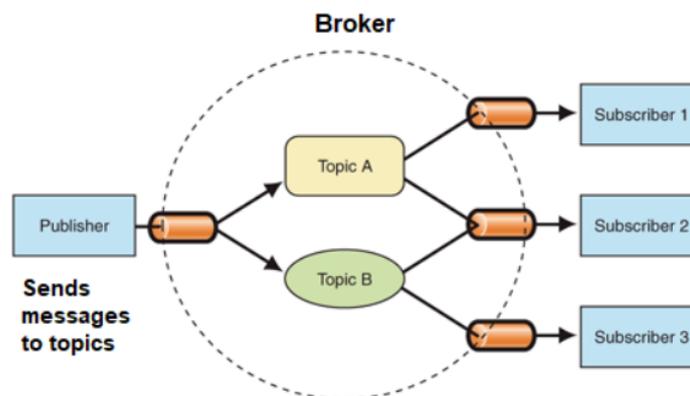
O protocolo *MQTT* (do inglês: *Message Queue Telemetry Transport*) é baseado em uma arquitetura *publish/subscribe* (publicação/assinatura) que opera na camada de aplicação sobre o protocolo *TPC/IP* (camada de transporte) (ZOLETT; RAMIREZ, 2020). O *MQTT* foi desenvolvido visando minimizar o uso de largura de banda da rede

e recursos dos dispositivos (ROCHA, 2021), cuja principais características descritas por (KPIZINGUI, 2020) são:

- A transmissão de grandes volumes de dados, distribuindo as sobrecargas;
- Possui uma arquitetura baseada em eventos com entrega bidirecional e assíncrona de baixa latência;
- É possível transmitir dados para vários clientes de uma forma simples;
- Baixo consumo de energia, pacotes de dados minimizados e facilidade de implementação.

A Figura 3 ilustra como ocorre a comunicação do protocolo *MQTT* entre o *publisher* e o *subscriber*. Todos os clientes são gerenciados por um *Broker*. Neste processo de comunicação da Figura 3 tem-se um cliente que terá o papel de *publish* e transmitirá a mensagem/dado, para um tópico de destino e o conteúdo da mensagem. Essa mensagem em questão será recebida pelo *Broker*, o qual, será o responsável por geri-la e encaminhá-la ao *subscriber*, que foi previamente inscrito no tópico. Caso o cliente queira tornar-se um *subscriber* em algum tópico, deverá apenas encaminhar uma solicitação ao *Broker* que conseguirá fazer uma ligação entre o cliente e o tópico em questão.

Figura 3 – Fluxo de dados do protocolo *MQTT*



Fonte: (VALEED, 2021)

## 2.4 Protocolo de comunicação ESP-NOW

O ESP-NOW é um protocolo de comunicação desenvolvido pela Espressif visando transmitir dados entre os microcontroladores ESPs.

Segundo (PASIC; KUZMANOV; ATANASOVSKI, 2021), uma das vantagens da utilização deste protocolo é quando não existe uma rede de internet e tem-se a necessidade

de enviar pequenos pacotes de dados de uma forma rápida e simples. Para isso, o protocolo ESP-NOW utiliza uma frequência 2.4GHz com os padrões IEEE 802.11 para o envio desses pacotes. Outro ponto avaliado pelos autores é a distância de comunicação, que gira em torno de 190 metros em linha reta. Em contrapartida, tem-se duas desvantagens principais. Uma delas é que a carga útil (mensagem) deve ser de, no máximo, 250 Bytes, sendo relativamente pequena dependendo da aplicação. A segunda, podem-se conectar no máximo 10 dispositivos *peer-to-peer*.

Na Figura 4 nota-se que o ESP-NOW pode transmitir dados de um para muitos, muitos para um, ou, um para um. A terminologia utilizada normalmente é *Master* e *Slave*, ou a combinação deles (PASIC; KUZMANOV; ATANASOVSKI, 2021).

Figura 4 – Comunicação unidirecional do ESP-NOW



Fonte: (PASIC; KUZMANOV; ATANASOVSKI, 2021)

## 2.5 Home Assistant

*Home Assistant* é um *software open source* escrito em Python que é normalmente utilizado em automação residencial. Quando instalado, transforma o dispositivo em um *smart hub* (FARIA, 2018). A documentação do *Home Assistant* explica exatamente todas as etapas necessárias para configuração, que pode ser instalado em um Raspberry pi, ou como imagem em uma máquina virtual.

A plataforma do *Home Assistant* foi pensada de maneira modular, ou seja, possui suporte para vários dispositivos *IoTs* como lâmpadas inteligentes, conectores inteligentes, assistentes virtuais, como Alexa e Google Assistente, e outros dispositivos *IoTs*. A parte de modularização depende muito dos fabricantes dos equipamentos, os quais desenvolvem aplicações justamente para atender uma demanda de controle. (SOLHA, 2018)

### 2.5.1 Add-ons - Complementos

#### 2.5.1.1 InfluxDB

O InfluxDB é um banco de dados *open source* de série temporal (*Temporal Series Database - TSDB*), desenvolvido com o intuito de trabalhar com um alto volume de escritas e consultas de dados utilizando pouquíssimo poder computacional e alta de performance.

Lançado em 2013 pela empresa Errplane, o InfluxDB é comumente utilizado no campo científico, mais precisamente no contexto *IoT*, com sensores enviando dados em pseudo tempo real, conseguindo trabalhar de uma forma elementar e muita rápida (CRUZ, 2017).

Esse banco de dados possui uma linguagem de consulta chamada InfluxQL, que se assemelha com o SQL (*Structured Query Language*), permitindo consultas extremamente rápidas e eficientes. Os principais fundamentos de funcionamento do InfluxDB derivam-se do conceito de *fields*, *tags*, *points* e *measurement*.

Os *fields* representam colunas que nomeiam e armazenam dados no modelo chave valor para a representação de uma série temporal (gráfico de tempo). Já as *tags* são muito similares aos *fields*, entretanto, os dados são armazenados em formato de metadados, serão os responsáveis por fazer as consultas no banco. O *point* é o campo definido na série e representa um único registro de dados, é semelhante a uma linha da linguagem SQL, como, por exemplo, o campo *timestamp*. Finalmente, o *measurement*, é o responsável por agrupar as *tags* e os *fields*, gerando o equivalente a uma tabela SQL. Por fim, pode-se descrever uma série sendo um conjunto de dados (pontos) parametrizados a partir das *tags*. Isto é, quando realizar uma consulta no *measurement*, tem-se como resultado uma série temporal (CRUZ, 2017).

#### 2.5.1.2 Grafana

O Grafana é um *software open source* e multiplataforma que possui uma gama de visualizações como gráficos, tabela e alertas. Este programa foi desenvolvido com o intuito de tornar os dados acessíveis a qualquer momento, além de, facilitar a visualização dos dados.

Através dele, é possível explorar dados de diversos bancos, como InfluxDB, MariaDB, Postgre, de uma forma simplista. O Grafana unifica os dados existentes, sendo possível compartilhar esses dados em painéis com gráficos esteticamente mais bonitos. O grafana Conta com recursos avançados de consulta e transformação, sendo possível gerar mapas de calor, histogramas e geo-mapas (KOMAROMI, 2022).

#### 2.5.1.3 Hass.io Access Point

O módulo *Hass.io Access Point* tem por objetivo de criar uma rede local usando o Raspberry Pi ou usando adaptadores de rede externos (USBs) e possui diversos tipos de configurações como servidor *DHCP* (*Dynamic Host Configuration Protocol*), opção de compartilhamento de internet, entre outros. Este módulo, não é um módulo oficial do Home Assistant, então é necessário adicionar aos complementos manualmente (LONG, 2022).

### 3 TRABALHOS CORRELATOS

Nesta seção serão apresentados alguns trabalhos encontrados na literatura que utilizam a ferramenta *Home Assistant* como base para automações com o uso da família de microcontroladores ESP, além de, projetos relacionados a automação de silos de armazenamento.

Trazendo os trabalhos relatados para o cenário deste estudo, tem-se como premissa a utilização do *Home Assistant* como ferramenta essencial para o projeto, como pode-se observar no trabalho de (TRIZÓLIO, 2022), no qual, foi desenvolvido um protótipo voltado para a automação residencial utilizando microcontroladores ESP8266 e ESP-01. Neste trabalho, foi utilizado *Home Assistant* para controlar e validar parâmetros externos do ambiente com uso de sensores de temperatura e umidade, luminosidade e dióxido de carbono. Como resultado, foi possível monitorar o ambiente interno da residência utilizando componentes de baixo custo e uma interface de monitoramento em código aberto com os dados monitorados.

Em (SOUZA, 2021) foi utilizado o sistema *Home Assistant* para criar um protótipo de controle universal, centralizando as funções de diversos dispositivos infravermelhos em um único dispositivo. Para a construção deste protótipo, foi utilizado o microcontrolador ESP8266 com um módulo receptor e transmissor de infravermelho. Para a comunicação entre os dispositivos, foi utilizado o protocolo *MQTT*, sendo capaz de realizar a comunicação entre o *Home Assistant* com o ESP8266 e com o assistente de voz Google Assistant.

Em (SOLHA, 2018) tem-se a criação de três módulos capazes de coletar dados e transmiti-los de maneira em que outros sistemas consigam consumir esses dados. Para isso, foi utilizado dois sensores para captação de sinais do ambiente, DHT11 (temperatura e umidade) e LDR (luminosidade), e também, utilizou um *Reed Switch* (interruptor eletromecânico) para verificar o estado da porta, aberto ou fechado. Na parte de *software*, foi utilizado o sistema operacional Raspbian, juntamente com o *software* Docker. O Docker funciona com o conceito de containerização, sendo assim, utilizou esse contexto para instalar uma imagem do *software* *Home Assistant* em um contêiner Docker. O conceito de containerização pode ser expresso como o empacotamento do código de programação em um contêiner, no qual, toda sua infraestrutura atua em um ambiente independente do sistema operacional (HAT, 2021).

Um projeto similar elaborado por (FERRASA; BIAGGIONI; DIAS, 2010), teve como objetivo desenvolver um sistema de comunicação sem fio que utiliza radio frequência para monitoramento de silos graneleiros. Para realizar este protótipo, foram utilizados sensores de temperatura (LM35), sensores de umidade (HS15P), microcontrolador PIC-16F877, além de diversos circuitos integrados (CIs). Utilizando sensores internos e externos, os autores conseguiram transmitir os dados a uma distância de 30 metros para um *software* desenvolvido em Java chamado ARGUS-RF. Além disso, o protótipo desenvolvido pelos

autores, teve um custo de U\$ 873.16, que mediante a época que o projeto foi desenvolvido, considerou-se potencialmente viável para os pequenos produtores de grãos.

No âmbito de controle de temperatura de um silo, utilizando o conceito *IoT*, tem-se o trabalho realizado por (QUARESMA DE ALMEIDA, 2018) que consiste em um sistema de monitoramento e controle para dispositivos móveis utilizando a plataforma *KNoT Network of Things*. Os autores utilizaram um equipamento de bancada chamado PP32 para secagem de grãos, além de, um microcontrolador Arduino, uma Raspberry, sensores de temperatura, termistor *NTC* e outros componentes. Como resultado, foi realizado diversos ensaios, a fim de, entender as técnicas de controle baseado em *on-off* e o PID. Os valores de  $K_p$ ,  $K_i$  e  $K_d$  foram alterados, e notou que o controlador *PID* produziu um consumo maior de energia, em contrapartida, aumentou a vida útil dos atuadores, devido o controle da velocidade.

## 4 METODOLOGIA DE DESENVOLVIMENTO

### 4.1 Requisitos

Para a modelagem do sistema foi necessário definir alguns requisitos de ponto de vista do usuário e do desenvolvedor. Nesta etapa, será descrito como o projeto foi definido conforme os levantamentos dos requisitos.

#### 4.1.1 Requisitos Funcionais

Os requisitos funcionais de um *software* e *hardware* são aqueles que definem o comportamento do sistema. Dentre eles, estão compreendidas todas as etapas de modificações, processos e transformações que ocorrem a partir de uma determinada entrada, consequentemente, resultando em uma saída. Normalmente, as características abordadas, são do ponto de vista do usuário. Para o projeto, foram considerados os requisitos funcionais abaixo:

- RF1: O sistema deve informar os dados obtidos através de forma remota;
- RF2: O sistema deve fornecer a análise de dados na forma gráfica, sendo que, o usuário conseguirá identificar as temperaturas e umidades lidas pelos sensores;
- RF3: O sistema deve capturar as informações dos sensores e alterar a velocidade do motor caso haja necessidade;
- RF4: O sistema deve possuir, no mínimo, dois sensores de DHT11 para obter os dados dos sensores de temperatura e umidade (interno e externo);
- RF5: O sistema deve possuir uma ponte H para regular a corrente elétrica do motor;
- RF6: O sistema deve possuir comunicação *WIFI*;

- RF7: O sistema deve permitir o armazenamento de dados remotamente;
- RF8: O sistema deve permitir a comunicação entre dois microcontroladores ESP32;
- RF9: O *cooler* só funcionará, se e somente se, a unidade externa do silo for menor que a unidade interna do silo;
- RF10: O sistema deve permitir ajustar as entradas/parâmetros do controlador PID.

#### 4.1.2 Requisitos Não Funcionais (RNF)

Os requisitos não funcionais, são todos aqueles relacionados à forma como o *software* tornará realidade de um ponto de vista mais técnico, ou seja, são os requisitos que não estão ligados diretamente às funcionalidades específicas de um *software* ou *hardware*. Neste projeto, foram retratados das seguintes formas:

- RNF1: O sistema deve ser de uso simples, para usuários que tenham o conhecimento básico na utilização de sistemas *web* (sites), ou seja, fácil de operar e intuitivo;
- RNF2: O sistema deve prever perdas de comunicação com a rede *WIFI* que, caso ocorram, deverá possuir outras alternativas de comunicação como a utilização ESP-NOW;
- RNF3: O sistema deve ser modelado tendo em vista a praticidade do usuário;
- RNF4: O sistema deve amenizar ao máximo falhas de obtenção de sinais dos sensores;
- RNF5: Deve ser utilizado um microcontrolador ESP32s para processamento e transmissão dos sinais obtidos;
- RNF6: Deve ser utilizado um microcontrolador ESP32s para comunicação com o motor;
- RNF7: O sistema deve possuir comunicação sem fio de baixo à longo alcance, operando em frequências de 2,4 GHz;
- RNF8: O sistema deve criar uma rede local para comunicação entre dispositivos;
- RNF9: O sistema deve utilizar o protocolo *MQTT* para a transmissão dos dados do ESP32 para *Home Broker MQTT*;
- RNF10: O sistema deve utilizar o protocolo de comunicação ESP-NOW *per-to-per* para comunicação entre os microcontroladores ESP32;
- RNF11: O sistema deve utilizar o ambiente de desenvolvimento Home Assistant;
- RNF12: O sistema deve utilizar um controlador PID (proporcional, integral) para atingir, ou manter próximo, o valor de umidade do grão referência;

## 4.2 Arquitetura do Sistema

Segundo (AVELAR *et al.*, s.d.) muitos projetos desenvolvidos utilizando conceito de *IoT* usam uma arquitetura de duas camadas empregando sensores e servidores. Já na arquitetura de três camadas é acrescentada uma camada intermediária, no caso a internet. Desta forma, a primeira camada refere-se a obtenção dos dados através de sensores e a segunda camada visa armazenar os dados obtidos em um servidor local. Já a terceira camada é responsável por fazer a intermediação entre um servidor *web* e os sensores, utilizando algum meio de comunicação para a transmissão desses dados.

Neste trabalho, foi utilizada uma arquitetura de três camadas: a camada de sensores, a camada de comunicação e tratamento dos dados, e, por fim, a camada de armazenamento dos dados. A camada de sensores será responsável pela obtenção de todos os sinais do ambiente interno e externo do silo.

A camada de comunicação e tratamento dos dados, responsável por receber todos os dados obtidos pela camada de sensores, para que posteriormente, sejam transmitidos para a terceira camada através do protocolo *MQTT*. Por fim, a terceira camada é responsável pelo armazenamento dos dados.

Além disso, tem-se um servidor *web* que utiliza um sistema operacional de código aberto específico para o uso da internet das coisas, denominado como *Home Assistant*. Como descrito na seção 2.5, o *Home Assistant* é um sistema modular, no qual é possível trabalhar com diversos bancos de dados. Neste caso, para o armazenamento dos dados, que serão transmitidos pela segunda camada, foi utilizado a biblioteca InfluxDB. A Figura 5 apresenta o diagrama da metodologia de desenvolvimento para a arquitetura do sistema de gerenciamento e monitoramento baseado no modelo em camada proposto por (AVELAR *et al.*, s.d.).

Figura 5 – Diagrama da metodologia do desenvolvimento.



Fonte: Próprio autor.

### 4.2.1 Diagramas

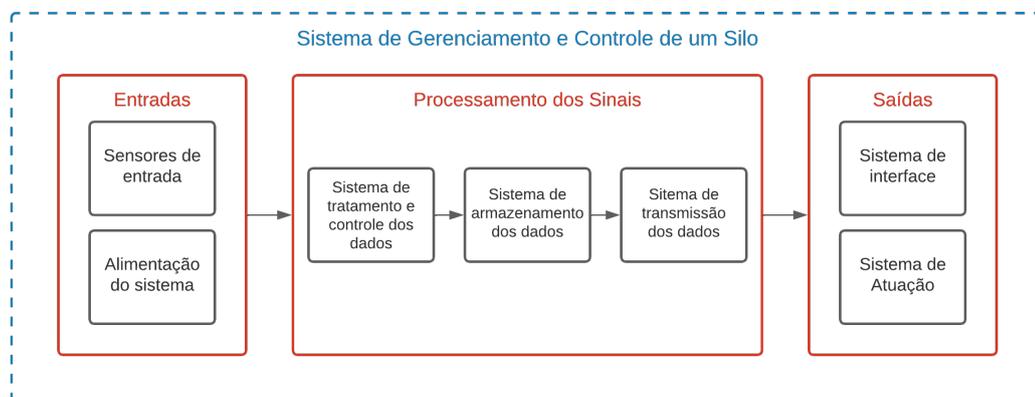
Uma vez definidos os requisitos funcionais e não funcionais, pode-se estabelecer a modelagem do sistema de controle e monitoramento do processo de secagem e armazenamento.

Para isso, foram utilizados o diagrama de blocos funcionais, que segundo (SILVA *et al.*, 2008) pode ser descrito como um conjunto de parâmetros de entradas, saídas, variáveis internas e um algoritmo, executado quando houver alguma solicitação, normalmente utilizado para análise entre relações de componentes. Também foi utilizado um diagrama de comunicação para representação o fluxo de mensagens transmitidas entre os objetos, fornecendo informações de fluxo de dados e fluxo de controle.

#### 4.2.2 Diagrama de Blocos

A Figura 6 ilustra como sistema funciona a partir de blocos funcionais. Neste cenário, existem duas entradas: a alimentação do sistema a partir de energia elétrica e a coleta de dados com os sensores DHT11. No segundo momento, tem-se o processamento e controle dos dados coletados, no qual, as informações do DHT11 utilizadas para realizar o controle do fluxo do ar através da variação da velocidade do *cooler* (saída) e também, serão transmitidas e armazenadas. Por fim, tem-se as saídas do sistema, onde o usuário conseguirá visualizar os dados obtidos a partir do DHT11. Dessa forma, espera-se que o motor será acionado com a velocidade ideal para garantir que a umidade do ar no silo esteja no valor estabelecido pela referência.

Figura 6 – Diagrama de Blocos.



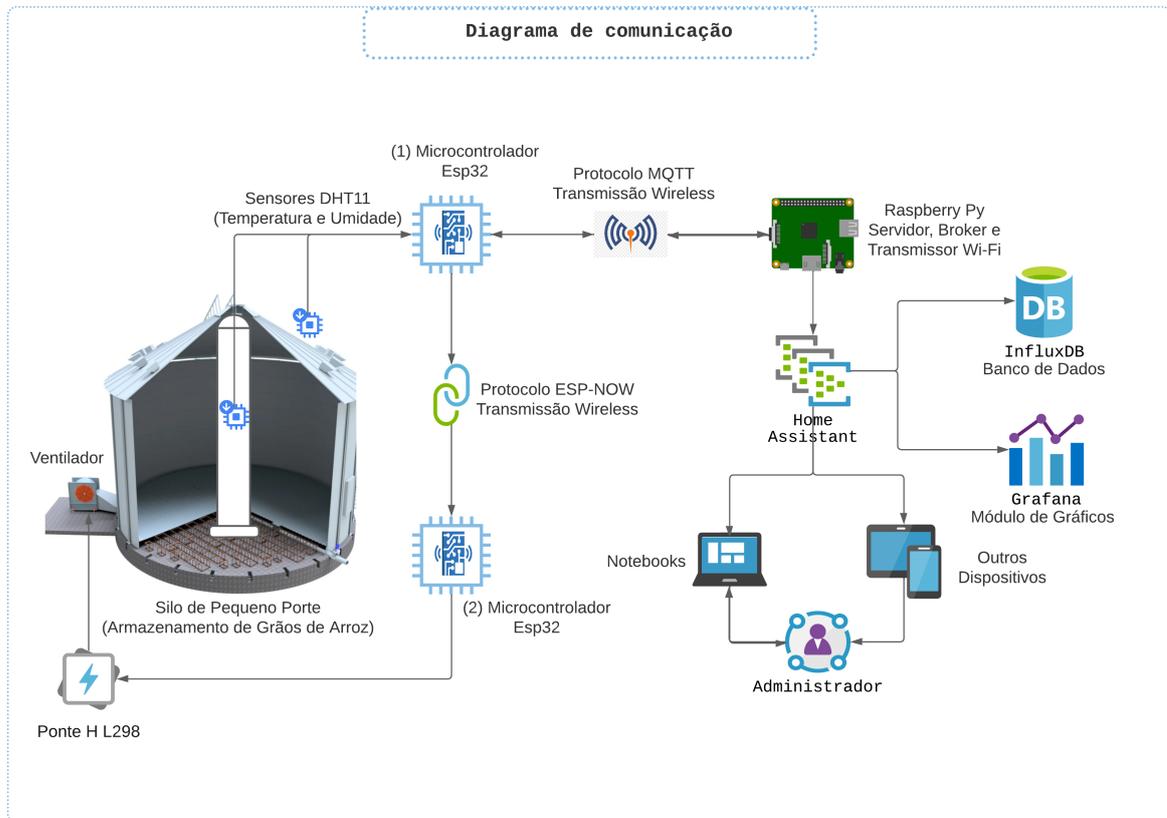
Fonte: Próprio autor.

#### 4.2.3 Diagrama de Comunicação

O diagrama de comunicação representa o fluxo de mensagens entre os dispositivos de monitoramento e o usuário. Na Figura 7 tem-se o diagrama de comunicação do sistema de controle e monitoramento do processo de secagem e armazenamento.

Inicialmente, pode-se notar que tem-se um único usuário principal, definido como o administrador do sistema, ou melhor dizendo, será a pessoa responsável por analisar os dados e trabalhar em cima deles. O *Home Assistant* permite criar dois perfis de usuários diferentes, como o administrador, ou, um usuário comum com acesso somente as visualizações do sistema.

Figura 7 – Diagrama de Comunicação.



Fonte: Próprio autor.

Para realizar o monitoramento da temperatura e umidade serão conectados dois sensores DHT11. Tais sensores irão transmitir a temperatura e umidade interna e externa do silo, se e somente se, houver diferença das informações coletadas. Caso não haja modificação nos valores, os dados não serão enviados para evitar consumo de energia elétrica desnecessária. Os sensores encaminharão os dados para o microcontrolador ESP32 (1) através de cabos condutores.

A partir do momento em que os dados chegam no primeiro ESP32 (microcontrolador 1), haverá um tratamento de dados utilizando funções de checagem dos parâmetros de temperatura e umidade. Caso tenha alguma divergência, vão existir funções recursivas garantindo que o valor do sensor será coletado. Além desse ponto, têm-se funções que garantem a estabilidade das conexões.

Em seguida, os dados serão transmitidos de dois modos e para lugares diferentes. O primeiro modo será enviado via protocolo ESP-NOW, no qual, existe um segundo microcontrolador ESP32. O segundo modo é a partir do protocolo *MQTT* por uma rede *wireless* local, criada pelo Raspberry, chegando a um *Broker*. Esta etapa tem como objetivo principal melhorar a confiabilidade do sistema de controle, ou seja, se por acaso ocorrer algum problema, e o microcontrolador 1 perder a conexão com o microcontrolador 2, devemos garantir que a lei de controle continue operando de forma ininterrupta, é claro,

com os últimos parâmetros de temperatura e umidade que foram enviados, deixando o *cooler* em um velocidade fixa por um determinado tempo, caso a umidade estiver superior ao *SetPoint*, caso contrário, o *cooler* permaneceria desligado.

O segundo microcontrolador ESP32 receberá os dados de umidade e temperatura, via protocolo ESP-NOW, e realizará um controle PID com a finalidade de estabilizar a umidade do grão a partir do acionamento de um *cooler*. Conforme a diminuição do erro e a chegada ao *SetPoint*, o sistema diminuirá ou aumentará a velocidade do *cooler* através de uma ponte H L298N.

Com relação à etapa no qual são enviados esses dados via protocolo *MQTT* para o *Broker*, tem-se o servidor responsável por manter a aplicação do *Home Assistant* em funcionamento. O *Home Assistant* é responsável por toda infraestrutura de banco de dados, *front-end* e *back-end*. Por fim, o administrador terá acesso ao *Home Assistant*, no qual, poderá acompanhar, analisar e alterar os valores de regra de controle, a partir de um sistema operacional como: Android, Linux, Windows, sendo necessário que, os dispositivos estejam conectados na mesma rede.

### 4.3 Implementação

Nesta seção, será apresentado todo o cenário para o desenvolvimento, como, por exemplo, ferramentas, módulos e componentes, além de, suas particularidades, desde o *hardware* até o *software*.

#### - *Hardware*

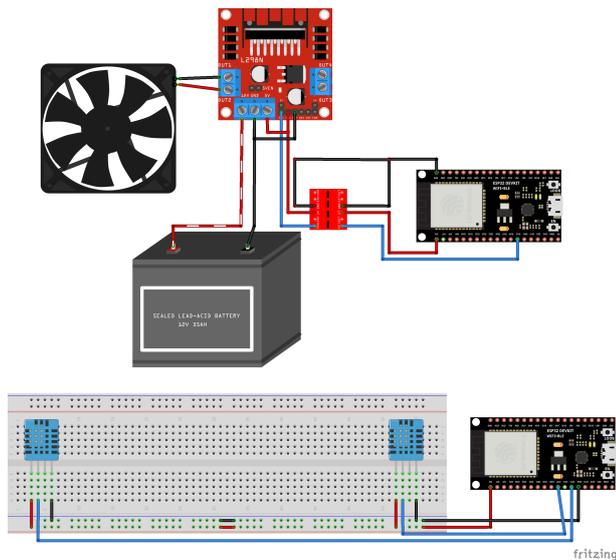
Para realização deste projeto, foi utilizado um Raspberry Pi 3 B+, dois microcontroladores ESP32s, dois sensores de temperatura e umidade (DHT11), uma Ponte H (L298N), um conversor de nível lógico, e um *cooler*. Este *hardware* também pode ser visto no esquema elétrico da figura 8.

Além do esquema elétrico, foi realizada a prototipação de um modelo de silo de pequeno porte, que utiliza dois baldes, palitos de madeira para a separação dos baldes e conexões de tubos de PVC para anexação dos sensores DHT11. As imagens do modelo em pequena escala podem ser vistas nas Figuras 9 e 10. Este protótipo desenvolvido no Laboratório de Automação e Robótica Móvel (LARM), com auxílio dos bolsistas de iniciação científica Baba Culda e Hélio M. Tamba.

#### - *Software*

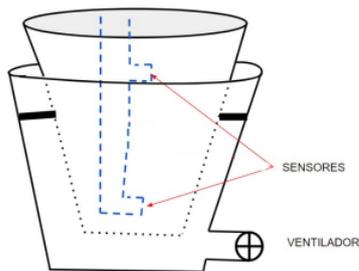
Nos atentando ao projeto como um todo, a parte mais complexa está relacionada ao desenvolvimento do *firmware*, pois ele, é responsável por toda a obtenção dos dados de temperatura e umidade a partir dos sensores. Uma vez recebidos os dados, é feito um tratamento para corrigir possíveis erros de leitura dos sinais devido a fatores externos.

Figura 8 – Esquema Elétrico do Projeto.



Fonte: Próprio autor.

Figura 9 – Modelo de sistema de silos a partir de dois baldes



Fonte: Próprio autor.

Para garantir a comunicação entre os dispositivos com inúmeras variáveis e diferentes parâmetros, foram utilizados os protocolos de comunicação como *MQTT* e *ESP-NOW*.

Além disso, também tem-se a parte do sistema de controle, que possui como parâmetros de entrada os dados de umidade interna e externa do silo, para garantir que o *cooler* esteja com uma rotação adequada, atingindo o valor referência de umidade do ar no silo. O sistema de controle implementado funciona em malha fechada, permitindo avaliar os parâmetros do ambiente com a utilização dos sensores.

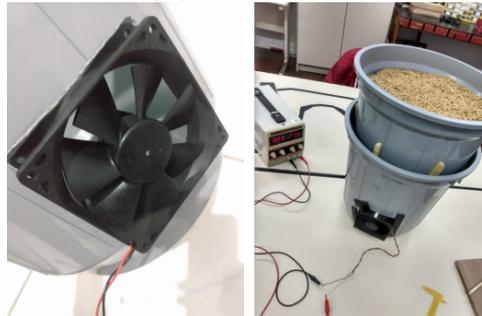
Para o sistema de controle em malha, fechada foi utilizado um controlador PID (Proporcional Integral), no qual as ações geradas visam a atuação do ambiente interno do silo. Assim, na medida que a variável de controle é alterada, o nosso sistema de controle buscará corrigir o erro proporcionalmente ao tamanho do desvio no tempo, caso o erro cresça, proporcionalmente a variável de controle crescerá. Para realização do controle PID, foi utilizada uma biblioteca já consolidada denominada de *PID\_V1*, desenvolvido por (BEAUREGARD, 2018), através dela têm-se funções que controlam os parâmetros de *KP*, *KI* e *KD* de forma simples e prática.

Para realizar a parte de controle e modificar a variável responsável pela velocidade

Figura 10 – Construção do protótipo



(A) Furos para Aeração (B) Palitos para Separação dos Baldes



(C) Cooler (D) Protótipo Final

**Fonte:** Próprio autor.

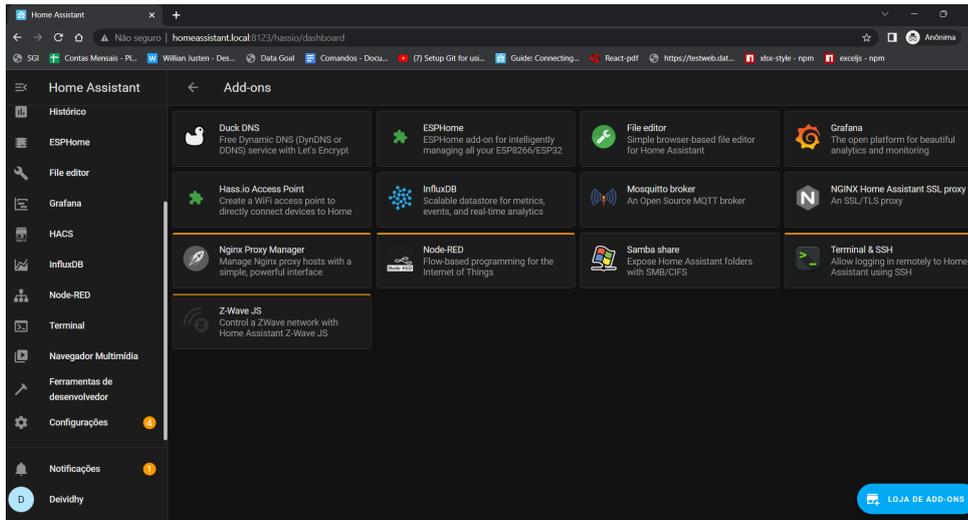
do motor, é necessário transmitir a umidade dos sensores internos e externos dos sensores. Para isso, foi necessário a utilização do protocolo de comunicação *ESP-NOW*, que encapsula estes dados e envia as informações do ESP32 (1) para o ESP32 (2), como pode ser observado na Figura 7.

Uma vez instalado o *Home Assistant*, foi necessária a utilização de alguns complementos/módulos com o intuito de atingir os nossos objetivos, como mostrado na Figura 11. Cada módulo tem um papel essencial na execução do projeto, sendo que, foram considerados para atender um público com pouco conhecimento na área de automação.

Com relação aos módulos mostrados na Figura 11, tem-se o Mosquitto Broker que auxilia a comunicação entre o microcontrolador e o *Home Assistant* utilizando o protocolo *MQTT* e um *Broker*, pré-configurado. Além deste módulo, também tem-se o InfluxDb, onde são armazenados os dados coletados dos sensores. Outro módulo instalado foi Grafana, já descrito anteriormente, que tem a função de mostrar gráficos utilizando a base do InfluxDb. Já o file editor tem a função de auxiliar na edição dos arquivos de configurações *YAML*.

Para garantir que o *Broker* consiga encaminhar as informações para os clientes externos, é necessário que os sensores enviem os dados através do protocolo *MQTT*. Para isso, o próprio módulo do *Mosquitto* disponibiliza uma ferramenta para realizar um *subscribe* de forma temporária, validando assim, as informações que estão chegando ao *Broker*, como pode-se visualizar na imagem 12. Do ponto de vista do cliente, existe outra forma de validar se as informações estão sendo enviadas para o *broker* corretamente. Neste

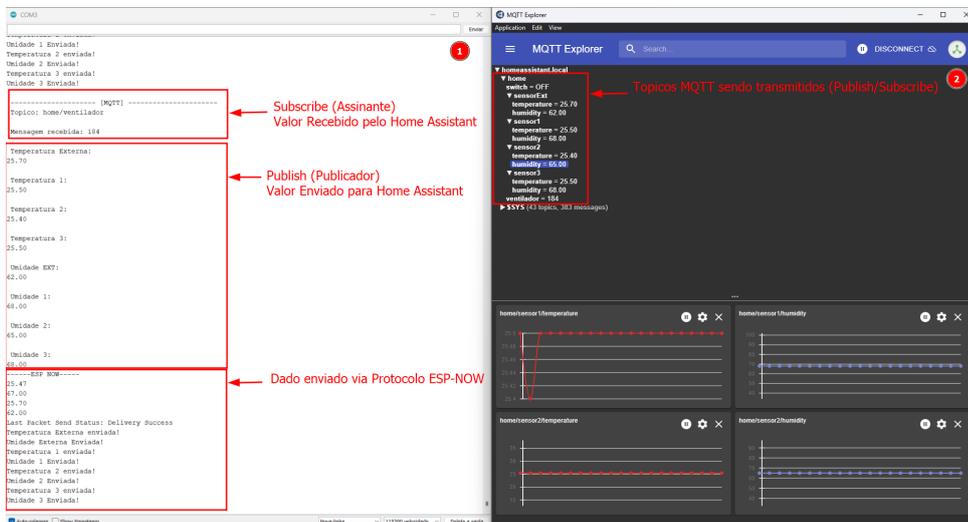
Figura 11 – *AddOns* Instalados.



Fonte: Próprio autor.

caso, pode-se optar pela utilização do *software* MQTT Explorer, mostrado na Figura 12 item 2, onde é possível receber os dados enviados pelo cliente e até mesmo plotar gráficos, porém os dados recebidos não serão persistidos.

Figura 12 – Transmissão de dados via *MQTT* e *ESP-NOW*



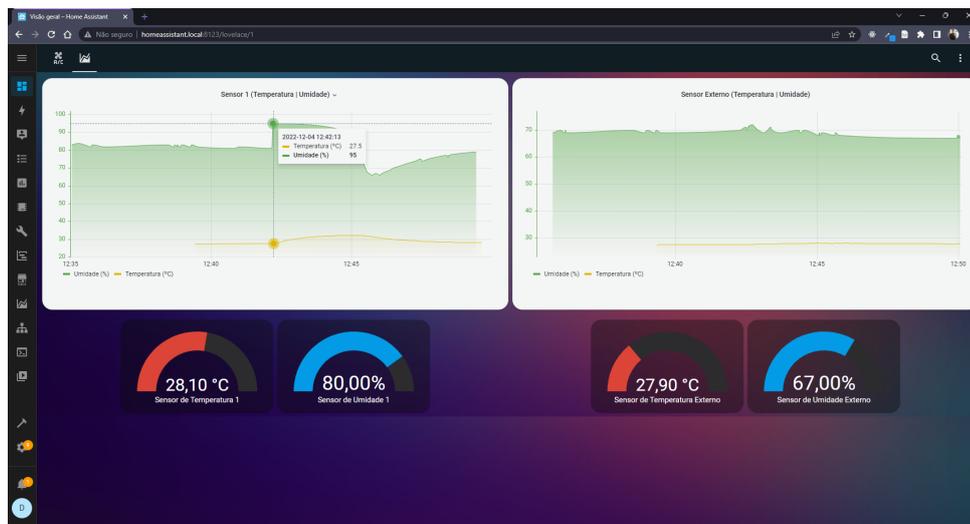
Fonte: Próprio autor.

## 5 RESULTADOS E DISCUSSÕES

Neste seção, será apresentada a implementação do sistema de controle e monitoramento para um silo de grãos aplicados em um modelo em pequena escala. Para isso, foram realizados alguns testes para a validação do sistema de controle e monitoramento para secagem de grãos, mapeando o comportamento do envio dos dados entre os microcontroladores e o *Home Assistant*.

Através da Figura 13, é possível visualizar os gráficos de monitoramento da temperatura e umidade do ambiente interno (sensor 1) e externo. Foi realizado um teste<sup>1</sup> com duração de 15 minutos, que teve como objetivo validar as informações transferidas via protocolo *MQTT* (Figura 12 item 1), entre o microcontrolador e o módulo do Grafana no *Home Assistant*. A partir deste teste, foi possível observar uma oscilação estimulada na umidade interna, sendo que, a umidade encontrava-se em torno 80%<sup>2</sup> e, após a aplicação de mais umidade no sensor 1, resultou em uma umidade de 95%, o máximo aceitável para o sensor DHT11.

Figura 13 – Teste de Temperatura e Umidade no *Home Assistant*



Fonte: Próprio autor.

A finalidade de alterar o valor da umidade interna (Figura 13) é para verificar a capacidade de aquisição dos dados obtidos no ambiente a ser controlado. Em outra aba interna do *Home Assistant* tem-se o controle do sistema, mostrado na Figura 14.

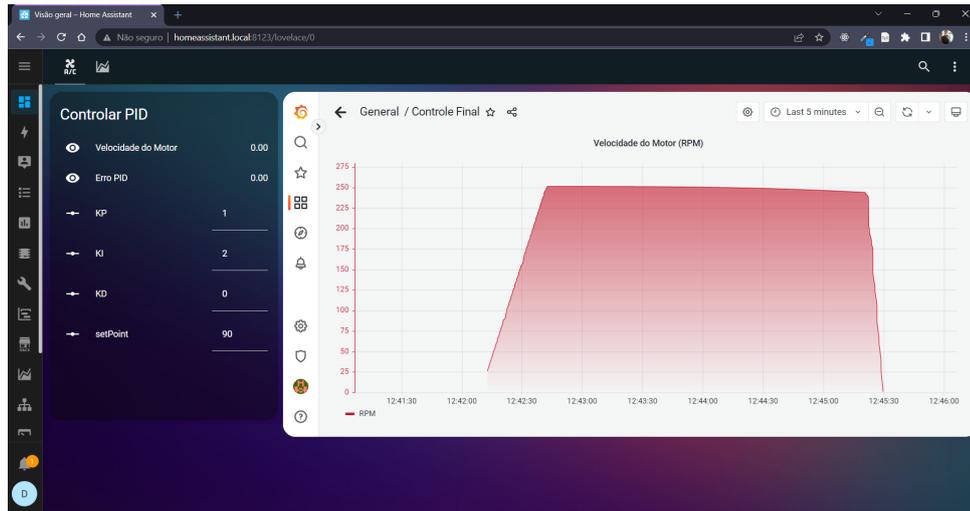
Na Figura 14 são apresentados os parâmetros do *PID*, sendo eles o *KP* (Ganho Proporcional), *KI* (Ganho Integral), *KD* (Ganho Derivativo) e o *SetPoint* (valor final desejado), que nesse caso foi, definido como 90% de umidade. Desde modo, o controlador *PID* deverá ligar o *cooler* caso a umidade do sensor 1 (Figura 13) ultrapasse nosso *SetPoint*, tentando reduzir o erro a 0 e exercendo a remoção da umidade através da aeração. Caso contrário, se o valor da umidade do sensor 1 estiver inferior ao valor do *SetPoint*, o *cooler* permanecerá desligado.

Esta ação gerada pela saída do nosso controle *PID* é possível visualizar no gráfico de velocidade do motor (Figura 14), no qual, comparando os gráficos da Figura 13 (Sensor 1) e Figura 14 (Velocidade do Motor), tem-se o aumento estimulado da umidade juntamente com o aumento proporcional da velocidade do motor, ou seja, a medida que nosso *SetPoint* é ultrapassado nosso erro aumenta proporcionalmente, resultado em um aumento na

<sup>1</sup> Os testes não incluem o ganho derivativo (*KD*)

<sup>2</sup> Os testes realizados foram em dias chuvosos

Figura 14 – Teste de Controle do PID *Home Assistant*



Fonte: Próprio autor.

velocidade do motor para diminuir a umidade do sistema. A medida que a umidade vai diminuindo, conseqüentemente o erro do sistema PID também vai diminuindo, incidindo em uma queda na velocidade do motor até estabilizar em 90%.

Uma vez feito os testes de validação do sistema de controle e monitoramento para os sensores, foi realizado um teste no protótipo, no qual, utilizaram-se os seguintes parâmetros:

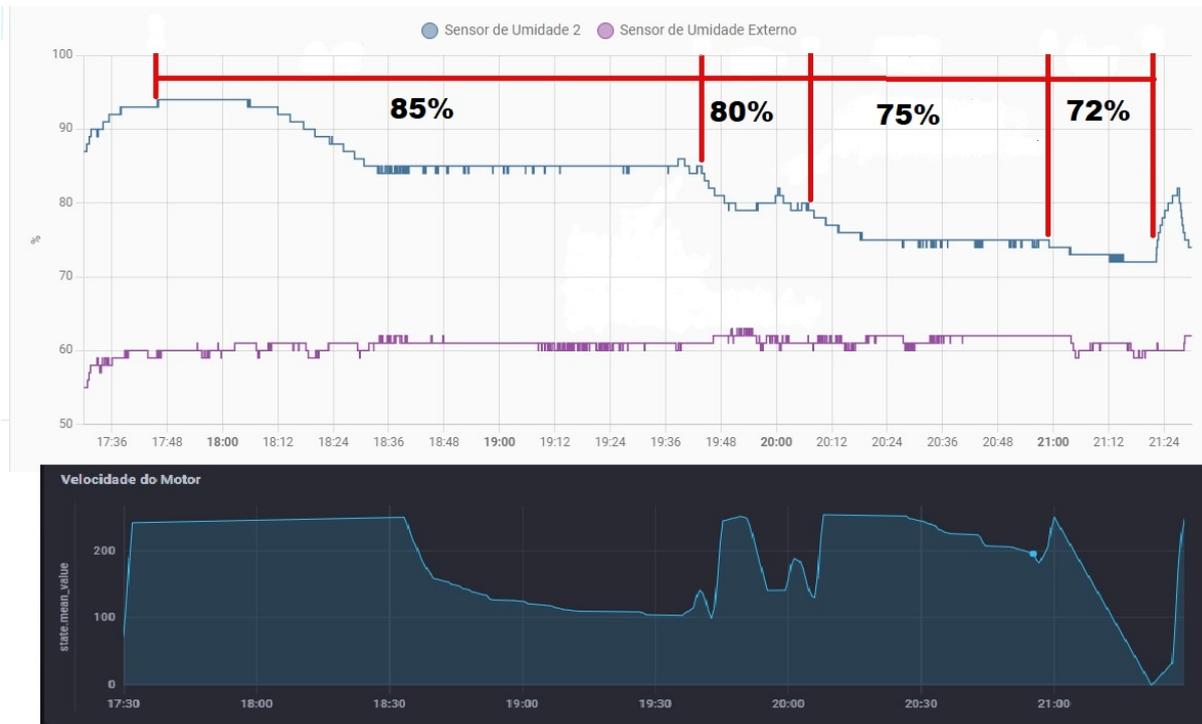
- Peso do grão: 2000 *g*
- Peso do grão umedecido: 2500*g*
- $KP= 2.0$
- $KI= 0.5$

Na Figura 15 é possível verificar o funcionamento do controlador PID, sendo que, a umidade inicial interna do protótipo (Sensor de Temperatura 2) estava em torno de 88% e o *setpoint* foi definido como 85%. Logo, o controlador PID tenta estabilizar a umidade interna em 85% através do acionamento do motor, ocorrendo por volta de 40 minutos. Passado esses 40 minutos, o controlador diminui gradativamente a velocidade do motor para manter a umidade interna do protótipo em 85%.

Visando testar o controlador PID foi ajustado o *setpoint* três vezes para 80%, 75% e 72%, respectivamente. Nota-se que o objetivo do PID é chegar ao mais próximo possível do *setpoint* e estabilizar a umidade reduzindo a velocidade do motor.

Na condição em que o *setpoint* encontrava-se em 80% às 20:00, foi adicionado mais umidade no sistema com a finalidade de testar a atuação do controlador PID. Desta forma, a umidade interna subiu para 82% e rapidamente o controlador atuou, aumentando a velocidade do motor e estabilizando a umidade em 80%.

Figura 15 – Teste de Controle do PID no protótipo.



Fonte: Próprio autor.

## 6 CONSIDERAÇÕES FINAIS

O presente trabalho propôs o desenvolvimento de um sistema de controle e monitoramento para o processo de secagem de grãos, aplicando em um modelo de um silo de pequena escala, inserido no contexto *IoT*. Para isso, foi utilizado a plataforma *Home Assistant* como auxiliador no monitoramento de parâmetros como umidade e temperatura, além de, possuir um controle *PID* ajustável, que vem sendo aplicado para o controle de diversos tipos de processos. Também é importante ressaltar que os resultados preliminares deste trabalho já foram publicados em um artigo no 11º Simpósio de Integração Científica e Tecnológica do Sul Catarinense – SICT-Sul em 2022.

Este trabalho delimita-se ao desenvolvimento do controle e monitoramento para o processo de secagem de grãos de um silo, principalmente focado no *software* e no *firmware*. Logo, os testes realizados tiveram como fundamento a comunicação entre os microcontroladores e a plataforma *Home Assistant*. Para isso, pode-se observar os teste de comunicação na Figura 12. Com relação ao controle do *PID*, nota-se a proporcionalidade no gráfico da velocidade do *cooler* (Figura 12). Neste contexto, o controle busca corrigir o tamanho do desvio do erro de processo ao máximo, atingindo e oscilando em torno do *SetPoint*. Os valores empregados no *KP*, *KI* e *SetPoint* foram utilizados, de modo a, validar o sistema de controle, ou seja, esses parâmetros são utilizados de maneira não otimizada. Portanto, para trabalhos futuros, existe a necessidade de otimizar estes parâmetros através de técnicas como, por exemplo, Ziegler Nichols (BAZANELLA; SILVA JUNIOR, 2005).

As tecnologias escolhidas para o desenvolvimento deste projeto mostraram-se eficientes e viáveis, uma vez que, os componentes eletrônicos utilizados são de fácil acesso e o *software* de gerenciamento utilizado é gratuito e de código aberto, permitindo uma grande integração com diversos tipos de dispositivos *IoT*s.

Os testes para a obtenção da forma otimizada dos parâmetros de controle não puderam ser realizados devido às condições climáticas em Araranguá - SC. Nos dias que houveram as tentativas dos testes, a umidade encontrava-se acima de 80%, influenciando também a calibração dos sensores. Outro ponto crucial foi a falta de entendimento sobre a dinâmica térmica do grão de arroz, ficando como trabalho futuro um estudo mais aprofundado.

Em resumo, para o desenvolvimento de trabalhos futuros, julga-se necessário a otimização dos parâmetros de controle e testes em um ambiente controlado para otimizar o desempenho e reduzir custos de energia, estudo da dinâmica térmica do grão de arroz em diversos climas, acrescentar o módulo *duckDNS* no *Home Assistant* para monitoramento e controle em redes externas, adicionar mais sensores para monitoramento e, por fim, aplicar o sistema em um ambiente real disponibilizado pela Epagri de Araranguá - SC.

## 7 AGRADECIMENTOS

O autor agradece a Deus pelos objetivos alcançados, a esposa Francielle Tonetti e aos pais Odair Tonetti e Sirlei Tonetti pelo cuidado e carinho durante essa jornada, aos professores Jim Lau e César Cataldo Scharlau, pelos ensinamentos, correções e pela dedicação sempre indicando a direção correta que o trabalho deveria tomar, aos bolsistas Baba Culda, Bruno Gobbi e Hélio M. Tamba, que auxiliaram em todo desenvolvimento do projeto, e por fim, a Univ. Fed. de Santa Catarina, Campus Araranguá, pelos recursos disponibilizados.

## REFERÊNCIAS

- AVELAR, Edson Adriano M. *et al.* **Arquitetura de comunicação para cidades inteligentes: uma proposta heterogênea, extensível e de baixo custo.** v. 92. [S.l.: s.n.]. Universidade Federal de Pernambuco (UFPE), Recife.
- BAZANELLA, Alexandre Sanfelice; SILVA JUNIOR, João Manoel Gomes da. **Sistemas de controle.** [S.l.: s.n.], 2005. Principios e Métodos de Projeto. UFRGS, Porto Alegre.
- BEAUREGARD, Brett. **Arduino playground - PIDLibrary.** [S.l.: s.n.], dez. 2018. <https://playground.arduino.cc/Code/PIDLibrary/>. (Accessed on 12/04/2022).
- CARBONERA, Juliano Aluísio. **Dinâmica da secagem do milho em silo secador automatizado.** 2015. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Agrícola) – Universidade Regional Integrada Do Alto Uruguai e das Missões, Erechim - RS.
- CEZARO STANGA, Guilherme de. **Sistema de monitoramento de temperatura e umidade para um silo.** 2017. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação) – Universidade Tecnológica Federal do Paraná (UTFPR), Pato Branco - RS.
- CITOLIN, Ricardo Samuel. **Sistema de termometria para silos.** Porto Alegre, RS: [s.n.], 2012. Trabalho de Conclusão de Curso, Engenharia Elétrica, Universidade Federal do Rio Grande do Sul.
- CLEBER, Zorzi. Controlador PID digital de velocidade de um motor de corrente contínua. **Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação).** Universidade São Francisco. Itatiba–SP, 2004.
- CONAB. **Acompanhamento da safra brasileira de grãos.** Brasília: [s.n.], fev. 2021. Disponível em: <https://www.conab.gov.br/info-agro/safras>. Acessado em 01/04/2021.
- CREPALDI, Aline. **Mapeamento e avaliação comparativa dos silos secadores e armazenadores de grãos instalados na região sul catarinense.** 2022. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Energia) – Universidade Federal de Santa Catarina - UFSC, Araranguá - SC.
- CRUZ, Guilherme. **Séries temporais e o InfluxDB.** [S.l.: s.n.], jun. 2017. <https://gcruz.com.br/blog/series-temporais-e-influxdb/>. (Accessed on 08/29/2022).
- CUNHA, Kaique M. R. *et al.* **Uma abordagem sobre a integração da Computação de Borda Móvel e a Rede 5G para Internet das Coisas na Agricultura 4.0.** [S.l.: s.n.], 2021. P. 118–131. Anais da IX Escola Regional de Informática de Goiás.

ELIAS, Haroldo Tavares; RUBIN, Cezar Augusto. **Diagnóstico e análise da armazenagem de grãos em Santa Catarina**. v. 33. Florianópolis - SC: [s.n.], 2020. P. 7–10.

ELIAS, Moacir Cardoso; OLIVEIRA, Mauricio de; VANIER, Nathan Levien. **Tecnologias de pré-armazenamento, armazenamento e conservação de grãos**. 2017. Material de apoio aos cursos de Graduação em Agronomia e Engenharia Agrícola – Pelotas - RS.

EPAGRI. **Nota Técnica Epagri/Cepa Nº 001/2021 - Conceitos e métodos aplicados à gestão de empreendimentos rurais e custos de produção nos programas da Epagri**. Florianópolis - SC, 2021.

\_\_\_\_\_. **Síntese anual da agricultura de Santa Catarina**. Florianópolis - SC: [s.n.], 2022. Acessado em 10/08/2022. Disponível em: [https://docweb.epagri.sc.gov.br/website\\_cepa/publicacoes/Sintese\\_2020\\_21.pdf](https://docweb.epagri.sc.gov.br/website_cepa/publicacoes/Sintese_2020_21.pdf).

FARIA, Artur Bersan de. **Automação residencial com foco em cenários**. 2018. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Software) – Universidade de Brasília - UNB, Brasília - DF.

FERRASA, Marcelo; BIAGGIONI, Marco Antonio Martin; DIAS, Ariângelo Hauer. **Sistema de monitoramento da temperatura e umidade em silos graneleiros via radiofrequência**. [S.l.: s.n.], 2010. P. 139–156. Energia na Agricultura.

GARCIA, Cibeli Ferro; SCHARLAU, César Cataldo. **Sistema automático de baixo custo para monitoramento de Silos de armazenamento de grãos**. [S.l.: s.n.], 2021. 10 Simpósio de Integração Científica e Tecnológica do Sul Catarinense (SICT-Sul).

GUHA, Paramita; MISHRA, Sunita. **Reduced order modeling & controller design for mass transfer in a grain storage system**. v. 11. [S.l.]: Springer, 2014. P. 399–403.

HAT, Red. **O que é containerização?** [S.l.: s.n.], mar. 2021. <https://www.redhat.com/pt-br/topics/cloud-native-apps/what-is-containerization>. (Accessed on 19/12/2022).

IBGE. **IBGE: capacidade de armazenagem agrícola subiu no 2º semestre de 2021**. Brasília: [s.n.], fev. 2021. Disponível em: <https://agenciadenoticias.ibge.gov.br/>. Acessado em 10/08/2022.

JÚNIOR, Paulo Cláudio Machado; REIS NETO, Stelito Assis dos. **Perdas em transporte e armazenagem de grãos: panorama atual e perspectivas**. Brasília: Companhia Nacional de Abastecimento (CONAB), 2021. Disponível em: [https://www.conab.gov.br/outras-publicacoes/item/download/35953\\_e93eba0ef12529526cef749f8d465f7b](https://www.conab.gov.br/outras-publicacoes/item/download/35953_e93eba0ef12529526cef749f8d465f7b).

KOMAROMI, Zsofia. **New grafana panels, RBAC for plugins, public dashboards, and more | Grafana Labs.** [S.l.: s.n.], ago. 2022.  
<https://grafana.com/blog/2022/08/16/grafana-9.1-release/?pg=graf&plcmt=top-promo-banner>. (Accessed on 09/03/2022).

KPIZINGUI, Maximilien. **Demystifying the MQTT maze: clients, servers, connection, publish, subscribe and its applications.** [S.l.: s.n.], 2020.  
<https://morioh.com/p/93ba2353480e>. (Accessed on 10/15/2022).

LEMOS, Eduardo Henrique. **Automação de um processo de malteação.** 2017. Trabalho de Conclusão de Curso (Bacharel em Engenharia de Automação Industrial) – Centro Federal de Educação Tecnológica de Minas Gerais, Araxá - MG.

LONG, Matt. **Access-point: Hass.io.** [S.l.: s.n.], jun. 2022.  
<https://github.com/mattlongman/Hassio-Access-Point>. (Accessed on 09/21/2022).

LOURENÇO, João. **Sintonia de controladores PID.** [S.l.: s.n.], 1997.  
<http://alvarestech.com/temp/smar/Pid.pdf>. (Accessed on 09/21/2022).

MARTINS, Ricardo Ramos *et al.* A armazenagem sustentável como inovação para a pequena propriedade. **Agroecologia e Desenv. Rural Sustentável, Porto Alegre**, v. 6, n. 1/2, p. 8–25, 2013.

NEETHIRAJAN, Sureshraj; JAYAS, Digvir S. **Sensors for grain storage.** [S.l.: s.n.], 2007. P. 1. American Society of Agricultural and Biological Engineers.

PASIC, Roberto; KUZMANOV, Ivo; ATANASOVSKI, Kokan. ESP-NOW communication protocol with ESP32. **Journal of Universal Excellence**, Faculty of Organizational Sciences-Slovenia, v. 6, n. 1, p. 53–60, 2021.

PINTO, GUSTAVO. **De que maneiras a queda no preço dos sensores de IoT impacta a indústria 4.0?** [S.l.: s.n.], nov. 2021.  
<https://v2com.com/2019/11/21/sensores-de-iot>. (Accessed on 12/16/2022).

QUARESMA DE ALMEIDA, João Gabriel. **Interface homem-máquina para instrumentação e controle de ambiência em silos de armazenagem de grãos no contexto da internet das coisas.** 2018. Monografia (Tecnólogo) - Curso de Graduação em Análise e Desenvolvimento de Sistemas – Universidade Federal do Rio Grande do Norte.

ROCHA, Juary Costa. **Chuveiro inteligente: conectado numa rede IoT.** Mai. 2021. Trabalho de Conclusão de Curso de Graduação - Engenharia de Energia – Universidade Federal de Santa Catarina - UFSC.

SANTOS, Bruno .P *et al.* **Internet das coisas: da teoria a pratica.** [S.l.: s.n.], 2016. Universidade Federal de Minas Gerais (UFMG), Belo Horizonte - MG.

SANTOS FURTADO, Wagner Vitor dos *et al.* **Low-cost automation for artificial drying of cocoa beans: A case study in the Amazon.** v. 40. [S.l.]: Taylor & Francis, 2022. P. 42–49. Drying Technology.

SILVA, Leandro Dias da *et al.* **Geração automática de autômatos temporizados para diagramas de blocos funcionais.** [S.l.: s.n.], 2008. Anais do Congresso Brasileiro de Automática (CBA'08). Universidade Federal de Alagoas (UFAL).

SOLHA, Matheus Coelho. **Construção de módulos baseados em internet das coisas para coleta de dados e gerenciamento de informações.** Nov. 2018. Trabalho de Conclusão do Curso (Bacharelado em Ciência da Computação) – Universidade Estadual Paulista (Unesp) - Câmpus de Bauru.

SOUZA, Lucas Nascimento Santo. **Projeto de um controle remoto universal infravermelho com suporte a comandos de voz.** Mar. 2021. Monografia (Bacharelado em Engenharia da Computação) – Universidade Estadual Paulista (Unesp) - Câmpus de Bauru.

TORGA, Diego Santana. **Desenvolvimento de uma plataforma didática para práticas de controle de velocidade de motor de corrente contínua.** 2016. Monografia (Tecnólogo) - Engenharia de Controle e Automação – Universidade Federal de Ouro Preto (UFOP).

TRIZÓLIO, Ana Beatriz Catarim. **Modúlo didático IOT para automação residencial baseado no ESP8266 ESP-12.** Mar. 2022. Trabalho de conclusão de curso - Bacharel Engenharia Elétrica – Universidade Estadual Paulista (Unesp).

VALEED, Muhammad. **CoAP and MQTT: analyzing the best IoT protocol - IoTEDU.** [S.l.: s.n.], mar. 2021. <https://iot4beginners.com/coap-vs-mqtt/>. (Accessed on 10/22/2022).

ZHANG, Lihui *et al.* **Parameters online detection and model predictive control during the grain drying process.** v. 2013. [S.l.]: Hindawi, 2013. Mathematical Problems in Engineering.

ZOLETT, Daniel; RAMIREZ, Alejandro Rafael Garcia. **Desenvolvimento de uma interface de monitoração remota para o sistema robótico ROBIX, integrando o protocolo MQTT e oROS.** v. 11. [S.l.: s.n.], 2020. P. 405–412. Anais do Computer on the Beach.