



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
UNIDADE ACADÊMICA ESPECIALIZADA EM CIÊNCIAS AGRÁRIAS  
ESCOLA AGRÍCOLA DE JUNDIAÍ  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS

**Interface Homem-Máquina para Instrumentação e Controle de Ambiência  
em Silos de Armazenagem de Grãos no Contexto da Internet das Coisas**

**João Gabriel Quaresma de Almeida**

MACAÍBA/RN

2018

João Gabriel Quaresma de Almeida

**Interface Homem-Máquina para Instrumentação e Controle de Ambiente  
em Silos de Armazenagem de Grãos no Contexto da Internet das Coisas**

Trabalho de conclusão de curso de graduação  
apresentado à Unidade Acadêmica Especiali-  
zada em Ciências Agrárias da Universidade  
Federal do Rio Grande do Norte como re-  
quisito parcial para a obtenção do título de  
Tecnólogo em Análise e Desenvolvimento de  
Sistemas.

Orientador: Prof. Dr. Josenalde Barbosa de  
Oliveira

MACAÍBA/RN

2018

Universidade Federal do Rio Grande do Norte - UFRN

Sistema de Bibliotecas - SISBI

Catálogo de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede – Escola  
Agrícola de Jundiá - EAJ

Almeida, João Gabriel Quaresma de.

Interface Homem-Máquina para instrumentação e controle de  
ambiência em silos de armazenagem de grãos no contexto da  
Internet das Coisas / João Gabriel Quaresma de Almeida. - 2018.  
65 f.: il.

Monografia (graduação) - Universidade Federal do Rio Grande  
do Norte, Unidade Acadêmica Especializada em Ciências Agrárias,  
Curso de Tecnologia em Análise e Desenvolvimento de Sistemas.  
Macaíba, RN, 2018.

Orientador: Prof. Dr. Josenalde Barbosa de Oliveira.

1. Internet das Coisas - Monografia. 2. Ambiência de silos -  
Monografia. 3. Automação de silos - Monografia. 4. Controle PID  
- Monografia. 5. KNoT - Monografia. I. Oliveira, Josenalde  
Barbosa de. II. Título.

RN/UF/BCZM

CDU 004.41:624.95

João Gabriel Quaresma de Almeida

**Interface Homem-Máquina para Instrumentação e Controle de Ambiente  
em Silos de Armazenagem de Grãos no Contexto da Internet das Coisas**

Trabalho de conclusão de curso de graduação apresentado à Unidade Acadêmica Especializada em Ciências Agrárias da Universidade Federal do Rio Grande do Norte como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Aprovado em: \_\_\_\_/\_\_\_\_/\_\_\_\_

Por:

---

Orientador: Prof. Dr. Josenalde Barbosa de Oliveira

---

Prof. Dr<sup>a</sup>. Laura Emmanuella Alves dos Santos Santana  
de Oliveira

---

Prof. Dr. Leonardo Rodrigues de Lima Teixeira

## AGRADECIMENTOS

Quem diria... O garoto que somente queria saber de jogar bola, que sonhava em ser jogador de futebol (mesmo sendo um jogador normal, embora que tem cada perna de pau jogando mundo a fora), estaria formado pela UFRN em Tecnologia em Análise e Desenvolvimento de Sistemas, pois é, a vida tem cada uma...

Quero agradecer primeiramente a Deus por essa conquista na minha vida, não é todo mundo que tem a oportunidade de se formar e poder evoluir na vida através dos estudos. Agradeço muito, aos meus pais, que mesmo formando uma família bem desajustada e louca, estavam comigo, mesmo eu me isolando no meu mundo e sempre me ensinando valores imprescindíveis para me tornar um ser humano melhor. Aos meus velhos amigos de ensino médio que nessa jornada universitária me ajudaram, mesmo que indiretamente, por meio das resenhas e zueiras, esquecer os problemas que surgiam durante minha caminhada. Obrigado Ítalo, Sabrina, Bob, Bia, Luiza, Paulo Antônio, Neto, Guilherme, Vinícius, Jefferson, Vitória, Gustavo, Heloísa, peço desculpas se esqueci de alguém. Aos meus novos amigos, que conheci na UF/EAJ, que me ensinaram muito mais que programação, mas, me ensinaram a dar valor as pequenas coisas da vida, pois na universidade pública, você se depara com as mais diversas realidades. Fabão, Adriano, Adryel, Tuanny, André Gomes, Brunão, Andrade, Hudson, Aryton (valeu pelas caronas), Laércio (meu canhota favorito), Joel & Iaslan (Sem vocês eu não seria nada na programação), Larysse, Suélliton, Assis, Washington, Arthur, Heriberto, Joffrey OBRIGADO. Se esqueci de alguém desculpa aí.

Ao meu orientador Josenalde, FANTÁSTICO, que com toda sua paciência e conhecimento vasto, me orientou da melhor forma, sempre me tranquilizando e me motivando nos momentos difíceis, até quando eu pensava em desistir. Obrigado grande mestre. À TODOS os professores que saíram e que ficaram até hoje. Taniro tu foi um irmão para mim, fazendo com que eu não desistisse do curso. Grande Leonardo, "pô bicho", muito obrigado também, seus carões me ajudaram a amadurecer muito como pessoa. Laura, muito obrigado, você que deu o *start* para eu entrar nesse mundo da programação, passando da melhor forma possível os seus conhecimentos. NUNCA esquecerei de vocês.

Ao Jiu-jitsu e toda a galera que treinou comigo !!! Ajudaram bastante para eu não enlouquecer nesse curso. OSSSS.

À todos aqueles das equipes do SIPAC e SIGRH da SINFO que me ajudaram a me tornar um programador melhor. E galera do CESAR que ajudaram a desvendar o KNoT.

À uma equipe tumbler de animação, que me fez ficar bem mais leve nessa reta final. MUITO OBRIGADO!!!

*"... It's time to begin, isn't it, I get a little bit  
Bigger, but then, I'll admit, I'm just the  
Same as I was  
Now, don't you understand  
That I'm never changing who I am..."*

It's Time - Imagine Dragons

## RESUMO

O processo da armazenagem de grãos na etapa de pós-colheita é um dos mais importantes para a indústria agrícola. No contexto nacional, temos ainda um campo amplo a ser explorado para aprimoramento desse processo. O presente trabalho tem como objetivo desenvolver um sistema embarcado, no contexto de Internet das Coisas (IoT), utilizando técnicas de controle de processos amplamente utilizados na indústria, para o controle e monitoramento remoto da ambiência de silos em escala laboratorial. A infraestrutura de *hardware* e *software* apresentada contribui para a validação da meta-plataforma nacional KNoT e amplia sua aplicação para a área de agroinformática. Os dados obtidos para as técnicas de controle liga-desliga e Proporcional Integral e Derivativo (PID) sugerem a eficácia da KNoT para transmissão remota de dados de referência para os controladores locais, e aquisição de dados para monitoramento, por meio da Interface Homem-Máquina proposta.

**Palavras-chave:** Internet das Coisas, Ambiência de Silos, Automação de Silos, Controle PID, KNoT.

## ABSTRACT

The process of grain storage in the post-harvest stage is one of the most important for the agricultural industry. In the national context we still have a broad field to be explored to improve this process. This work aims to develop an embedded system in the Internet of Things (IoT) context, using process control techniques widely used in the industry, for the remote control and monitoring of the ambience of silos in laboratory scale. The hardware and software presented contribute to the validation of the national KNoT meta-platform and extends its application to the field of agro-informatics. The data obtained for the on-off and Intergral and Derivative Proportional (PID) control techniques suggest the effectiveness of KNoT for remote transmission of reference data to local controllers, and acquisition of data for monitoring, through the proposed Human-Machine Interface.

**Keywords:** Internet of Things, Silos Environment, Silo Automation, PID Control, KNoT.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Figura A) Silo de base plana / Figura B) Silo em formato de cone . . .	19
Figura 2 – Diagrama de blocos controle Liga-desliga . . . . .	21
Figura 3 – Figura A) . . . . .	22
Figura 4 – Diagrama de blocos controle PID . . . . .	23
Figura 5 – Figura A) Sem o uso do KNoT / Figura B) Com o uso do KNoT . . .	24
Figura 6 – Arquitetura da meta-plataforma KNoT . . . . .	26
Figura 7 – Bancada PP32 de Secagem de Grãos . . . . .	31
Figura 8 – Figura A) Controlador PID N1100 / Figura B) Relé SSR <i>MetalLex</i> TSZC - 40A . . . . .	32
Figura 9 – Arduino UNO (Microcontrolador ATmega328P) . . . . .	33
Figura 10 – Termistor NTC . . . . .	33
Figura 11 – Gráfico termistor NTC . . . . .	34
Figura 12 – Modulo <i>wireless</i> NRF24L01 . . . . .	35
Figura 13 – Raspberry Pi 3 . . . . .	36
Figura 14 – KNoT <i>WebUI Dashboard</i> . . . . .	36
Figura 15 – A) NRF24L01; B) Arduino UNO; C) <i>Raspberry Pi 3</i> . . . . .	38
Figura 16 – Diagrama de Componentes do Sistema Desenvolvido . . . . .	39
Figura 17 – Topologia de rede do Sistema Desenvolvido . . . . .	40
Figura 18 – Disposição dos sensores no silo acadêmico . . . . .	42
Figura 19 – Circuitos para Arduino UNO (Figura A) e <i>Raspberry Pi 3</i> (Figura B) .	43
Figura 20 – Figura A) Etapa de configuração do <i>Cloud Service</i> / Figura B) Etapa de informe das credenciais do usuário . . . . .	44
Figura 21 – Telas do aplicativo <i>Grain Control</i> . . . . .	51
Figura 22 – Temperatura média da massa de grãos usando controle liga-desliga para potência de 25% e $\Delta = 0.1, 0.5, 1.0$ . . . . .	53
Figura 23 – Sinal de controle liga-desliga para potência de 25% e $\Delta = 0.1, 0.5, 1.0$ .	54
Figura 24 – Temperatura média da massa de grãos usando controle liga-desliga para potência de 50% e $\Delta = 0.1, 0.5, 1.0$ . . . . .	55
Figura 25 – Sinal de controle liga-desliga para potência de 50% e $\Delta = 0.1, 0.5, 1.0$ .	56
Figura 26 – Temperatura média da massa de grãos usando controle liga-desliga para potência de 75% e $\Delta = 0.1, 0.5, 1.0$ . . . . .	57
Figura 27 – Sinal de controle liga-desliga para potência de 75% e $\Delta = 0.1, 0.5, 1.0$ .	58
Figura 28 – Temperatura média da massa de grãos usando controle PID para po- tências de 25%, 50% e 75% . . . . .	59
Figura 29 – Sinal de controle PID para potências de 25%, 50% e 75% . . . . .	60

## LISTA DE TABELAS

Tabela 1 – Densidade da massa de Grãos . . . . .	19
Tabela 2 – Recomendação de fluxos de ar para aeração . . . . .	20
Tabela 3 – Comparação entre os módulos <i>wireless</i> com antena externa e embutida	35
Tabela 4 – Resultados de eficiência <i>on-off</i> . . . . .	61
Tabela 5 – Resultados de eficiência PID . . . . .	61

## LISTA DE ABREVIATURAS E SIGLAS

PID	<i>Propocional Integrativo Derivativo</i>
KNoT	<i>KNoT Network of Things</i>
SP	<i>Setpoint</i>
SSR	<i>Solid State Relay</i>
OMC	<i>Organização Mundial do Comércio</i>
CONAB	<i>Companhia Nacional de Abastecimento</i>
IoT	<i>Internet of Things</i>
M2M	<i>Machine to Machine</i>
Embrapa	<i>Empresa Brasileira de Pesquisa Agropecuária</i>
IDE	<i>Integrated Development Environment</i>
IHM	<i>Interface Homem Máquina</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>13</b>
1.1	JUSTIFICATIVA . . . . .	15
1.2	OBJETIVOS . . . . .	16
1.3	ORGANIZAÇÃO DO TRABALHO . . . . .	16
<b>2</b>	<b>REFERENCIAL TEÓRICO . . . . .</b>	<b>18</b>
2.1	APRESENTAÇÃO DO TEMA . . . . .	18
2.2	CONCEITOS BÁSICOS DO CONTEXTO DE APLICAÇÃO . . . . .	18
2.2.1	Armazenagem . . . . .	18
2.2.2	Silos . . . . .	19
2.2.3	Aeração . . . . .	19
2.2.4	Técnicas de controle de processos . . . . .	21
2.3	Meta-Plataforma KNoT . . . . .	24
2.3.1	Arquitetura <i>Hub and Spoke</i> . . . . .	26
2.3.2	Pilares KNoT . . . . .	27
2.3.2.1	KNoT <i>Thing</i> . . . . .	27
2.3.2.2	KNoT <i>Gateway</i> . . . . .	27
2.3.2.3	KNoT <i>Service</i> . . . . .	27
2.3.2.4	KNoT <i>Cloud</i> . . . . .	28
<b>3</b>	<b>TRABALHOS RELACIONADOS . . . . .</b>	<b>29</b>
<b>4</b>	<b>MATERIAIS E MÉTODOS . . . . .</b>	<b>31</b>
4.1	MATERIAIS . . . . .	31
4.1.1	Bancada PP32 de Secagem de Grãos . . . . .	31
4.1.2	Arduino UNO & Arduino IDE . . . . .	32
4.1.3	Termistor <i>Negative Temperature Coefficient</i> (NTC) 10K . . . . .	33
4.1.4	Módulo Wireless NRF24L01 . . . . .	34
4.1.5	<i>Raspberry Pi 3</i> . . . . .	35
4.1.6	<i>KNoT WebUI</i> . . . . .	36
4.1.7	<i>PySerial</i> . . . . .	36
4.1.8	IDE <i>GNU Octave</i> . . . . .	37
4.1.9	<i>KNoT Cloud library</i> . . . . .	37
4.1.10	<i>Firebase Real Time Database</i> . . . . .	37
4.1.11	<i>Android Studio IDE</i> . . . . .	37

4.2	ARQUITETURA . . . . .	38
4.3	MÉTODOS . . . . .	41
<b>5</b>	<b>DESENVOLVIMENTO . . . . .</b>	<b>42</b>
5.1	CONFIGURAÇÃO DO AMBIENTE PARA DESENVOLVIMENTO NA META-PLATAFORMA KNOT . . . . .	42
5.2	DESENVOLVIMENTO DE <i>SKETCHES</i> ARDUINO . . . . .	44
5.3	DESENVOLVIMENTO DE ALGORITMOS <i>JAVASCRIPT</i> PARA ES- CRITA E LEITURA . . . . .	48
5.4	DESENVOLVIMENTO DO APLICATIVO <i>MOBILE GRAIN CONTROL</i>	50
5.5	VALIDAÇÃO DO SISTEMA . . . . .	51
<b>6</b>	<b>RESULTADOS E DISCUSSÃO . . . . .</b>	<b>53</b>
6.1	RESULTADOS COMPUTACIONAIS . . . . .	53
6.1.1	Gráficos controle liga-desliga . . . . .	53
6.1.2	Gráficos controle PID . . . . .	58
6.2	ANÁLISE DOS RESULTADOS . . . . .	60
6.2.1	Resultados do controle de processo liga-desliga . . . . .	60
6.2.2	Resultados controle de processo PID . . . . .	61
<b>7</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS . . . . .</b>	<b>63</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>64</b>

## 1 INTRODUÇÃO

O Brasil, apresenta em sua economia uma base bastante concentrada no agronegócio, seja na agricultura familiar ou na agricultura de grande porte, ou seja, indústrias ou grandes latifundiários. No entanto, mesmo ainda sendo base da economia nacional, de acordo com a Organização Mundial do Comércio (OMC), perdeu espaço no contexto mundial para outros países como Estados Unidos e nações Europeias, ainda que se sustente em terceiro lugar na classificação no mercado internacional, a parte pertencente ao Brasil diminuiu (RURAL, 2017), e muito se deve ainda ao atraso tecnológico de produções, por mais que já se tenha avançado bastante nos últimos anos nos mais diversos aspectos, sejam eles tecnológicos ou até de infraestrutura, ainda há espaço para melhorias.

Como exemplo desse atraso tecnológico, um dos desafios/problemáticas enfrentados pelos produtores é a armazenagem das maiores produções agrícolas do país (milho, soja, arroz e feijão). De acordo com a Companhia Nacional de Abastecimento (CONAB), a capacidade bruta de estocagem agrícola no país é de aproximadamente 143,2 milhões de toneladas de grãos, entretanto cerca de 39,2 milhões de toneladas não são aproveitadas (CILO, 2016), tendo em vista que essa armazenagem é realizada em silos, local onde os grãos são conservados e que estão sujeitos a interferências internas e externas (fatores bióticos e abióticos) no ciclo de produção. E é nessa questão em que se apresenta o desafio, pois a conservação dos grãos é algo determinante, principalmente em termos financeiros. Segundo Carlos Favaro (CILO, 2016): “Essa situação atual exige que os produtores mantenham a alta rotatividade da colheita e uma incrível eficiência nas negociações de venda futura”.

Dada esta necessidade de um melhor monitoramento e controle de ambiência nos silos, inclusive de modo remoto, o conceito de Internet das Coisas (em inglês, *Internet of Things* – IoT), é uma solução promissora, possibilitando o monitoramento e gerenciamento remoto de tarefas a centenas de quilômetros de distância. A IoT redefine a maneira como interagimos com o mundo físico e viabiliza formas através do uso de técnicas computacionais de empreender, fazer negócios, gerenciar infraestrutura pública, prover segurança e organizar a vida das pessoas.

Estima-se que já existam mais de quinze bilhões de dispositivos conectados em todo o mundo, incluindo *smartphones* e computadores. Prevê-se que na próxima década esse valor aumentará drasticamente, atingindo 35 bilhões de dispositivos em 2025, ou 5 vezes a população mundial (MASSRUHÁ; LEITE, 2016). Esse número crescente de dispositivos interligados, conectados, que compartilham informações resulta na geração de novos dados. E a partir desses dados faz-se necessário o uso de técnicas inteligentes de cunho computacional que ofereçam análises que possam gerar novas soluções a problemas das mais diversas áreas da sociedade.

O conceito de IoT está inserido na era da Indústria 4.0. Tal era teve início na indústria de automóveis na cidade alemã de Hannover, que se difundiu nas demais fábricas e segmentos ao redor do globo, devido a completa automatização dos processos vinculados a produção. Constituída por novas competências, técnicas e conhecimentos na indústria global, como *Big Data*, robôs autônomos, integração de sistemas, IoT, segurança de dados, computação em nuvem, manufatura aditiva (impressão 3D), realidade virtual aumentada, comunicação máquina para máquina (M2M), conectividade entre dispositivos móveis (SANCHES, 2018) que também se fazem presente no contexto do agronegócio que agrega outros conceitos como: *smart farming*, agricultura de precisão, agricultura digital e agroinformática (MASSRUHÁ; LEITE, 2016). Além disso, a forma de como se comercializa, produz e presta serviços é algo que se transforma pelo fato das barreiras físicas de produção e comércio encurtarem.

Diante desse cenário da Indústria 4.0, uma pesquisa realizada pela *McKinsey & Company* — empresa de consultoria empresarial americana, reconhecida como a líder mundial no mercado de consultoria empresarial — apontou que o maior desafio encontrado é de caráter cultural e comportamental, pois, são seres humanos que, no final das contas, decidem e influenciam na adoção de novas tecnologias (GORAN; LABERGE; SRINIVASAN, 2017).

Como foi citado anteriormente, as competências relativas ao agronegócio, além de fazerem parte da Indústria 4.0, estão por sua vez, inseridas em uns dos segmentos dessa mesma era: a Agricultura 4.0 (Agro 4.0) ou agricultura digital. Nela se aplicam métodos computacionais e soluções analíticas para processar grandes volumes de dados e construir sistemas de suporte à tomada de decisões de manejo, utilizando, inclusive, técnicas de inteligência computacional para melhores resultados. O objetivo é elevar os índices da indústria Agro, sendo eles de produtividade, eficiência no uso de insumos, redução de custos com mão de obra, melhora da qualidade do trabalho e a segurança dos trabalhadores, diminuindo os impactos ao meio ambiente. Como exemplo disso, a Empresa Brasileira de Pesquisa Agropecuária (Embrapa) através de sua unidade Embrapa Informática Agropecuária é referência no país de tecnologia de informação e comunicação aplicado no campo, além de investir em projetos de tecnologia da informação para organizar e estruturar conjuntos de informações, conhecimentos e tecnologias. Atenta ao cenário do ambiente rural, a Embrapa vem priorizando ações de pesquisa, desenvolvimento, inovação e transferência de tecnologia aos distintos segmentos do agronegócio brasileiro, com linguagem adaptada de modo que produtores rurais, extensionistas, agricultores familiares, cooperativas e outros segmentos da produção agrícola possam assimilá-los com maior facilidade (MASSRUHÁ; LEITE, 2016).

Tendo em vista a inserção portanto de tecnologias que implementam o conceito de

IoT no contexto do armazenamento pós-colheita de grãos é recomendável e está alinhado com o desenvolvimento agrícola mundial, proporcionando melhoras da qualidade do produto final.

## 1.1 JUSTIFICATIVA

Devido aos problemas e desafios enfrentados no processo de armazenamento de grãos que foram citados anteriormente, é pertinente a utilização de tecnologias e soluções que busquem o aprimoramento desse processo visando uma melhora na qualidade do produto e na infraestrutura que trabalha em prol da conservação do estado dele aliado a um menor custo para obtenção desse objetivo, dando um melhor retorno financeiro ao produtor.

Existem no mercado do agronegócio empresas que trabalham no desenvolvimento e possuem soluções que atendam as demandas do setor, em âmbito nacional, como o foco na automatização dos processos quem englobam a etapa de armazenagem.

O *Smart Air* criado pela empresa, da cidade gaúcha de Panambi, FHTech, que consiste em um sistema de termometria automatizada que efetuar de forma precisa o monitoramento dos grão armazenados como também controla automaticamente o sistema de aeração de silos e armazéns. Esse sistema possui um interface gráfica que permite ao usuário o monitoramento da situação dos silos e armazéns, onde se encontram os grãos.

A empresa paranaense *Bry-air* realiza a aplicação de um sistema de diminuição efetiva da umidade do ar dentro do local de armazenagem, garantindo a secagem e estabilidade no local de acordo com a necessidade de cada tipo de produto (grãos). A instalação de desumidificadores de ar proporcionam uma armazenagem para longos períodos reduzindo ao máximo as perdas de produto. Além disso, a implantação desses sistemas economizadores de energia em secadores convencionais reduzem drasticamente o consumo energético. Ademais pode ser aplicá-lo para outros produtos além de grãos, como por exemplo: resinas plásticas e vidros laminados, isso mostra a versatilidade do sistema para diversos outros produtos, fora do contexto agro.

A Senior, empresa catarinense especializada na oferta de *softwares* para gestão empresarial, de pessoas, logística, relacionamento com clientes e controle de acesso, desenvolveu um projeto, pioneiro no Brasil que envolve a automatização dos processos por meio do uso de Internet das Coisas (IoT) e computação cognitiva. O projeto consiste em equipamentos instalados dentro do silo voltado para o armazenamento de grãos de arroz, incluindo emissores baseados em LED e receptores óticos que capturam o volume de grãos armazenados e enviam as informações em tempo real para o sistema desenvolvido pela empresa, que permite o acesso tanto por computador como por *smartphone*. E a partir



de comandos de voz ou texto, os funcionários conseguem solicitar informações analíticas geradas pela sistema acerca dos silos, recebendo os dados e relatórios nos mais diversos dispositivos. Esta espécie de “conversa” é provida pela tecnologia criada pela IBM, *Watson*, que permite dentro do projeto, responder perguntas a respeito do silo. Assim o sistema tem capacidade de processar informações e aprender com elas, simulando o processo de aprendizagem do cérebro humano.

## 1.2 OBJETIVOS

O objetivo geral deste trabalho se fundamenta no conceito de Internet das Coisas (IoT) e visa aprimorar o monitoramento remoto e controle por meio de uma aplicação para dispositivos móveis da temperatura média da massa de grãos, voltado para secagem e armazenagem de grãos em silos, tendo objetivos específicos:

- Validar a plataforma KNoT *Network of Things* (KNoT) para automação de processos; complementar para um sistema mais complexo;
- Desenvolver uma aplicação para dispositivos móveis, intitulada *Grain Control*, na qual o usuário monitora e informa valores de referência (*setpoint*) desejáveis acerca da temperatura média da massa de grãos;
- Monitorar e controlar a temperatura média na massa de grãos, a qual influencia diretamente na durabilidade dos grãos armazenados;
- Fazer uso das técnicas de controle de processos mais comuns na indústria, Liga-Desliga (ON/OFF) e Proporcional Integral e Derivativo (PID).

## 1.3 ORGANIZAÇÃO DO TRABALHO

O Capítulo 2 traz uma revisão de literatura, abordando a definição de alguns conceitos essenciais para viabilizar o entendimento do que foi realizado e abordado no trabalho.

O Capítulo 3 apresenta a plataforma KNoT, tratando dos conceitos que a cercam, além de dispositivos, protocolos e bibliotecas que são utilizados para que seja possível o seu uso.

O Capítulo 4 apresenta os materiais e métodos utilizados neste trabalho, sendo descritas as especificações técnicas dos equipamentos utilizados e etapas de desenvolvimento realizadas.

O Capítulo 5 apresenta os resultados obtidos após testes específicos com comparações e análises entre os controladores de processos utilizados por meio de tabelas, apresentando as variações entre os resultados como forma de verificar sua precisão.

O Capítulo 6 traz as conclusões inferidas a partir dos resultados obtidos e analisa possibilidades de expansão para novas ideias que acarretem em aprimoramento deste trabalho além de inspirar novos trabalhos.

## 2 REFERENCIAL TEÓRICO

### 2.1 APRESENTAÇÃO DO TEMA

O armazenamento de grãos é uma das etapas da pós-colheita. Armazenar grãos não significa a conservação em qualquer lugar. A importância da armazenagem reside no fato de que o armazenamento adequado e bem realizado pode evitar perdas e preservar a qualidade do produto. As perdas da massa de grãos armazenada está relacionada a um ecossistema em que a deterioração e o descarte é o resultado da interação entre variáveis físicas (temperatura e umidade da massa de grãos), químicas e biológicas (FARONI; SOUSA E SILVA, 2009).

Neste trabalho, dentre os variáveis citadas, a variável da temperatura da massa de grãos é o parâmetro em estudo, sendo trabalhado a sua média.

### 2.2 CONCEITOS BÁSICOS DO CONTEXTO DE APLICAÇÃO

Com o intuito de que seja possível a melhor compreensão na temática ao longo dos capítulos seguintes, faz-se necessário apresentar e distinguir os conceitos que serão abordados neles.

#### 2.2.1 Armazenagem

De acordo com Oliveira (2017), no pós-colheita, a tarefa de armazenagem que é realizada tem como foco principal assegurar a qualidade do produto (grãos) que foi colhido. Para que esse objetivo seja de fato alcançado, boas práticas são necessárias para ter um armazenamento de grão com qualidade. Em diversos países, as problemáticas no que diz respeito à colheita, armazenamento e manuseio (secagem, limpeza, etc.) de grãos são pautas de estudo permanente. Tanto em países desenvolvidos ou em desenvolvimento, há a prioridade em reduzir o desperdício devido a falta de silos adequados, falta de limpeza dos locais de armazenamento (silos, armazéns, galpões...), secagem de grãos mal executada, logística e má qualidade do transporte do produto e diversos outros fatores ligados a armazenagem.

Em maio de 2000 foi aprovada a Lei nº 9.973, que trata do tema de armazenagem no Brasil, e é considerada um grande avanço para o setor agrícola. Essa lei consiste em exigências de recursos tecnológicos como termometria, aeração e outras providências, com o objetivo de eliminar perdas e garantir a qualidade dos grãos armazenados (BRASIL, 2000).

### 2.2.2 Silos

Os silos (Figura 1) são construções destinadas ao armazenamento e conservação de grãos secos, sementes, cereais e forragens verdes (CARNEIRO, 1948). Estes por sua vez possuem diversos tipos, tamanhos, configurações arquiteturais e materiais, podendo possuir uma base plana ou em formato de cone, isolado a gás ou não, arejado ou não para entrada de ar.



Figura 1 – Figura A) Silo de base plana / Figura B) Silo em formato de cone

Fonte: Figura A) Portal Metálica Construção Civil / Figura B) Becker Metalúrgica

Já na questão da capacidade de armazenamento, tem sua unidade de medida dada em toneladas ou em metros cúbicos ( $m^3$ ). Para determinar a capacidade de toneladas, multiplica-se a capacidade cúbica do silo pela massa de densidade do grão. A Tabela 1 apresenta algumas densidades de massa de grãos comuns (DAVIS; STAFFORD, 2016):

Tabela 1 – Densidade da massa de Grãos

Grão	Densidade da Massa ( $t/m^3$ )
Trigo	0.80
Canola	0.67
Cevada	0.68
Triticale	0.62
Sorgo	0.73
Milho	0.72
Tremoço	0.80
Brotos de Feijão	0.75
Semente de girassol	0.42
Semente de algodão	0.40

### 2.2.3 Aeração

A aeração de grãos armazenados consiste em realizar a promoção da passagem de baixa vazão de ar natural ou resfriado por meio de ventiladores pela massa de grãos

(HARA, 2003). A utilização dessa técnica tem como foco atender aos seguintes objetivos (SILVA; LACERDA FILHO; DEVILLA, 2008):

- Estabelecer condições que permitam o resfriamento de determinadas áreas da massa de grãos;
- Deixar uniformizada a temperatura na massa interna de grãos;
- Evitar aquecimento e regular a umidade do produto em si;
- Promover a secagem, dentro de limites pré-estabelecidos na literatura, como temperatura e umidade;
- Promover remoção de odores, além de diminuir a proliferação de insetos ou fungos;
- Promover todo o resfriamento da massa de grãos.

Na Tabela 2 abaixo pode-se observar que para cada tipo de silo atrelado a sua finalidade, há uma recomendação de fluxo de ar que pode ser emitido para regiões frias e quentes dentro do silo.

Tabela 2 – Recomendação de fluxos de ar para aeração

Tipo de unidade/finalidade	Fluxo de ar ( $m^3 \text{ min}^{-1} \cdot t^{-1}$ de grãos)	
	Região fria	Região quente
Horizontal/grão seco	0,05 a 0,10	0,10 a 0,20
Vertical/grão seco	0,02 a 0,05	0,03 a 0,10
Pulmão/grãos úmidos	0,30 a 0,60	0,30 a 0,60
Seca-aeração	0,50 a 1,00	0,50 a 1,00

Fonte: SILVA; LACERDA FILHO; DEVILLA (2008)

O uso da aeração, pode ser feito de acordo com as seguintes condições (SILVA; LACERDA FILHO; DEVILLA, 2008):

- Para a umidade relativa superior a 90%, a aeração é recomendável somente em caso de diferença de temperatura entre os grãos e o ar for superior a 5°C;
- Para umidade relativa inferior ou igual a 60%, a aeração somente é recomendável e aplicável em grãos úmidos ou que estejam aquecidos a uma temperatura demasiadamente superior à do ar, necessitando, assim, de resfriamento. Em um outro contexto, pode-se haver uma supersecagem da massa;
- Em resfriamento inferior a 3°C, se torna desnecessária a aeração;

- Em resfriamento entre 3°C e 5°C, se torna recomendável a aeração;
- Já em um resfriamento com uma crescente de temperatura superior a 7°C, se torna o uso da aeração possível, contudo pode provocar condensação do vapor d'água na superfície da massa e nas paredes do silo.

A decisão por ativar o processo de aeração depende de uma medida homogênea que descreva a condição da massa de grãos. Para isto, medidas individuais de sensores são insuficientes. O conceito atual de *sensor fusion* permite unir os dados individuais providos pelos sensores, sejam eles de temperatura, peso, umidade etc., e com isto obter medidas estatísticas, por exemplo, médias, que descrevam melhor o ambiente. No caso de medidas das mesmas grandezas tem-se a fusão homogênea. Quando grandezas diferentes são consideradas, o conceito se aproxima mais de *smart sensors*, pois o processamento de modelos matemáticos torna-se possível (KARIMI, 2018).

Neste trabalho, o conceito de *sensor fusion* homogêneo é utilizado, tendo em vista que sensores de temperatura do mesmo tipo são distribuídos em posições que capturam o valor da temperatura da massa de grãos e o microcontrolador calcula a média da temperatura do ambiente interno do silo acadêmico utilizado.

#### 2.2.4 Técnicas de controle de processos

O controle de processo de sistemas serve para manter o valor de uma variável no seu valor de referência, denominada, *setpoint* (SP). A variável de processo (ou controlada) é a temperatura média interna da massa de grãos (BAYER; ARAÚJO, 2011).

A técnica liga-desliga, consiste de um comparador, em malha-fechada de acordo com o diagrama de blocos genéricos na Figura 2 abaixo:

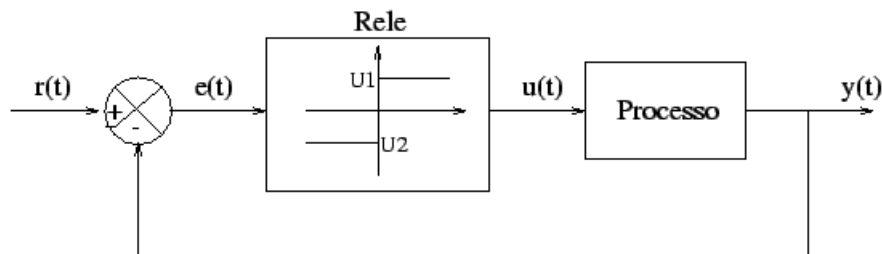


Figura 2 – Diagrama de blocos controle Liga-desliga

Fonte: <http://www.ece.ufrgs.br/~jmgomes/pid/Apostila/apostila/node20.html>

onde  $r(t)$  é o sinal de referência SP. O erro de rastreamento  $e(t) = r(t) - y(t)$  deve ser o menor possível.

O sinal de controle  $u(t)$  pode assumir apenas dois valores, dependendo do valor do erro positivo ou negativo. Sendo assim a ação *bang-bang* pode ser descrita matematicamente como:

$$u(t) = \begin{cases} U_1 & \text{se } e(t) > 0 \\ U_2 & \text{se } e(t) < 0 \end{cases} \quad (2.1)$$

No caso deste trabalho,  $U_2$  é considerado zero, somente havendo o sinal de controle  $U_1$  (Figura 3B), devido a não haver o resfriamento da temperatura de maneira ativa, mas, passiva, por meio do desligamento do relé, que atua sobre a resistência elétrica do ventilador do silo, dependendo do valor de temperatura média atual em comparação com o valor do SP. Com isso ação de *bang-bang* passa a ser chamada de liga-desliga.

Esta ação, é sensível à presença de ruídos e pode ocorrer chaveamentos bruscos quando o sinal  $e(t)$  for próximo de zero. Para evitar este tipo de problema, utiliza-se na prática o que chamamos de controlador liga-desliga com histerese mostrado na Figura 3A (BAZANELLA; SILVA JR., 2005).

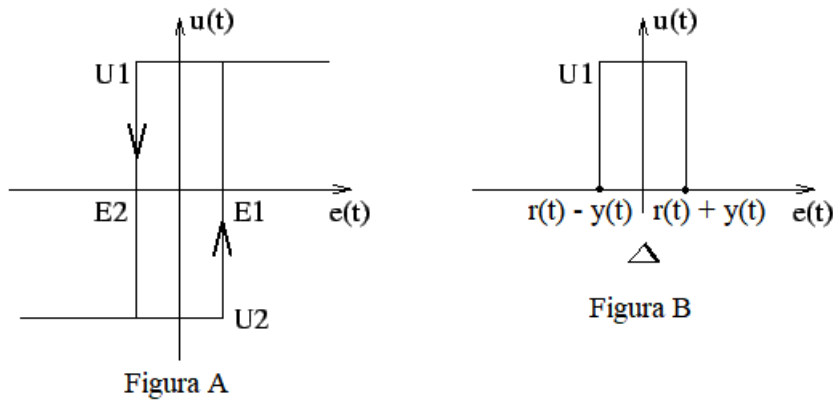


Figura 3 – Figura A)

Fonte: <http://www.ece.ufrgs.br/jmgomes/pid/Apostila/apostila/node20.html>

Nesse tipo de controlador podemos descrever:

- Se  $u(t) = U_1$ , é necessário que o valor de  $e(t)$  desça abaixo de  $-E_2(t)$  para que haja um chaveamento para  $U_2$ ;
- Se  $u(t) = U_2$ , é necessário que o valor de  $e(t)$  ultrapasse o valor de  $E_1$  para que haja um chaveamento para  $U_1$ .

Há portanto uma região  $\Delta$  na qual não há chaveamento. Pode-se associar  $E1 = e(t)$  na Figura 2, a se  $|E1| > \Delta, U1 = U2 = 0$  (Figura 3B).

Já o controle PID utiliza o desvio, a diferença, entre o valor esperado de uma variável de processo e seu valor medido, permitindo ao sistema que funcione de maneira estável no ponto de ajuste desejado com baixo erro (GUERRA, 2009). A instabilidade no ponto de ajuste pode ocorrer pela saturação do sinal de controle. O valor do erro que provoca um sinal de controle igual a 100% define a banda proporcional ( $P_b$ ). Com  $P_b$  alta, a saída  $u(t)$  só irá assumir um valor alto para corrigir o processo se o desvio for alto. Com  $P_b$  baixa, a sinal  $u(t)$  assume valores altos de correção para o processo mesmo para pequenos erros. Em resumo, quanto menor o valor de  $P_b$ , mais forte é a ação proporcional de controle (NOVUS, 2003).

Nesta técnica as ações proporcional, integral e derivativa são combinadas para gerar um só sinal de controle, aproveitando as características particulares de cada uma destas ações a fim de uma melhora significativa do comportamento transitório e em regime permanente do sistema controlado. O sinal de controle gerado pelo controlador PID é expressado matematicamente como:

$$u(t) = K(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt}) \quad (2.2)$$

Analisando a expressão linear (2.2), tem-se três parâmetros de sintonia no controlador: o ganho proporcional  $K$  (ação proporcional), o tempo integral  $T_i$  (ação integral) e o tempo derivativo  $T_d$  (ação derivativa) (BAZANELLA; SILVA JR., 2005). Segue abaixo a Figura 4, para melhor compreensão desse processo.

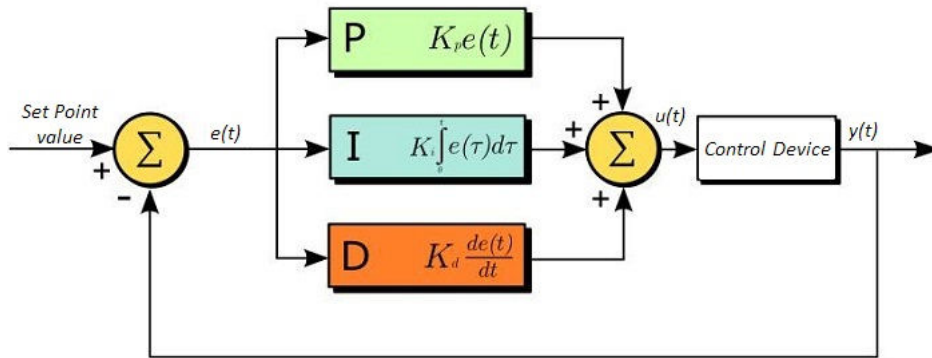


Figura 4 – Diagrama de blocos controle PID

Fonte: ResearchGate – Representação do controlador PID

Comparando (2.2) com a Figura 4, observa-se outra forma de escrever os parâmetros,  $K_p = K$ ,  $K_i = K/T_i$  e  $K_d = KT_d$ . O ganho proporcional  $K_p$  pode ser escrito em função da banda proporcional  $P_b$  como  $K_p = 1/P_b$ .

Levando em conta o contexto desse trabalho, em ambos os controladores o SP é fornecido remotamente por meio do aplicativo *mobile* desenvolvido, sendo transmitido pela



plataforma de IoT, KNoT.

### 2.3 Meta-Plataforma KNoT

O KNoT consiste numa meta-plataforma desenvolvida pelo Centro de Estudos e Sistemas Avançados do Recife - PE (CESAR), lançada em 2017, que implementa o conceito da arquitetura de topologia de rede em estrela denominada *Hub and spoke* voltada para o contexto de IoT (DORNELAS; OLIVEIRA, 2017). A Figura 5 mostra claramente o conceito e objetivo principal da meta-plataforma, que é de oferecer ao desenvolvedor uma interface de abstração e homogeneidade na comunicação entre dispositivos de camadas de baixo e alto nível, onde estes possuem heterogeneidade em seus protocolos de comunicação, fazendo com que eles tenham um ponto de acesso único/comum de comunicação.

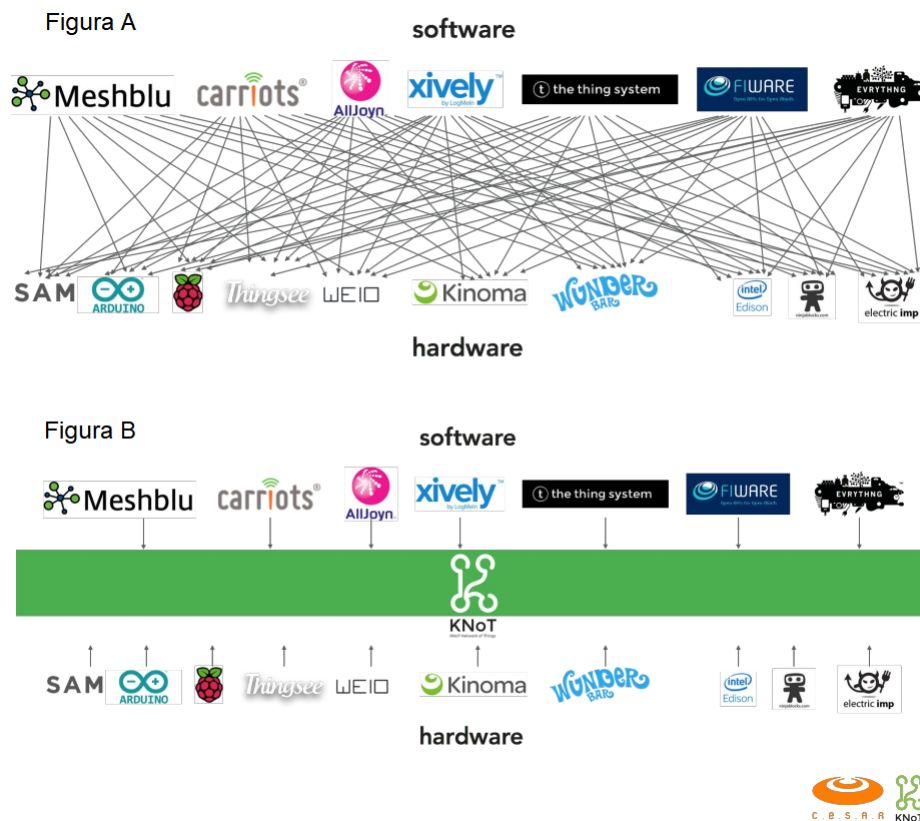


Figura 5 – Figura A) Sem o uso do KNoT / Figura B) Com o uso do KNoT

Fonte: slideshare.net/CESAR/knot-knot-network-of-things

Os principais componentes que fazem parte dessa plataforma são: o *KNoTCloud*, interface que é executada na nuvem (*cloud*), sendo responsável por gerenciar usuários, aplicações e dispositivos (coisas ou *things*), além de armazenar, recuperar e compartilhar os dados desses dispositivos; *KNoTThing*, é o componente de *hardware* e *software* que implementa o acesso a sensores e atuadores conectados fisicamente aos objetos; e o *KNoT*

*Gateway*, componente que funciona como um intermediador entre as camadas de *hardware* e *software* responsável por direcionar as mensagens entre os dispositivos (KNoTThing) e a nuvem de armazenamento de dados (KNoTCloud), abstraindo protocolos e meios físicos de comunicação aliado a conectividade à internet.

Para o desenvolvimento do KNoT, o CESAR ainda firmou parcerias com a plataforma de IoT *Meshblu* e WND, representante *Sigfox* (empresa francesa, que constrói redes sem fio para conectar objetos de baixa potência para transmissão de pequenos dados) no Brasil. Um dos próximos passos da equipe de desenvolvimento é utilizar a tecnologia construída pela *Sigfox* para ser uma das alternativas de comunicação sem fio da plataforma KNoT. O projeto ainda recebe apoio para execução de projetos com o Centro de Informática (CIn) da Universidade Federal de Pernambuco e o Instituto SENAI de Inovação, além de realizar parcerias com a Fundação de Amparo à Ciência e Tecnologia do estado de Pernambuco (FACEPE), por meio do Programa de Apoio a Parcerias para a Inovação Tecnológica e a Formação Qualificada (PITEC).

A plataforma revoluciona os projetos de IoT brasileiros ao ser capaz de conectar as diferentes plataformas existentes, fornecendo um ecossistema nacional de tecnologia da informação e comunicação, numa infraestrutura adequada à realidade das empresas no que se refere às suas necessidades e disponibilidade de gastos com desenvolvimento (SENAI, 2018). Recentemente a plataforma foi utilizada num evento de maratona de programação voltada para IoT, denominado de *Hackthon Semish* que ocorreu no XXXVIII Congresso da Sociedade Brasileira de Computação (CSBC), no ano de 2018, onde neste evento foi determinado aos participantes o desenvolvimento de uma ideia que contemplasse o conceito de IoT atrelado ao uso da plataforma KNoT.

Há ainda outras plataformas de IoT desenvolvidas por empresas e instituições de ensino e pesquisa, como SAP, IBM e Amazon. Uma dessas plataformas que vale destacar é a *mySense IoT++*, focada para agricultura de precisão e no contexto de Agro 4.0, abordada anteriormente. *mySense* representa um ambiente inovador de integração de dados em desenvolvimento para oferecer soluções para os problemas específicos da agricultura de precisão. Desse modo, o ambiente *mySense* oferece suporte para o desenvolvimento de soluções particulares fundamentadas em sensores estáticos, móveis, de proximidade ou remotos com o propósito de fornecer dados/informações essenciais de suporte às mais variadas práticas do mundo agro incluindo também o suporte à algoritmos de tomada de decisão (MYSENSE, 2018).

A escolha do uso da meta-plataforma KNoT neste trabalho, se deu pela motivação de usar uma tecnologia nacional que está em processo de evolução. Além de poder contribuir para o estado da arte da plataforma, que ainda não possui muitos trabalhos

que fazem uso dela e apresentar uma nova solução com a sua utilização.

### 2.3.1 Arquitetura *Hub and Spoke*

A arquitetura de *Hub and spoke* leva em conta dispositivos com baixo poder de processamento e sem IP. Consequentemente, esses dispositivos não têm recursos de processamento suficientes para tomar decisões ou executar algoritmos complexos. Por isso, há a passagem de responsabilidade para tais tarefas à outros elementos arquiteturais, como serviços em nuvem, para armazenamento ou análise de dados. Geralmente, tais dispositivos de baixo custo são microcontroladores muito limitados, que por sua vez, tornam o custo algo viável, como por exemplo, lâmpadas inteligentes. O dispositivo geralmente usa uma comunicação via rádio para se conectar a um *gateway*, que encaminha seus dados para a nuvem (KNOT, 2015). Há três pilares que constituem tal arquitetura: *thing*, *gateway* e *cloud* (Figura 6).

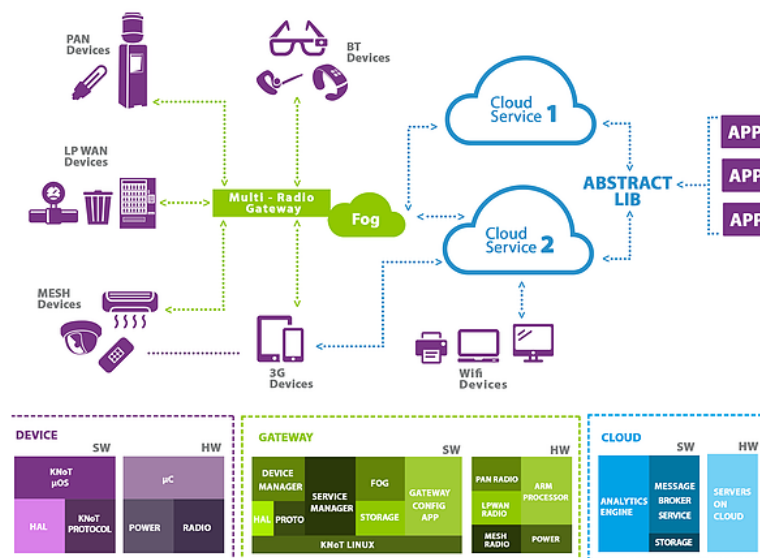


Figura 6 – Arquitetura da meta-plataforma KNoT

Fonte: KNoT CESAR

Os sensores são acoplados ao *thing*, que por sua vez pode ser uma plataforma de prototipagem Arduino, por exemplo, que envia os dados obtidos via rádio transceptor por meio de um protocolo específico para o *gateway*. O *gateway* por sua vez armazena essas informações adquiridas em um repositório local (*fog*) que faz sincronização com a *cloud*. Uma vez os dados disponíveis na nuvem, estes podem ser acessados por aplicações (aplicativos *mobile*, sítios web, etc.). Além disso, essas mesmas aplicações também podem

demandar ações para os *things* que agem como atuadores, realizando assim o caminho inverso, *cloud* até o *thing* (DORNELAS; OLIVEIRA, 2017).

### 2.3.2 Pilares KNoT

#### 2.3.2.1 KNoT *Thing*

KNoT *Thing* consiste em uma biblioteca para o desenvolvimento simplificado no dispositivo, pois sua utilização torna mais fácil a elaboração de algoritmos, sendo necessário poucas instruções de código com funções e objetivos simples, que se baseiam na realização de leitura/escrita de dados e configuração de ações a atuadores. No quesito comunicação com o *gateway*, há a dependência de outras duas bibliotecas, a KNoT *Protocol* que define o protocolo da camada de aplicação, atuando sobre o rádio nRF24L01+ e a KNoT HAL que cria a abstração para os periféricos do hardware. Até o momento, a meta-plataforma somente dispõe suporte para a plataforma de prototipagem Arduino (DORNELAS; OLIVEIRA, 2017).

#### 2.3.2.2 KNoT *Gateway*

O KNoT *Gateway* possui um sistema operacional, que é uma distribuição Linux customizada que contém todo o *software* necessário para o funcionamento do *gateway* dentro da rede. Para a versão (1.03), o hardware suportado é limitado ao pequeno computador de placa única *Raspberry Pi* (DORNELAS; OLIVEIRA, 2017).

#### 2.3.2.3 KNoT *Service*

Neste pilar da plataforma é provido o serviço ao gateway que funciona como interface entre os dispositivos cadastrados da rede local KNoT (*things*) e a *cloud*, realizando o gerenciamento dos dispositivos e facilitando a troca de dados entre *thing* e *cloud*. Essa troca de informações utiliza o mesmo rádio das *things*. Já para comunicação com a internet para que seja possível o sincronismo com a cloud, utiliza *Ethernet* (Arquitetura de interconexão para redes locais), presente na placa Raspberry Pi 3 e suportada pelo sistema operacional que esteja instalado. Em virtude desse uso da *Ethernet* é possível acessar através da rede local TCP/IP uma aplicação web de configuração disponível no gateway, chamada de KNoT Gateway Web UI. Essa aplicação permite visualizar o estado de atuadores ou sensores que tenham sido registrados no algoritmo gravado na *thing* (DORNELAS; OLIVEIRA, 2017).

#### 2.3.2.4 KNoT *Cloud*

O *Cloud* consiste em um serviço *web* que interliga a rede local e as aplicações. A meta-plataforma KNoT, é um *fork* (palavra utilizada na comunidade de *software* livre que se baseia em usar arquivos de alguma outra pessoa para editá-los e depois subir para o repositório desta mesma pessoa) da plataforma *Meshblu* (MESHBLUE, 2017). É embarcado no KNoT *Gateway* funcionando como uma instância local da *cloud* (chamada de *fog*) e também em servidores dedicados atuando como *cloud*. O KNoT *Cloud* possui um repositório de dados que atua como sistema gerenciador de banco de dados (SGBD) NoSQL (não-relacional) orientado a documentos chamado *MongoDB* (MONGODB, 2018). Tanto o *Meshblu*, quanto o MongoDB são aplicações de código aberto (open-source) de licenças flexíveis, sendo assim compatíveis com a proposta da plataforma. Porém ao contrário da plataforma *Meshblu* que, por sua vez, é portátil e tem suporte a uma série de protocolos de comunicação (MQTT, CoAP, HTTP, *Websocket*, AMQP, etc.) que facilita a troca de dados com aplicações web ou *mobile* (DORNELAS; OLIVEIRA, 2017), a plataforma KNoT, foco deste trabalho, até o exato momento, se detém aos protocolos de HTTP e *Websocket* (SOUZA, 2017), havendo muito espaço para desenvolvimento. Já no tocante ao SGBD *MongoDB*, este é referência em portabilidade, performance e escalabilidade.

### 3 TRABALHOS RELACIONADOS

Na literatura existem diversos trabalhos que fazem uso de técnicas computacionais aliadas ao contexto agrário, que abordam a questão da armazenagem, aeração de grãos, secagem, controle de temperatura e umidade, em prol de automatização e maior eficiência desse processo de pós-colheita.

No trabalho realizado por Siviero (2017) é apresentado o monitoramento via ultrassom, temperatura e umidade no armazenamento de grãos com acionamento do sistema de aeração, por meio da união de sensores que são utilizados em conjunto com um microprocessador MSP430 e um sensor de ultrassom que realiza a detecção de grãos dentro do silo. Essa unidade remota é responsável pelo monitoramento de uma pequena região da massa de grãos e se comunica com uma unidade central, micro processada, pelo barramento RS485. Essa mesma unidade apresenta uma interface gráfica que permite ao usuário final acionar o sistema de aeração, visualizar as informações medidas pelas demais unidades remotas, além de prover o controle manual da aeração e permitir a escolha do tipo de grão que está sendo armazenado. Essa escolha de grãos acarreta na alteração dos limiares de temperatura e umidade que o sistema utiliza para o acionamento automático dos ventiladores. Contudo, há nesse trabalho um espaço para aprimoramentos, citados na seção de trabalhos futuros, tais como a proteção dos sensores contra impurezas que permita ao mesmo tempo o seu funcionamento e o estudo no número máximo de unidades remotas possíveis devidamente espaçadas.

Já no trabalho de Silva Jr. (2004) utiliza-se sensores para coletar a temperatura do interior dos silos a umidade do ar externo, que por sua vez abastecem o *software* de gerenciamento e controle automático de aeração existente na empresa, o qual por meio de uma programação pré-definida, determina o momento de ligar/desligar os aeradores. Levando em conta a rotação dos aeradores em cada estágio de frequência e fazendo uso de outro *software*, o MAEV-2000, torna-se possível obter as características de desempenho dos ventiladores em todos os períodos de tempo. É apresentado neste trabalho possíveis complementações, como: o desenvolvimento de sensores de umidade de grãos, além de estudos acerca de compatibilidade eletromagnética em silos metálicos e interferência de harmônicas no sistema elétrico.

Outro trabalho que envolve controle de temperatura e secagem, foi realizado por Pereira Jr. (2018), onde é realizada a aplicação de controlador *Fuzzy* supervisor num secador voltado para indústria de fertilizantes. Em seu trabalho o controlador *Fuzzy* é aplicado no processo de secagem numa planta de produção de fertilizantes, onde foram traçado regras e funções de pertinência desenvolvidas com auxílio e base de conhecimento de especialistas do processo. No que diz respeito a trabalhos futuros, são mencionados o

aprimoramento das funções de pertinência com o intuito de diminuir a variabilidade de resposta e torná-la mais rápida, além de aplicar algoritmos genéticos e *neurofuzzy* para sintonia automática dos parâmetros das mesmas funções de pertinência, fazendo com que haja uma sintonia adaptada a determinado tipo de processo.

Continuando no âmbito de controle de temperatura, um artigo desenvolvido por Zhao (2009) consiste num sistema de monitoramento e controle de temperatura do local de armazenagem de grãos, baseado em comunicação *wireless* e um simples *chip* microprocessador. É apresentado então neste trabalho uma solução *wireless*, adotando um chip de transmissão, também, *wireless*, nRF905 aliado a um circuito integrado MSP430F149, além de um fio que compõe um componente de monitoramento de temperatura digital DS18B20 para construir um sistema de controle e monitoramento de temperatura distribuída.

Enquanto neste artigo realizado por Xiaodong *et al.* (2016) há uma solução mais avançada, com uma implementação de um sistema embarcado que se baseia, também, no monitoramento da ambiência no local de armazenagem de grãos. O sistema faz uso de um microcontrolador S3C6410X e um sistema operacional ARM-Linux embarcado e apresenta arquitetura de três camadas baseada em sistema de monitoramento distribuído, onde é incluído: camada de aplicação, transmissão de dados e aquisição de dados. O sistema é capaz de executar o monitoramento em tempo real de determinados parâmetros do local onde se encontram os grãos, tais como temperatura, umidade, imagem de vídeo e outras informações de relevância para os grãos.

Embora os trabalhos citados apresentem algum monitoramento e controle, eles não utilizaram ou são desenvolvidos no contexto de IoT, foco deste trabalho. Neste trabalho tem-se como objetivo o desenvolvimento de uma interface homem-máquina (aplicativo *mobile*), com o foco e uso de plataformas de desenvolvimento IoT, que proporcione ao usuário a facilidade de poder monitorar e controlar a temperatura média do local de armazenagem da massa de grãos (em escala laboratorial) por meio de técnicas de controle de processos. Sendo assim, pode-se afirmar que esses são os principais diferenciais deste trabalho para os demais citados acima.

## 4 MATERIAIS E MÉTODOS

### 4.1 MATERIAIS

Nesta seção serão abordados os materiais utilizados ao longo do desenvolvimento do sistema de IoT, desde a parte que engloba as ferramentas da camada de *hardware* até a camada de *software*, caracterizando assim um projeto de sistema embarcado.

#### 4.1.1 Bancada PP32 de Secagem de Grãos

Neste trabalho foi utilizado uma bancada fabricada pela empresa de equipamentos eletrônicos T&S (Figura 7), de cunho acadêmico e disponível no Laboratório de Sistemas Embarcados e Eletrônica (SEELAB) da Escola Agrícola de Jundiaí, para a validação em escala laboratorial do controle e monitoramento de temperatura da massa interna de grãos dentro de um silo feito de acrílico, sob uma plataforma de metal, que em sua base possui um soprador que realiza a emissão de ar.

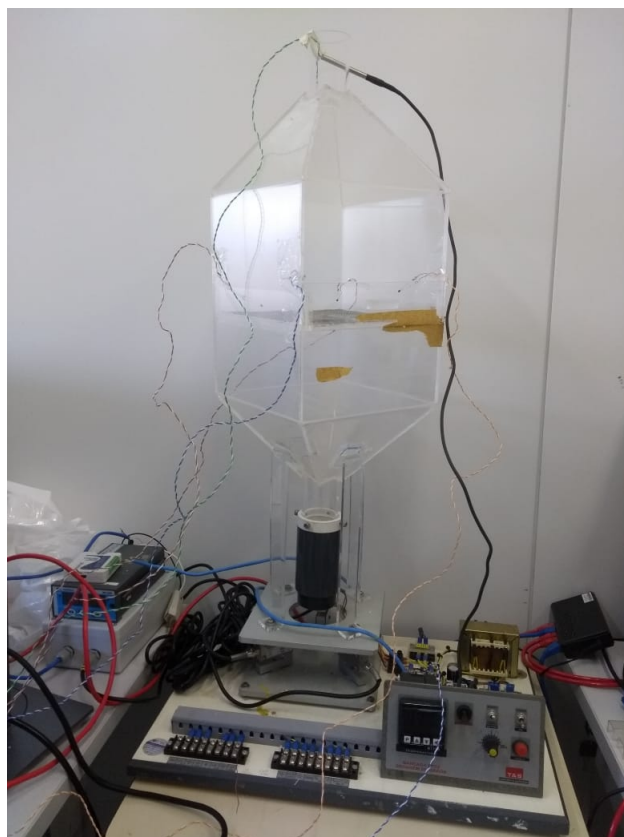


Figura 7 – Bancada PP32 de Secagem de Grãos

O princípio de funcionamento da bancada se dá por meio de um soprador de fluxo de ar ajustável, por um potenciômetro, localizado no painel da bancada, que pela sua resistência elétrica, ativada via relé de estado sólido (SSR) que recebe um sinal de controle



enviado pelo controlador PID, emite o ar que circula por todo espaço interno do silo passando pela gaveta de grãos localizada dentro do silo e evacuando ao chegar a saída de ar localizado no topo do material. Esta gaveta possui um design tal que permite uma melhor dispersão do fluxo de ar a fim de atingir todo o espaço de secagem. Sua principal aplicação é facilitar o estudo de técnicas experimentais de modelagem e controle de sistemas dinâmicos, tais como PID e liga-desliga (T&S, 2005).

Esta bancada possui como seus principais componentes o controlador PID embarcado, o N1100 fabricado pela empresa NOVUS, que possibilita, configurar os parâmetros de PID e informar do SP para o controle da temperatura interna do silo, e o relé SSR *MetalLex* TSZC - 40A (Figura 8).



Figura A



Figura B

Figura 8 – Figura A) Controlador PID N1100 / Figura B) Relé SSR *MetalLex* TSZC - 40A

Fontes: Figura A) Manual Controlador N1100/ Figura B) eletropecas.com

O SSR ativa a resistência elétrica que se localiza na parte inferior do soprador de ar, sendo o elemento ativo para o aumento da temperatura interna do silo. Este tipo de relé não funciona de maneira eletromecânica convencional, mas, sim por meio óptico, que ao receber um pulso elétrico, aciona um componente eletrônico chamado TRIAC, que regula o quanto de corrente elétrica alternada fluirá pela resistência.

#### 4.1.2 Arduino UNO & Arduino IDE

O Arduino (Figura 9) é líder mundial do que diz respeito a ecossistema de hardware e *software* livre, além de ser a ferramenta mais popular para o desenvolvimento de produtos na área de IoT (ARDUINO, 2018).

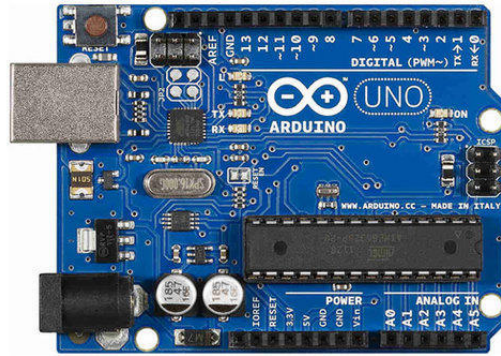


Figura 9 – Arduino UNO (Microcontrolador ATmega328P)

Fonte: indiamart.com

O microcontrolador (ATmega328P) foi utilizado para processar o sinal de controle baseado nas estratégias de controle liga-desliga e PID, e enviar este sinal ao SSR para controlar a temperatura média interna do silo.

E por meio da biblioteca *KNoTThing* que é possível transmitir os dados coletados e receber dados via módulos *Wireless* NRF24L01 conectados a *Thing* e *Gateway*.

#### 4.1.3 Termistor *Negative Temperature Coefficient* (NTC) 10K

O Termistor NTC (Figura 10) é um tipo de componente eletrônico que tem sua resistência elétrica diminuída termicamente de acordo com o ambiente, ou seja, apresentam um valor de resistência para cada temperatura absoluta (Figura 11). Devido ao seu excelente desempenho com longa estabilidade e precisão térmica, os termistores NTC, com faixas de temperatura de  $-50^{\circ}\text{C}$  até  $300^{\circ}\text{C}$  (em vidro), são os mais indicados para medir e controlar temperatura com eficiência em qualquer equipamento (ADD-THERM, 2018).



Figura 10 – Termistor NTC

Fonte: instructables.com

A obtenção do valor da temperatura do ambiente interno do silo, tem a utilização do método matemático de *Steinhart-Hart* (4.1). Tal método é abstraído por meio do uso de uma biblioteca para Arduino, chamada de "*Thermistor.h*", que por meio de uma única

função retorna o valor da temperatura.

$$T = 1/(A + B\ln(R) + C(\ln(R))^3) \quad (4.1)$$

- $T$  é a temperatura (em Celsius)
- $R$  é a resistência elétrica na temperatura  $T$  (em ohms)
- $A$ ,  $B$  e  $C$  são os coeficientes desta equação que dependem diretamente do tipo de construção, material e margem de temperatura. No arquivo da biblioteca "*Thermistor.h*", a definição dos valores desses coeficientes são:  $A = 0.001129148$ ,  $B = 0.000234125$  e  $C = 0.0000000876741$

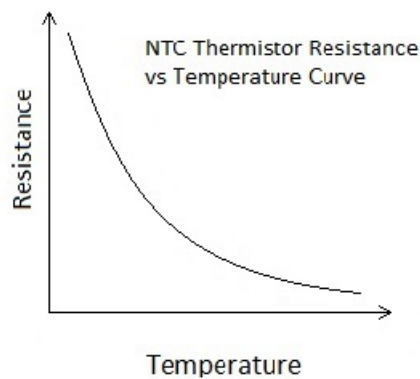


Figura 11 – Gráfico termistor NTC

Fonte: components101.com

#### 4.1.4 Módulo Wireless NRF24L01

Esse módulo *wireless* (Figura 12) é um simples chip de comunicação via rádio com frequência *Industrial Scientific and Medical* (ISM) de 2.4 - 2.5 GHz, com um baixo consumo de energia, somente 9.0mA, possui antena embutida e permite uma configuração descomplicada através de programação utilizando a interface *Serial Peripheral Interface* (SPI) (NORDIC, 2006).

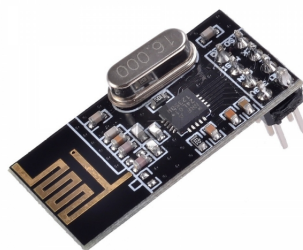


Figura 12 – Modulo *wireless* NRF24L01

Fonte: buildbot.com.br

O fato de possuir uma antena embutida gera desvantagens em comparação a um módulo que possua uma antena externa. Segue abaixo a Tabela 3 com as comparações entre os dois tipos.

Tabela 3 – Comparação entre os módulos *wireless* com antena externa e embutida

	Módulo com antena externa	Módulo com antena embutida
Alcance	1km em 250kbps, 750m em 1MB/s e 520m em 2MB/s	100 metros em 250kbps
Taxa de dados	250kbps a 2MB/s	250kbps a 0.25 MB/s
Temperatura de trabalho	-45 graus a 125 graus	-40 a 85°C
Sensibilidade	-95dBm	-85dBm

Fonte: natalmakers.lojaintegrada.com.br

#### 4.1.5 *Raspberry Pi 3*

O *Raspberry*, Figura 13 abaixo, é uma espécie de *desktop* do tamanho equivalente a um cartão de crédito, que é capaz de operar sistemas operacionais como Windows ou distribuições Linux gravadas em um cartão SD inserido no dispositivo, como a sua distribuição oficial, *Raspbian*. Além disso este dispositivo possui interfaces para periféricos como teclado, *mouse*, *bluetooth* e conexão cabeada ou *wi-fi* com a Internet (RASPBERRY, 2018).

O *hardware* do *Raspberry* é algo a se ressaltar, essa versão utilizada, Pi 3, conta com memória RAM de 1GB, processador com um *clock* de 1.4 GHz de quatro núcleos.

As aplicações que este dispositivo pode ser utilizado são as mais diversas, seja para utiliza-lo como uma central de mídia, servidor de nuvem, estação meteorológica e/ou realizar automação residencial.

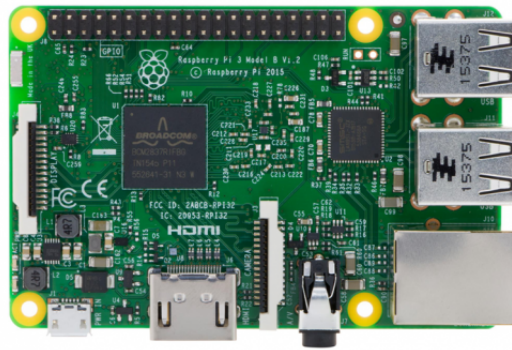


Figura 13 – Raspberry Pi 3

Fonte: element14.com

#### 4.1.6 *KNoT WebUI*

*KNoT WebUI* (Figura 14) é a interface web para o desenvolvedor acompanhar os parâmetros que foram definidos no algoritmo embarcado no Arduino. No caso deste trabalho, os parâmetros levados em conta foram a média da temperatura interna da massa grãos e o SP da temperatura desejada.

Para o acesso a interface necessita-se de configurações prévias, tais como: definição de *host* e porta, além de informe de e-mail e senha para o acesso.

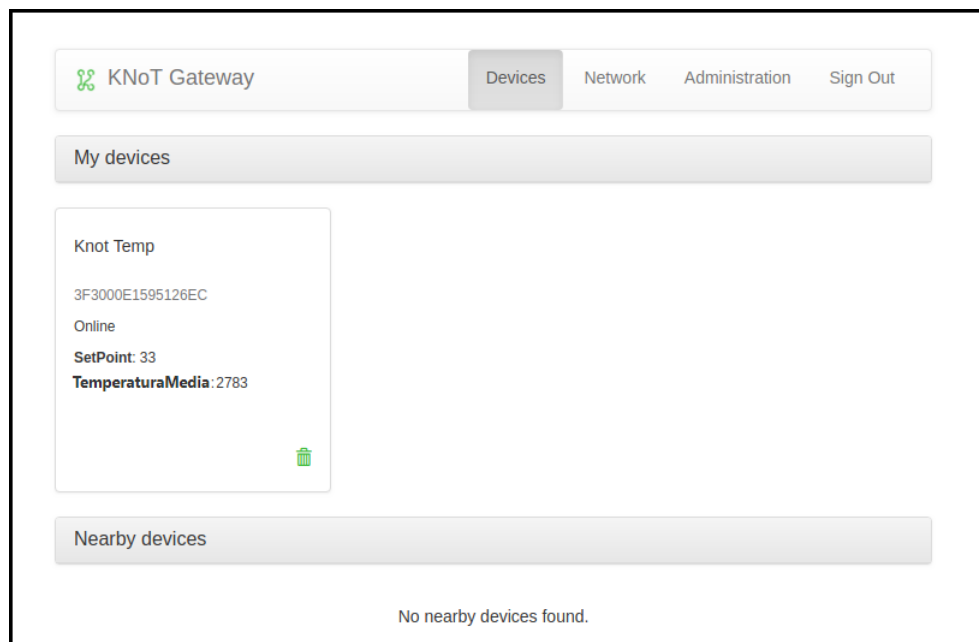


Figura 14 – KNoT *WebUI Dashboard*

Fonte: Do autor

#### 4.1.7 *PySerial*

A biblioteca *PySerial* (PYSERIAL, 2015) é desenvolvida na linguagem de programação *Python* e tem como objetivo ler dados que são enviados para a porta serial do

computador. No caso deste trabalho, são coletados os dados da temperatura interna, SP, e o sinal de controle do relé, para que tais informações sejam gravadas em um arquivo de texto (.txt). Posteriormente, com os dados gravados neste arquivo, os gráficos foram gerados via IDE *GNU Octave*.

#### 4.1.8 IDE *GNU Octave*

Este IDE é uma ferramenta bastante poderosa que é capaz de resolver problemas comuns de álgebra linear numérica, encontrar raízes de equações não lineares, realizar processamentos digitais de imagens, dentre outras tarefas (OCTAVE, 2018). Nesse trabalho o uso desta ferramenta se deu exclusivamente para a geração de gráficos a partir de arquivos de texto, fornecidos pela biblioteca, já mencionada, *PySerial*. Tais gráficos serviram para análise e comparação dos resultados obtidos neste trabalho, que serão abordados no capítulo 6.

#### 4.1.9 *KNoT Cloud library*

A *KNoT Cloud library*, consiste em uma biblioteca desenvolvida em linguagem de programação *JavaScript*, que tem como objetivo poder fornecer uma API que possui métodos para o consumo e edição de dados que se localizam no KNoT Cloud (Ver secção 3.2.4). Neste trabalho o uso dessa biblioteca se constituiu no envio de dados e a mudança de SP da temperatura para o serviço de armazenagem remota de dados em tempo real, provido pela plataforma de desenvolvimento da Google, *Firebase*.

#### 4.1.10 *Firebase Real Time Database*

*Firebase Real Time Database*, é um dos diversos serviços fornecidos pela plataforma *Firebase*, adquirida pela Google, considerada uma solução completa de *back-end* tanto para o desenvolvimento *mobile* quanto *web*. Este serviço provê um banco de dados não-relacional (NoSQL) utilizado para armazenar e sincronizar dados em tempo real, em formato JSON (AVRAM, 2018). Levando em conta o contexto desse trabalho, o uso desse serviço se justifica para armazenar os dados de temperatura e valor de SP, para que estes sejam acessados e modificados por qualquer aplicação, neste caso, *mobile*. Uma vez que o KNoT *cloud* não estava hospedado em um servidor.

#### 4.1.11 *Android Studio* IDE

O *Android Studio* é um IDE oficial para o desenvolvimento de aplicativos *Android*, baseado no *IntelliJ IDEA* (ANDROID, 2018). Essa ferramenta serviu para o desenvol-

vimento em linguagem de programação *Java* (JAVA, 2018) do aplicativo *mobile* deste trabalho, o *Grain Control*.

## 4.2 ARQUITETURA

A arquitetura do sistema desenvolvido pode ser compreendida no âmbito de materiais de *hardware* e *software*, que foram utilizados ao longo do desenvolvimento, pela Figura 15 que mostra como, na prática, foi configurado o ambiente de desenvolvimento do sistema, pelo diagrama de componentes (Figura 16) e pela topologia de rede que representa as comunicações entre os dispositivos e serviços (Figura 17).

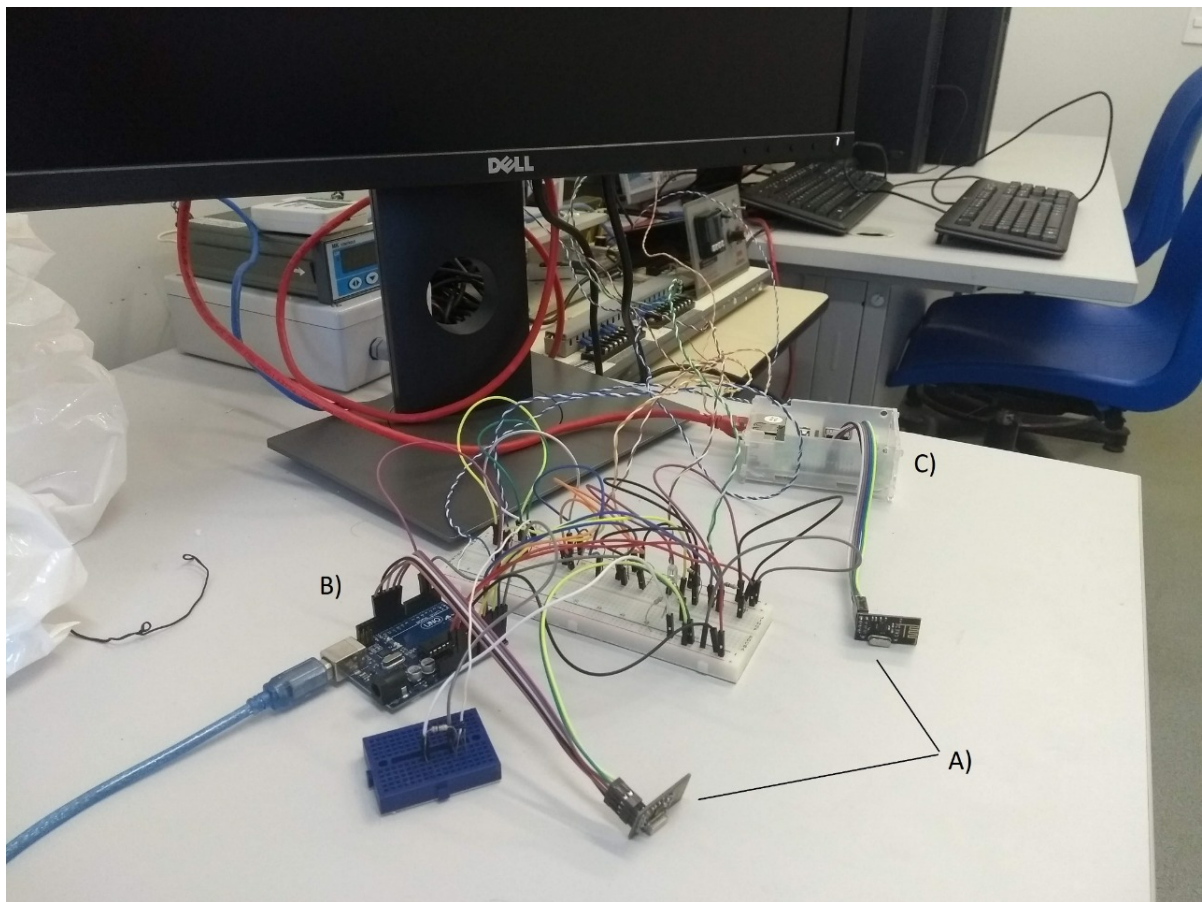


Figura 15 – A) NRF24L01; B) Arduino UNO; C) *Raspberry Pi 3*

Fonte: Do autor

Na figura acima apresenta como foram feitas as comunicações entre os dispositivos de *hardware* para que fosse possível a captura dos dados e envio dos dados para a camada de *software*, no caso a *cloud*.



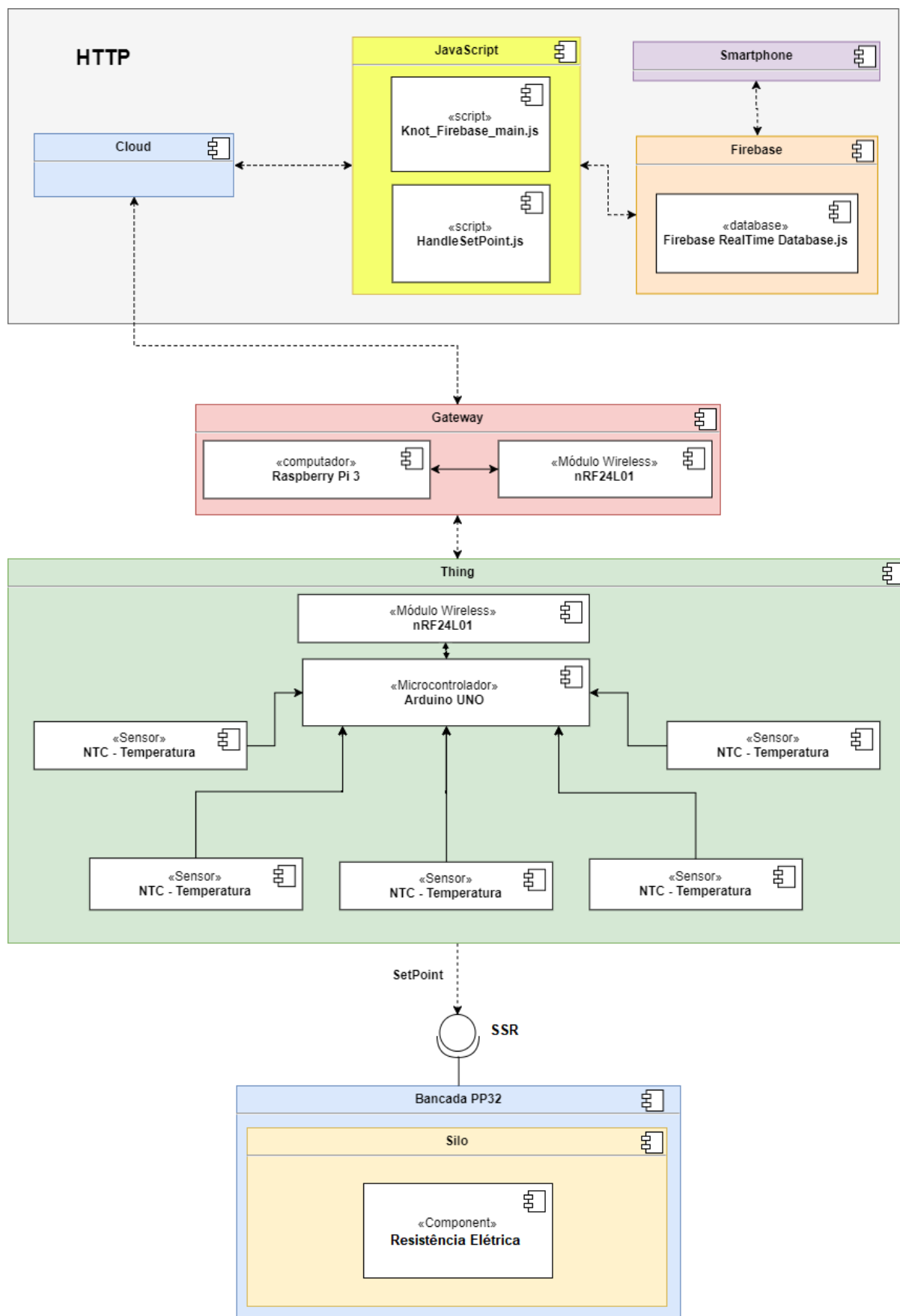


Figura 16 – Diagrama de Componentes do Sistema Desenvolvido

Fonte: Do autor



No diagrama de componentes acima é apresentado, com uma visão geral, a comunicação entre cada componente das camadas de baixo e alto nível que constituem o sistema.

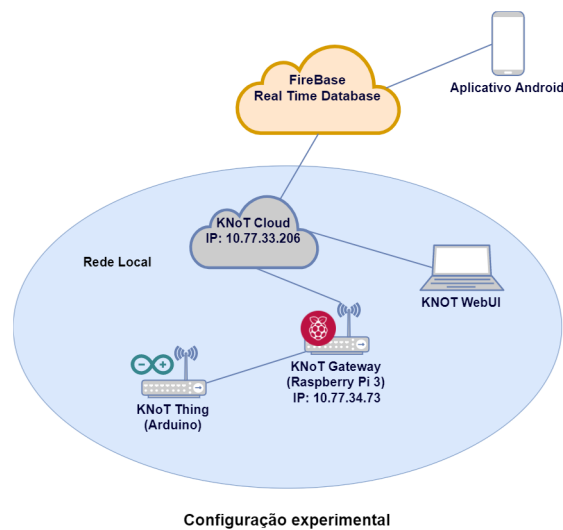
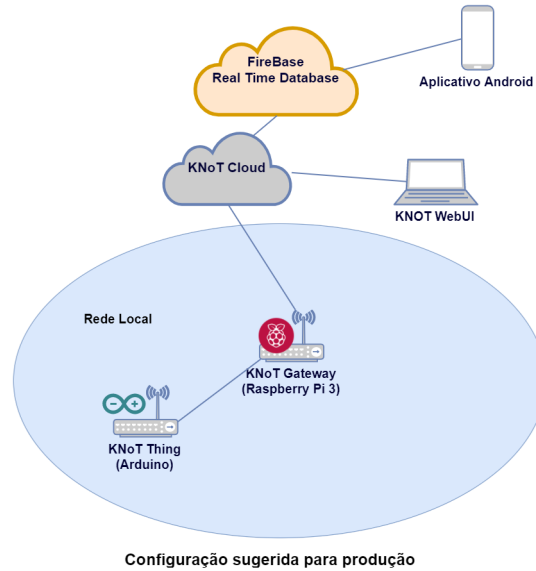


Figura 17 – Topologia de rede do Sistema Desenvolvido  
Fonte: Do autor

A topologia de redes mostrada acima, demonstra como a comunicação dentro de um ambiente em rede se dá entre a *thing*, *gateway*, *cloud* e *Firebase Realtime Database* em uma configuração sugerida para produção e outra experimental (usada neste trabalho), onde a principal diferença entre as duas é o acesso a *cloud* depende diretamente da comunicação

com o serviço *Firebase RealTime Database* na configuração experimental, enquanto que na de produção não há essa dependência direta.

### 4.3 MÉTODOS

Tendo em vista os materiais apresentados, foram realizados a partir do uso deles os seguintes métodos para o desenvolvimento do sistema:

- Configuração do ambiente para desenvolvimento na plataforma KNoT
- Desenvolvimento de *Sketches* Arduino
- Desenvolvimento de Algoritmos *JavaScript* para escrita e leitura
- Desenvolvimento do Aplicativo *mobile Grain Control*
- Validação do Sistema

No próximo capítulo será detalhado como se desenvolveu/executou cada método citado acima.

## 5 DESENVOLVIMENTO

Neste capítulo será discorrido como se deu cada etapa do processo do desenvolvimento do sistema, desde a etapa de configurações do ambiente para o uso correto da meta-plataforma KNoT, passando pelo desenvolvimento dos algoritmos da camada de baixo nível (KNoT *Thing*) e alto nível (consumo de dados da *cloud* pela aplicação *mobile*).

### 5.1 CONFIGURAÇÃO DO AMBIENTE PARA DESENVOLVIMENTO NA META-PLATAFORMA KNOT

Começando pela captura dos valores de temperatura do ambiente interno do silo acadêmico. Se deu a partir da aplicação do conceito de *sensor fusion*, fazendo a utilização de cinco sensores NTC com o intuito de calcular a média aritmética da temperatura interna do silo. Para isso, quatro deles foram distribuídos nos vértices da gaveta e um na abertura, localizada no topo do silo como se pode conferir na Figura 18.

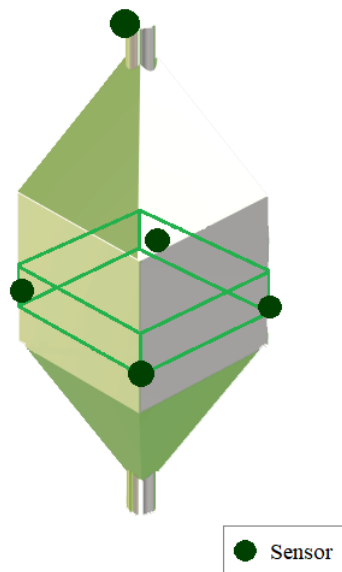


Figura 18 – Disposição dos sensores no silo acadêmico

Fonte: Do autor

Na questão da comunicação entre a *thing* e *gateway* foram utilizados dois módulos NRF24L01 (4.1.4) ao longo do trabalho, conectados tanto ao Arduino UNO como ao Raspberry PI, que por sua vez serviam para a comunicação (troca de informações) entre ambos os dispositivos. A Figura 19 demonstra como foi feita a conexão do módulo entre os dispositivos.

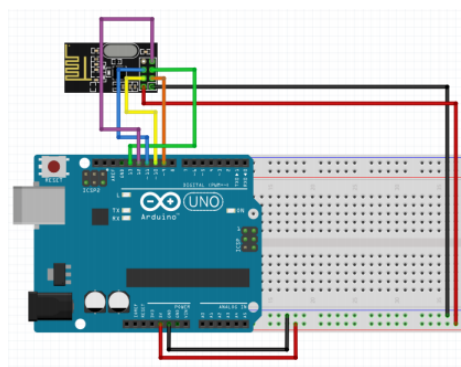


Figura A

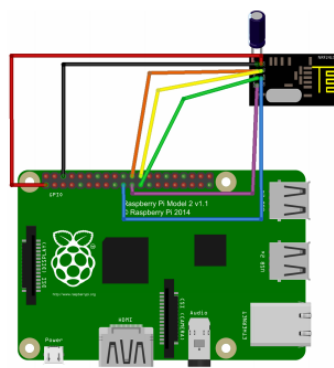


Figura B

Figura 19 – Circuitos para Arduino UNO (Figura A) e *Raspberry Pi 3* (Figura B)

Fonte: KNoT CESAR. Guia do usuário

A configuração do ambiente de desenvolvimento foi realizada no sistema operacional *Linux Ubuntu 18.04 LTS*, devido ao *KNoT Cloud* somente ter suporte para *Linux* ou *Linux Server*. A instalação da *Cloud* foi feita primeiramente realizando o *download* do repositório localizado no *Github* do CESAR, denominado "knot-cloud-source". Após o *download*, basta seguir os passos do guia de usuário da plataforma, que pode ser encontrado no *site* do KNoT (<https://www.knot.cesar.org.br>). Concluída a instalação da *Cloud*, liga-se então o *Raspberry Pi 3* (estando já previamente instalado com a distro linux customizada da plataforma, também disponível no *site*), que atua como *Gateway*, como já foi mencionado. Depois de ter ligado ele, aguarda-se alguns segundos e digita-se o usuário e senha, que será pedido pelo terminal, ambos "root".

Partindo para o acesso ao KNoT *WebUI*, o primeiro passo é acessar o endereço "knot.local". Uma vez acessado, deve-se informar no primeiro passo o endereço IP como *Host* e Porta padrão 3000. O segundo passo e último é a criação de um usuário (proprietário do *gateway* e *things* conectados esse *gateway*), provendo um e-mail e senha (Figura 20).

KNoT Gateway

Step 1. Cloud service

Set the KNoT Cloud server address that this gateway will connect to. This is the server where your device data and user credentials will be stored.

Hostname: 127.0.0.1

Port: 3000

Previous Next

Figura A

KNoT Gateway

Step 2. User credentials

Create a user account to sign in on this gateway. This account will be saved in the KNoT Cloud server configured in the previous step.

E-mail: j.quaresmasantos\_98@hotmail.com

Password: .....

Password confirmation: .....

Previous Finish

Figura B

Figura 20 – Figura A) Etapa de configuração do *Cloud Service*/ Figura B) Etapa de informe das credenciais do usuário

Fonte: Do autor

## 5.2 DESENVOLVIMENTO DE *SKETCHES* ARDUINO

Os dois *sketches* que foram desenvolvidos para coletar os valores de cada um dos cinco sensores instalados no silo da bancada PP32, e com a coleta desses valores, aplicar o conceito de *Sensor Fusion*. Cada um desses dois *sketches* aplicam uma técnica de controle de processo diferente. O primeiro realiza o controle usando a técnica de liga-desliga com histerese. No trecho de código abaixo pode se compreender como é realizado essa técnica.

---

```

1  if (avgControl < setPoint_value - deltaTemp) {
2      stats = 1;
3      digitalWrite(3, HIGH);
4  } else if (avgControl > setPoint_value + deltaTemp) {
5      stats = 0;
6      digitalWrite(3, LOW);
7  }

```

---

A variável "*avgControl*", representa o valor da média de temperatura do ambiente interno do silo, enquanto que a variável "*deltatemp*" é equivalente ao valor de histerese,

que é definida no começo do algoritmo. E este valor é alterado de acordo com os testes que foram realizados, podendo assumir valores de 0.1, 0.5 e 1.0.

A técnica de PID utiliza uma biblioteca específica chamada "*AutoPID*" voltada para o controle de processo PID. No código é possível definir os parâmetros de  $K_p$ ,  $K_i$  e  $K_d$  para um controle mais preciso e menos oscilatório da variável de temperatura. Pode-se compreender no trecho de código abaixo o funcionamento dessa técnica.

---

```
1 #include <AutoPID.h>
2 //...
3 #define OUTPUT_MIN 0
4 #define OUTPUT_MAX 255
5 #define KP 1/0.6 //Pb = 0.6
6 #define KI KP/1.77 //Ti = 1.77
7 #define KD KP*6 //Td = 6
8 //...
9 AutoPID myPID(&temperature, &setPoint, &outputVal, OUTPUT_MIN, OUTPUT_MAX, KP,
10 KI, KD);
11 //...
12 void loop(void) {
13     long start = millis();
14     thing.run();
15
16     setPoint = setPoint_value;
17     updateTemperature();
18
19     myPID.run();
20     //...
21 }
```

---

No trecho acima, nas linhas 5 a 7, há a definição dos parâmetros KP, KI e KD, a escolhas dos valores se deu pela obtenção desses valores via acesso as configurações do controlador NOVUS, após diversas tentativas e erros em busca dos melhores parâmetros. Na função "*loop(void)*" é chamada outra função, a "*myPID.run()*" (linha 18), que executa o processo de cálculos matemáticos do PID e discretizações da equação (2.2), tendo como base os parâmetros que foram definidos.

Contudo, para que o uso da técnica PID de fato ocorre-se para a simulação do controle da temperatura média da massa de grãos, a primeira solução foi utilizar o SP remoto através de comunicação com o Arduino, no entanto essa comunicação não foi possível devido a versão utilizada, neste trabalho, do controlador PID da NOVUS não

permitir tal comunicação, logo foi feito um *bypass* (contorno da situação), no controlador usado na bancada, e se implementou o PID no próprio Arduino. Com o PID da bancada em modo *autotune*, obteve-se os valores iniciais dos parâmetros  $K$ ,  $T_i$  e  $T_d$  (2.2), que por sua vez, foram utilizados no algoritmo de PID (demostrado acima).

---

```
1 TCCR2A = _BV(COM2A1) | _BV(COM2B1) | _BV(WGM20);
2 TCCR2B = TCCR2B & B11111000 | B00000111;
```

---

O trecho de código acima tem como objetivo a manipulação nos registradores do ATmega328P, sendo realizada diminuição da frequência para 30Hz. Uma vez que para que fosse possível a alteração do valor da resistência via Arduino, já que o controlador PID da NOVUS não seria mais utilizado para executar controle da média de temperatura, foi necessário o que o pino de número três (PWM) do microcontrolador fosse ligado ao relé, porém, para o controle da temperatura média do ambiente interno por PID, especificamente, foi necessário a diminuição da frequência em que se opera na saída do pino. Visto que a essa frequência padrão equivale a 490Hz (SOUZA, 2014) e que em contra partida a frequência do SSR é inferior.

Para que então todo esse processo funcione da maneira correta, se faz presente o uso da biblioteca da plataforma KNoT para Arduino, KNoT *Thing*, que fornece funções que facilitam a vida do desenvolvedor, para que seja possível enviar e receber dados para *Cloud*. Funções essas que registram o valor da média de temperatura obtida e o SP que está suscetível a alterações do usuário por meio do aplicativo.

---

```
1 //...
2 #include <KNoTThing.h>
3 //...
4 KNoTThing thing; // KNoTThing instance
5 //...
6 static int temp_read_1(int32_t *val) {
7     *val = temp_value_1;
8     return 0;
9 }
10
11 static int temp_write_1(int32_t *val) {
12     temp_value_1 = *val;
13     return 0;
14 }
15 //...
16 void setup(void) {
```

```

17  thing.init(THING_NAME);
18  //Registrar SetPoint
19  thing.registerIntData(SETPOINT_NAME, SETPOINT_ID,
20                        KNOT_TYPE_ID_SPEED, KNOT_UNIT_SPEED_MS,
21                        setPoint_read, setPoint_write);
22  //Registrar Temperatura Media
23  thing.registerIntData(TEMP_NAME_1, TEMP_ID_1,
24                        KNOT_TYPE_ID_TEMPERATURE, KNOT_UNIT_TEMPERATURE_C,
25                        temp_read_1, temp_write_1);
26
27  /* Envia dados a cada dez segundos */
28  thing.registerDefaultConfig(TEMP_ID_1,
29                              KNOT_EVT_FLAG_TIME, 10, 0, 0, 0, 0);
30  //...
31  }
32
33  void loop(void) {
34  //...
35    thing.run();
36  //...
37  }

```

---

Nas linhas de 2 a 10 tem-se duas funções que uma representa a escrita ("*temp\_write\_1*") e a outra a leitura de dados ("*temp\_read\_1*") obtidos via sensores. Dentro da função "*setup(void)*" há na linha 17, a chamada da função "*init(THING\_NAME)*" que realiza a tarefa de registrar o nome da *thing* na *Cloud*, ao mesmo tempo que, as funções "*registerIntData(...)*" (linhas 19 e 23) e "*registerDefaultConfig(...)*" (linha 28), possuem como objetivo a primeira, registrar o nome e valores, de tipo primitivo *int32\_t*, das variáveis de SP e temperatura média e a segunda, por sua vez tem como propósito, a atualização do valor de temperatura média em um intervalo de dez segundos.

Contudo ao longo do desenvolvimento foram constatadas limitações acerca da quantidade de variáveis que se podem registrar no KNoT, que somente permitia o registro de duas variáveis: temperatura média e SP remoto em uma única *thing*, limitando a possibilidade de adição de mais sensores e atuadores. Uma solução possível seria a utilização de mais de uma *thing* (Arduino) para realização de tarefas diferentes.



### 5.3 DESENVOLVIMENTO DE ALGORITMOS *JAVASCRIPT* PARA ESCRITA E LEITURA

Foram desenvolvidos dois algoritmos que trabalham em paralelo, onde um é encarregado de coletar e enviar os valores de temperatura média, providos pela *Cloud* para o banco de dados em *real time* localizado na plataforma *Firebase*. No trecho abaixo do algoritmo que foi desenvolvido, é demonstrado como ele opera.

---

```
1 //...
2 const cloud = new KNoTCloud(
3     '10.77.35.43',
4     3000,
5     'ac424a73-3508-4ca7-a067-8ea92e510000',
6     'f32f559942c54ef95e04c8236ddc0f9adccad6bb',
7 );
8
9 admin.initializeApp({
10     credential: admin.credential.cert(serviceAccount),
11     databaseURL: "https://graincontrol-8b600.firebaseio.com"
12 });
```

---

No trecho acima, na linha 2 é instanciado o objeto "*cloud*", definindo os parâmetros de *host*, *port* e credenciais do usuário (UUID e *Token* do construtor da classe *KNoTCloud*) para seja possível manipular os dados providos pelo *KNoTCloud*. Na linha 9 é inicializado o serviço da plataforma *Firebase*, onde é fornecido o arquivo do tipo JSON da conta de serviço da plataforma (linha 10) e URL do serviço (linha 11). Vale ressaltar que esses procedimentos descritos acima, são realizados em ambos os algoritmos.

---

```
1 while (true) {
2     let array = await cloud.getData(ID);
3     //...
4     if (arrayTempAvg != []) {
5         for (let i = 0; i < arrayTempAvg.length; i++) {
6             const tempValue = arrayTempAvg[i];
7
8             if (i == 0 && tempValue >= 1){
9                 admin.database().ref('/temperaturaSensorAvg/' +
10                     i).set(tempValue);
11                 arrayGoodValues.push(tempValue);
12             } else if ((tempValue != arrayTempAvg[i - 1]) && tempValue >= 1) {
```

```

12         admin.database().ref('/temperaturaSensorAvg/' +
13             i).set(tempValue);
14
15         arrayGoodValues.push(tempValue);
16     }
17     lastAvgValue = arrayGoodValues[arrayGoodValues.length - 1];
18     admin.database().ref('/average/').set(lastAvgValue);
19 }
20 }
21 //...

```

---

Continuando a explanação, como pode se observar acima, basicamente o fluxo do código consiste num laço de repetição em *loop* infinito que tem como premissa ler e armazenar os dados de temperatura média fornecidos pela *cloud* em um vetor com uma quantidade de dez amostras de temperatura média, tal quantidade é padrão, podendo ser alterada pelo informe do limite de amostras a serem lidas, através do método provido pela biblioteca KNoT *Cloud* "*getData()*". Esse vetor em questão será manipulado para que os dados contidos nele sejam enviados para o *Real Time Database*, sendo dispostos, em formato JSON, nessa estrutura no banco de dados NoSQL:

```

1 {
2   "average" : 23.8,
3   "setpoint" : {
4     "valor" : 20
5   },
6   "temperaturaSensorAvg" : [ 23.8, 23.8, 23.8, 23.8, 23.8, 23.8, 23.8, 23.8,
7     23.8, 23.8 ]

```

---

Enquanto ao outro algoritmo, demonstrado logo abaixo, que é encarregado de monitorar mudança de valor de SP, por meio do método "*on('value', function (snapshot){})*" (linha 16), fornecido pela biblioteca do *Firebase*, para que em caso de alteração desse valor, este seja enviado para *Cloud*, por meio do método provido pela biblioteca *KNoT Cloud Library* que por sua vez envie esse valor recebido para a *thing*, para que ela possa realizar o controle da temperatura interna do silo. Os testes apontaram um tempo médio de dois segundos para a transmissão do valor do *setpoint* para a *thing*.

```

1 //...
2 async function main(val) {

```

```

3     try {
4         await cloud.connect();
5         const devices = await cloud.getDevices();
6         console.log(devices);
7
8         await cloud.setData(ID, [{ sensorId: 6, value: val }]);
9
10    } catch (err) {
11        console.error(err);
12    }
13    await cloud.close();
14 }
15 const ref = admin.database().ref('setpoint/valor')
16 ref.on("value", function (snapshot) {
17
18     console.log(snapshot.val())
19     main(snapshot.val());
20
21 }, function (error) {
22     console.log("Error: " + error.code);
23 });

```

---

Todos esses processos são possíveis graças ao auxílio imprescindível das bibliotecas citadas: a *KNoT Cloud Library*, que possibilita a leitura e escrita de dados fornecidos pela *cloud* e a biblioteca do serviço *RealTime Database*, que provê métodos otimizados que realizam o processo de envio da temperatura média para o banco de dados e monitoramento do valor do SP remoto.

#### 5.4 DESENVOLVIMENTO DO APLICATIVO *MOBILE GRAIN CONTROL*

O desenvolvimento do aplicativo *mobile* (Figura 21), se consistiu na construção de uma interface homem-máquina (IHM) para que o usuário, com poucos toques na tela, possa realizar as ações de alteração do SP, visualizar a média da temperatura interna do silo e também visualizar o gráfico que é composto pelas dez amostras de temperatura média, que são obtidas ao longo do tempo. Para a realização dessas ações foi adicionado a dependência do *Firebase-database*, via gerenciador de dependências *Gradle* (GRADLE, 2018), que proporciona o acesso e consumo aos dados localizados no banco de dados. No contexto de IoT, aplicações *mobile* são modelos de IHM funcionais e recomendados.

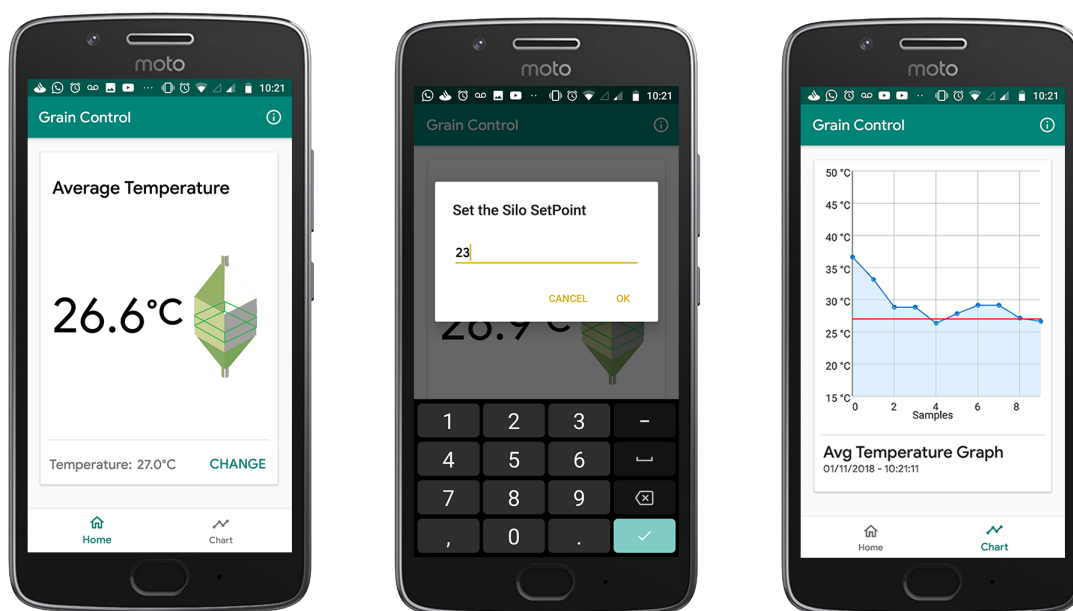


Figura 21 – Telas do aplicativo *Grain Control*

Fonte: Do autor

## 5.5 VALIDAÇÃO DO SISTEMA

O processo de validação do sistema, foi composto por testes, como o objetivo de avaliar o envio de SP remoto para a bancada PP32 por meio do aplicativo, para que houvesse o controle da temperatura média interna do silo. Esses testes tiveram como objetivo comum a captura dos valores de temperatura média interna do silo e com isso ser possível gerar gráficos com o propósito de análise do controle de temperatura e comparações de resultados. A primeira etapa serviu para coletar 7.324 dados da temperatura ao longo de um período de tempo, aproximadamente, de dois minutos, com o uso do *sketch* que utiliza o controle de processo liga-desliga com histerese, sendo testado em três níveis de histerese (0.1, 0.5 e 1.0) e potência do ventilador do silo (25%, 50% e 75%). Já na segunda etapa foi utilizado o *sketch* que trabalha com o controle de processo PID. Nele foram coletados 22.824 dados ao longo de aproximadamente oito minutos, dados da temperatura em três potências do ventilador (25%, 50% e 75%). Em ambas as etapas a troca do valor de referência foi dada a partir do informe via aplicativo *mobile*, com um tempo de envio de aproximadamente três segundos até chegar a *thing*.

O motivo pelo qual foram definidos tempos diferentes de coleta de dados, se deu pelo fato de que o *sketch* que usa a técnica de liga-desliga, permite que a análise dos dados de oscilação de temperatura fosse feita em menos tempo devido a menor complexidade do algoritmo, enquanto que o inverso ocorre com o *sketch* que utiliza PID, demandando mais tempo para que se possa analisar e tirar conclusões.

Após então a realização dessas duas etapas foram obtidos como parâmetros de desempenho os resultados de somatório da integral do erro quadrático (ISE) ( $\sum_{k=1}^N e^2(k)$ ) e o total de energia ( $\sum_{k=1}^N u(k)$ ) gasto pela geração do sinal de controle em ambos os *sketches*, que serão explanados no próximo capítulo.

## 6 RESULTADOS E DISCUSSÃO

### 6.1 RESULTADOS COMPUTACIONAIS

Os resultados computacionais foram obtidos com o auxílio da biblioteca *PySerial* (ver seção 4.1.7) e a geração de gráficos através da IDE *GNU Octave*, para proporcionar uma melhor análise comparativa entre os resultados do controle de processo tanto de liga-desliga como do PID.

#### 6.1.1 Gráficos controle liga-desliga

A seguir são apresentados os gráficos dos resultados relativos ao controle de processo liga-desliga, cado um com seu nível de histerese (0.1, 0.5 e 1.0), com as médias de temperatura ao longo do tempo levado em cada teste e potência do ventilador do silo (25%, 50% e 75%).

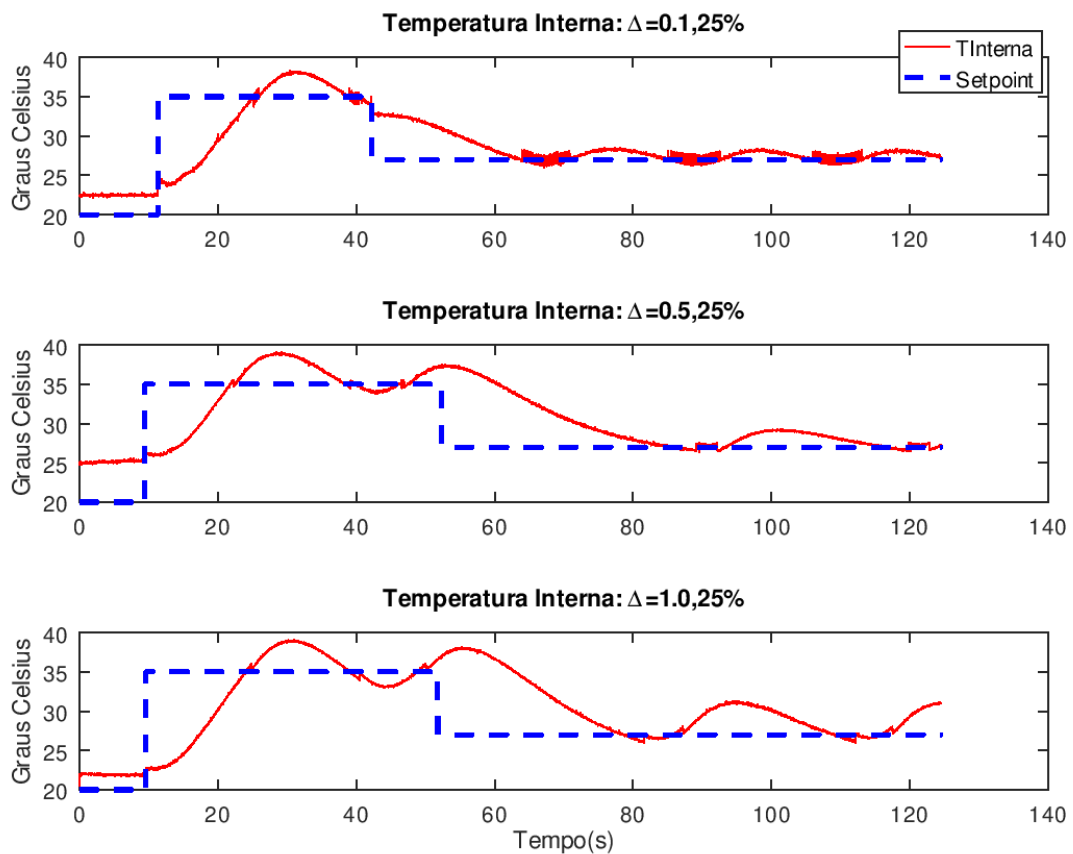


Figura 22 – Temperatura média da massa de grãos usando controle liga-desliga para potência de 25% e  $\Delta = 0.1, 0.5, 1.0$

Fonte: Do autor

O gráfico acima (Figura 22) foi gerado a partir da potência do ventilador de 25%,

que tomando como análise os dados apresentados, tem se uma menor eficácia para a estabilização do valor de temperatura média em relação ao valor de referência (*setpoint*), devido a baixa potência aplicada pelo ventilador.

No gráfico de sinal de controle a seguir (Figura 23) foi gerado também a partir da potência do ventilador de 25%, que tomando como análise os dados apresentados, tem se um chaveamento maior no sinal de controle, onde a histerese é menor (0.1), gerando um efeito denominado na literatura como "balde de tinta".

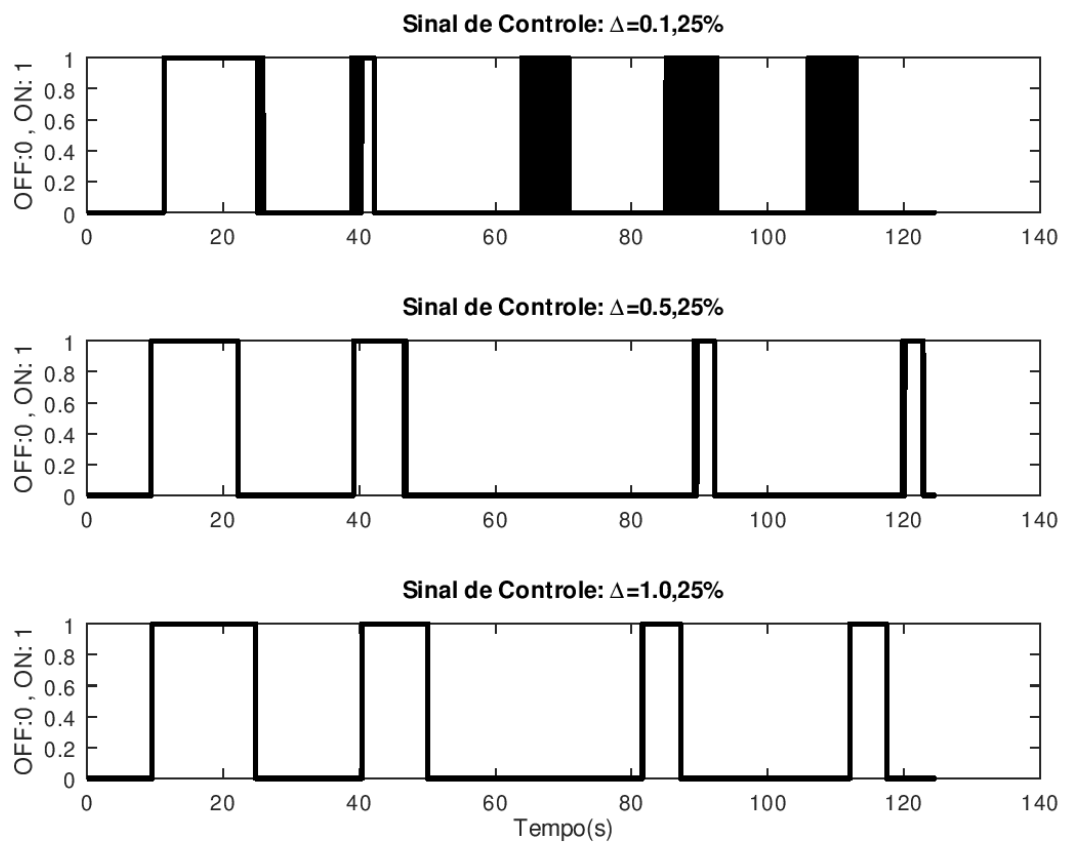


Figura 23 – Sinal de controle liga-desliga para potência de 25% e  $\Delta = 0.1, 0.5, 1.0$

Fonte: Do autor

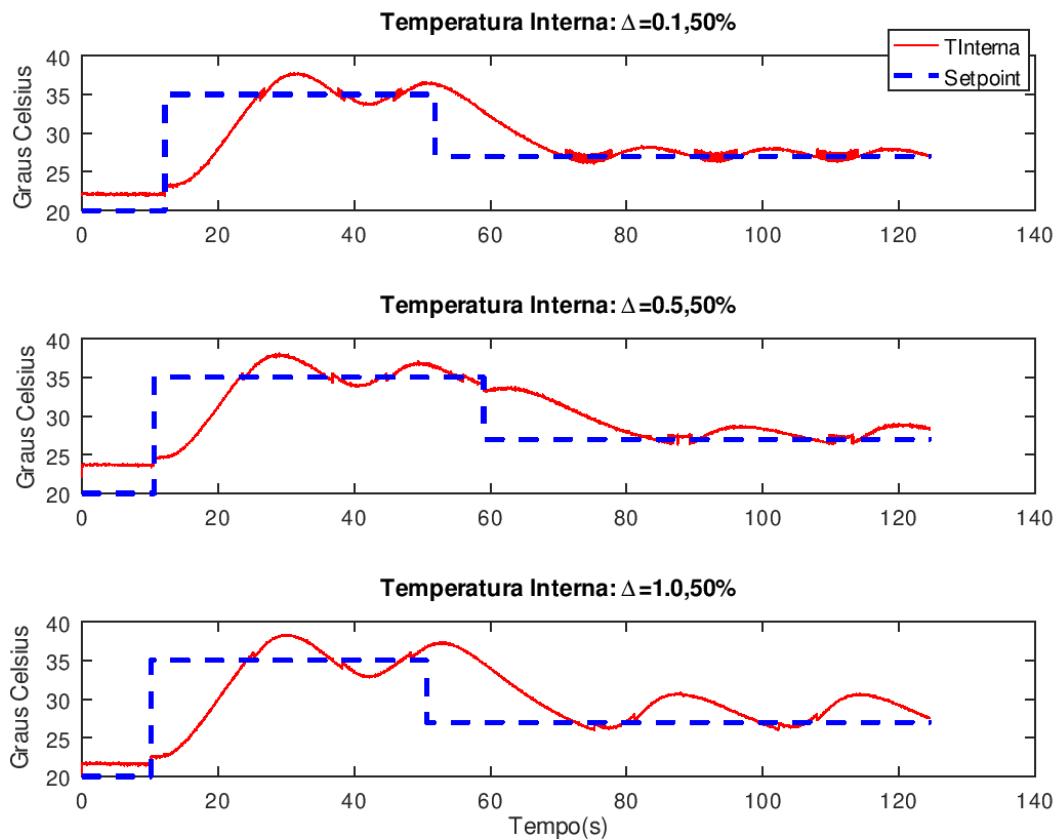


Figura 24 – Temperatura média da massa de grãos usando controle liga-desliga para potência de 50% e  $\Delta = 0.1, 0.5, 1.0$

Fonte: Do autor

O gráfico acima (Figura 24) foi gerado a partir da potência do ventilador de 50%, que tomando como análise os dados apresentados, tem se uma maior eficácia para a estabilização do valor de temperatura média em relação ao valor de referência (*setpoint*) em comparação com o gráfico anterior (Figura 22), devido ao aumento de 25% da potência aplicada pelo ventilador.



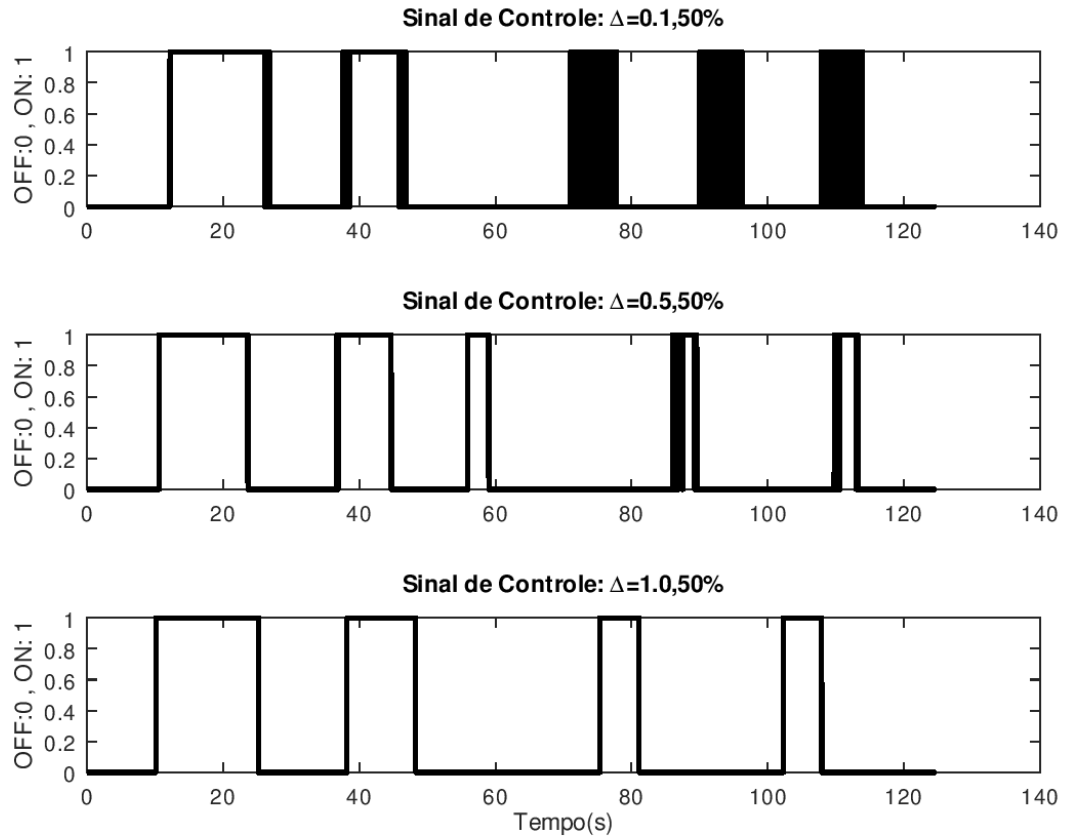


Figura 25 – Sinal de controle liga-desliga para potência de 50% e  $\Delta = 0.1, 0.5, 1.0$

Fonte: Do autor

No gráfico de sinal de controle acima (Figura 25) foi gerado também a partir da potência do ventilador de 50%, que tomando como análise os dados apresentados, tem se um chaveamento maior no sinal de controle, onde a histerese é menor (0.1) em comparação com o gráfico de sinal de controle anterior (Figura 23), devido a como se foi mencionado, ao aumento da potência da ventilador do silo.

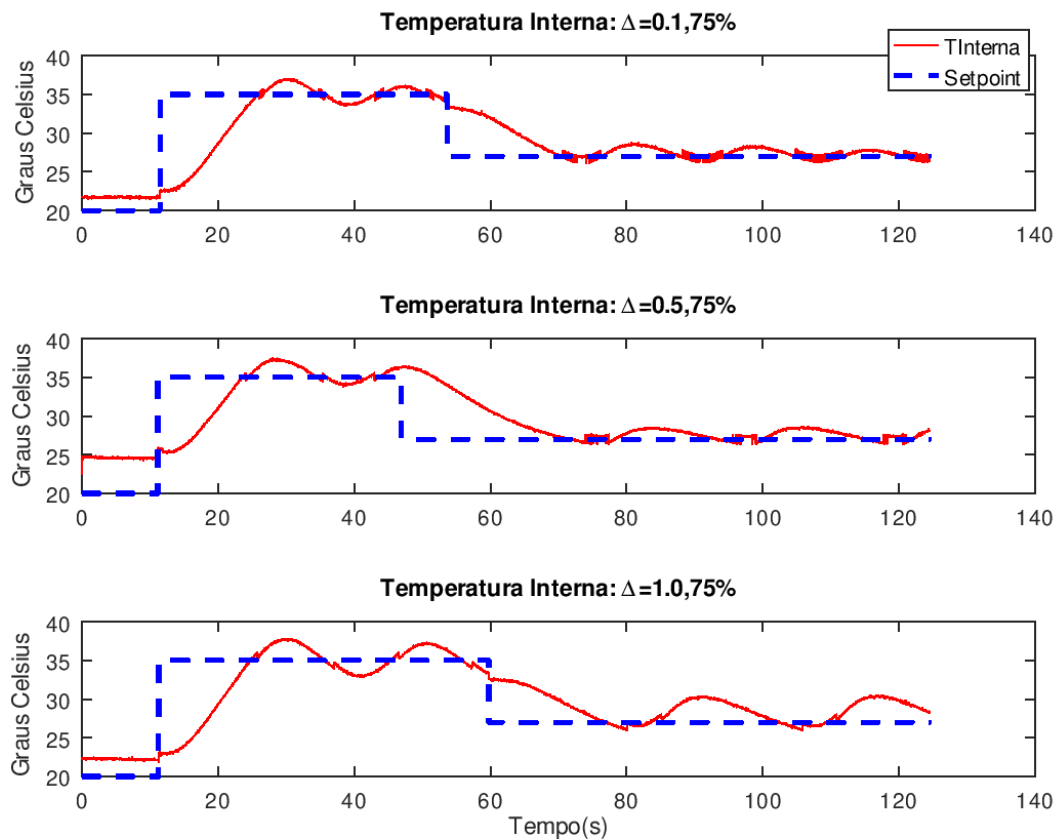


Figura 26 – Temperatura média da massa de grãos usando controle liga-desliga para potência de 75% e  $\Delta = 0.1, 0.5, 1.0$

Fonte: Do autor

O gráfico acima (Figura 26) foi gerado a partir da potência do ventilador de 75%, que tomando como análise os dados apresentados, tem se uma maior eficácia para a estabilização do valor de temperatura média em relação ao valor de referência (*setpoint*) nos três níveis de histerese que são trabalhados em comparação com os dois gráficos anteriores, que envolvem essa relação (Figura 22 e 24), devido ao aumento de 50% da potência aplicada pelo ventilador.

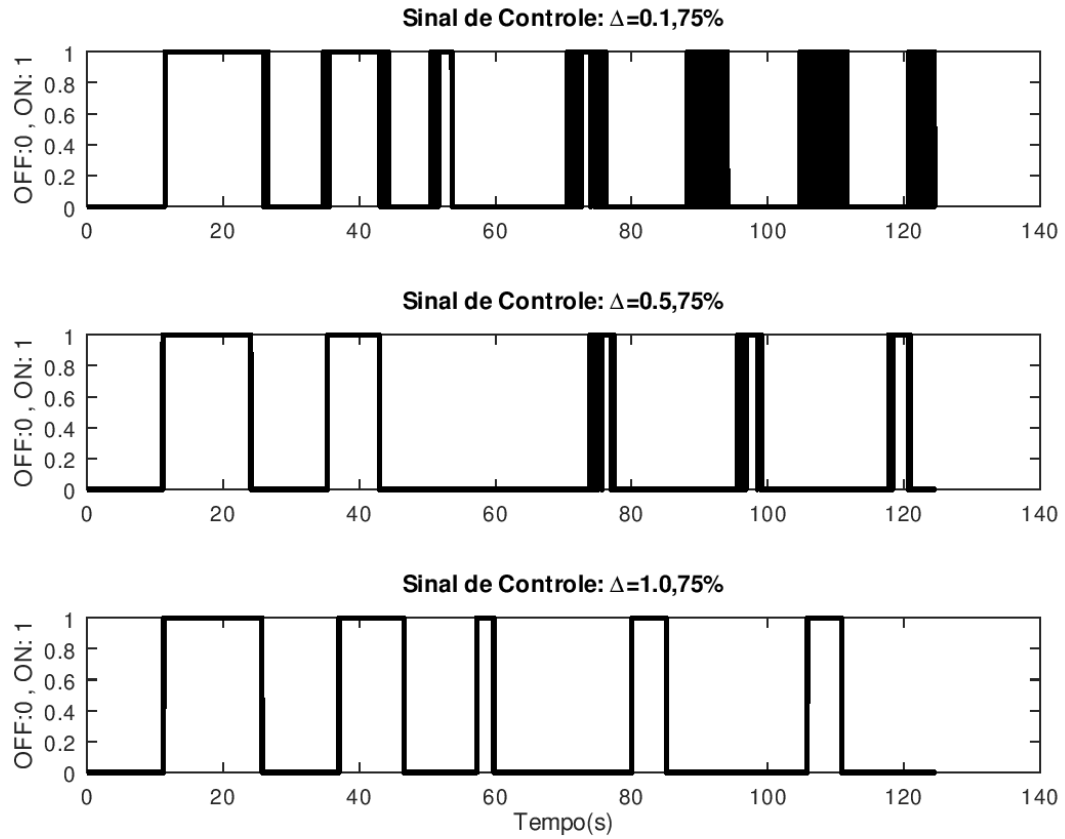


Figura 27 – Sinal de controle liga-desliga para potência de 75% e  $\Delta = 0.1, 0.5, 1.0$

Fonte: Do autor

No gráfico de sinal de controle acima (Figura 27) foi gerado também a partir da potência do ventilador de 75%, que tomando como análise os dados apresentados, tem-se um chaveamento bem maior no sinal de controle, onde a histerese é menor (0.1) em comparação a demais gráficos de sinal de controle anteriores, em detrimento do aumento da potência do ventilador do silo.

### 6.1.2 Gráficos controle PID

A seguir são apresentados os gráficos dos resultados relativos ao controle de processo PID, cada um com seu nível de potência do ventilador do silo.

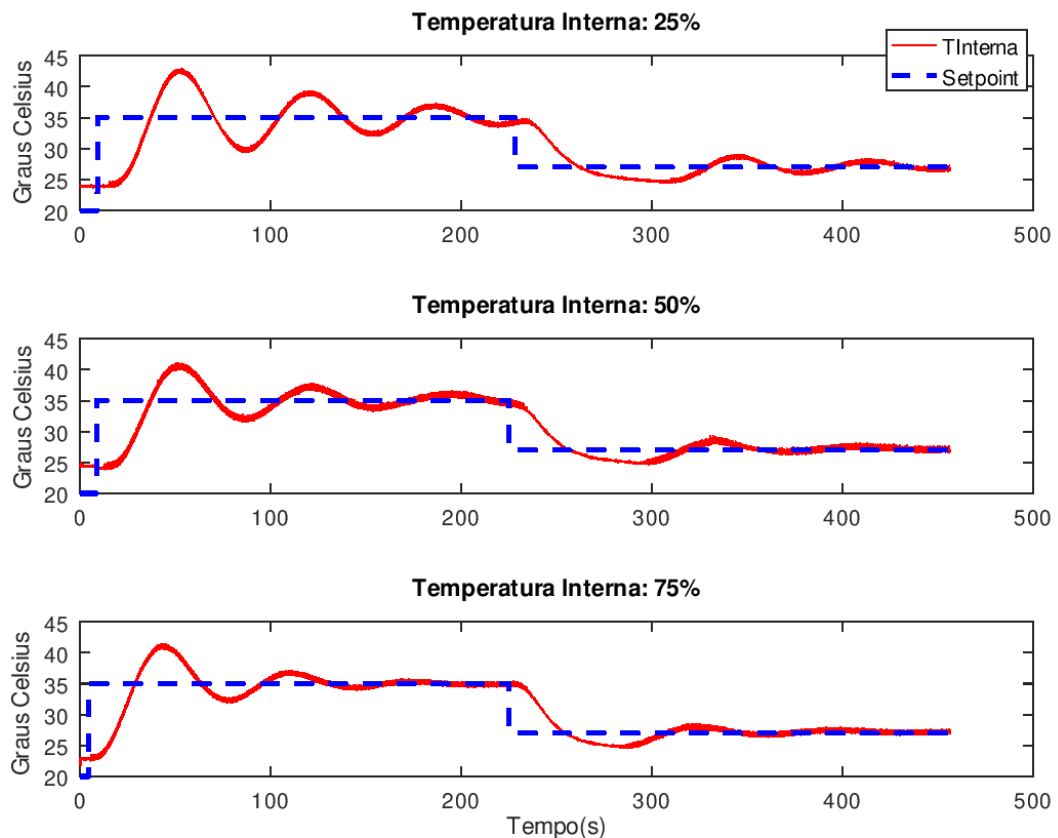


Figura 28 – Temperatura média da massa de grãos usando controle PID para potências de 25%, 50% e 75%

Fonte: Do autor

O gráfico acima (Figura 28) foi gerado a partir de três potências para o ventilador, que tomando como análise os dados apresentados, tem se uma maior eficácia para a estabilização do valor de temperatura média em relação ao valor de referência (*setpoint*) para o maior valor de potência (75%), em contra partida há um maior gasto de energia, em prol dessa eficácia.

No gráfico de sinal de controle abaixo (Figura 29) foi gerado também a partir de três potências para o ventilador, que tomando como análise os dados apresentados, tem se um maior fluxo de energia emitido pelo relé SSR ao sinal de controle, quando se há maior valor de potência configurado.

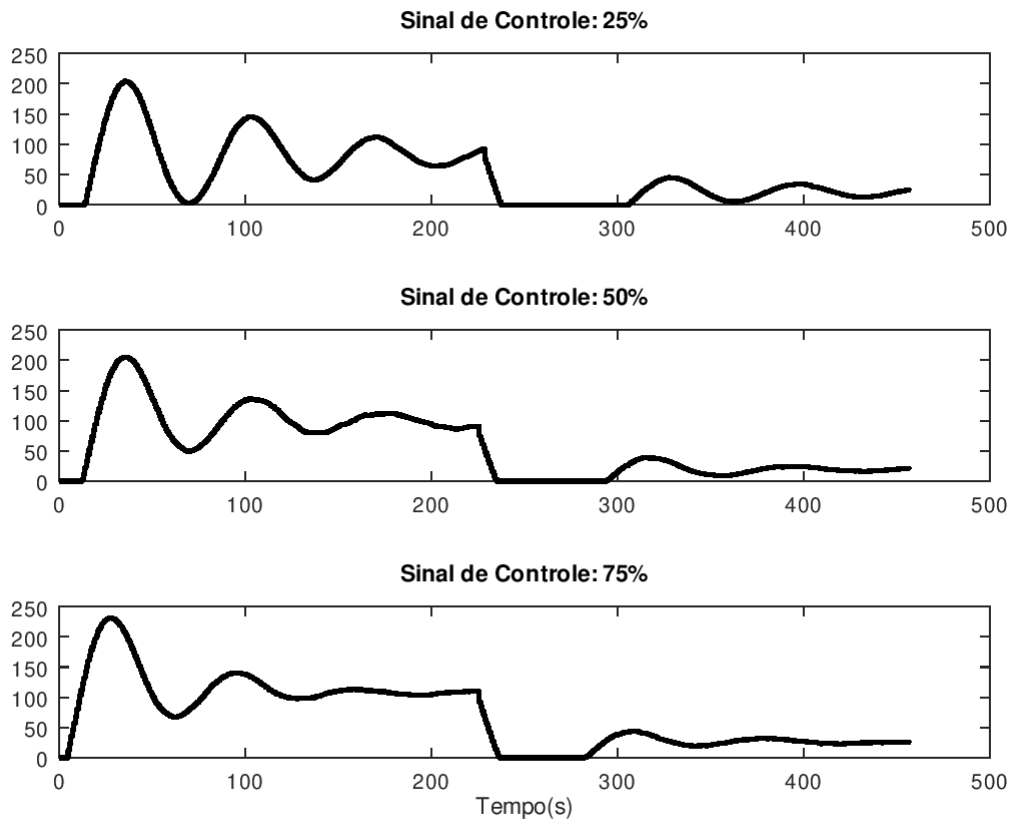


Figura 29 – Sinal de controle PID para potências de 25%, 50% e 75%

Fonte: Do autor

## 6.2 ANÁLISE DOS RESULTADOS

### 6.2.1 Resultados do controle de processo liga-desliga

Foram realizados nove ensaios para que fosse possível a obtenção de resultados em prol da análise e comparações dos dados que representa, o desempenho da técnica de controle *on-off*. Como pode ser observado na Tabela 4, na região de maior histerese (1.0), há um elevado consumo total de energia em todas as potências definidas do soprador/ventilador, gerado pelo envio de sinal do Arduino para o Relé SSR *MetalLex* TSZC. Em contrapartida, nas demais regiões, quase todos os totais de energias são inferiores em relação a região de maior histerese, além de possuírem ISE menor, em virtude de suas regiões de histerese serem menores. Outra observação que pode ser feita é que na região de menor histerese (0.1) onde a potência do soprador é maior se tem o maior consumo total de energia dentre os demais consumos, devido ao forte chaveamento do sinal de controle emitido pelo relé.

Tabela 4 – Resultados de eficiência *on-off*

Região de Histerese $\Delta$	Potência do Soprador	$\sum_{k=1}^N e^2(k)$ *	$\sum_{k=1}^N u(k)$
0.1	25%	$7.82 \times 10^4$	1563
	50%	$9.59 \times 10^4$	1873
	75%	$8.36 \times 10^4$	<b>2260</b>
0.5	25%	$1.24 \times 10^5$	1549
	50%	$8.47 \times 10^4$	1760
	75%	$9.47 \times 10^4$	1673
1.0	25%	$1.72 \times 10^5$	2106
	50%	$1.41 \times 10^5$	2153
	75%	$8.91 \times 10^5$	2151

\*Foi adotado o uso de duas casas decimais para facilitar compreensão dos dados relativos a integrativa do erro ao quadrado.

### 6.2.2 Resultados controle de processo PID

Tabela 5 – Resultados de eficiência PID

Potencia do soprador	$\sum_{k=1}^N e^2(k)$	$\sum_{k=1}^N u(k)$
25%	$2.65 \times 10^5$	$1.13 \times 10^6$
50%	$1.94 \times 10^5$	$1.31 \times 10^6$
75%	$1.86 \times 10^5$	$1.56 \times 10^6$

No caso dos resultados apresentados na Tabela 5 acima, foram realizados somente três ensaios para se obter o conjunto de dados que permitissem a análise e comparação, através da definição dos parâmetros de ganho proporcional (KP), integrativo (KI) e derivativo (KD) no algoritmo de controle PID para Arduino, adquiridos via controlador NOVUS, equivalentes a:  $KP = 1/0.6$ ,  $KI = KP/1.77$ ,  $KD = KP.6$ . Assim sendo pode-se realizar a análise e notar uma inversão de proporcionalidade, em razão da maior quantidade de total de energia gasta pelo envio do sinal de controle ao relé, levando em conta o aumento da potência do soprador, para se obter o menor ISE. O fato do sinal de controle do PID quase não ser desligado, mas mantido em um pouco período de tempo em valor mínimo (zero) (Figura 29), produz um consumo maior de energia, mas aumenta a vida útil dos atuadores, por não haver chaveamentos bruscos. O controle é mais suave e mantém a variável controlada mais estável sobre o SP. O objetivo destes testes não foi a comparação direta das técnicas, mas podê visualizar o *trade-off* (seu significado representa escolha) analisando o custo-benefício para o uso dessas técnicas de controle, mais comuns da indústria, utilizadas em conjunto com a plataforma KNoT.

Os parâmetros utilizados não estão otimizados e, portanto, podem ser melhorados por meio de metodologias diversas disponíveis na literatura, determinísticas como ganho crítico, Ziegler Nichols (BAZANELLA; SILVA JR., 2005) ou estocásticas, como a Otimização por Enxame de Partículas (PSO) (OLIVEIRA *et al.*, 2014).

## 7 CONCLUSÃO E TRABALHOS FUTUROS

Nesta monografia foi abordado o desenvolvimento do sistema *Grain Control* que está inserido no contexto de IoT, onde seu objetivo de monitoramento e controle da temperatura média, por meio da plataforma desenvolvida pelo CESAR, KNoT foi alcançado aliado a técnicas de controle de processos liga-desliga e PID, amplamente utilizados na indústria de automação.

No entanto, ainda há um campo aberto enorme para melhorias e trabalhos futuros. Tais melhorias se consistem na adição de uma novas funcionalidades, como a possibilidade do usuário do aplicativo *Grain Control* poder selecionar qual processo de controle ele deseja utilizar, seja ele, liga-desliga ou PID. Como também o aprimoramento do aplicativo para que ele se torne algo mais robusto, permitindo o cadastro de usuários, para que cada usuário/produtor possa cadastrar os locais em que os grãos ou qualquer matéria-prima estejam e com isso poder monitorar e controlar variáveis, não somente de temperatura, mas, de umidade, como também poder realizar previsões sobre qual seria a melhor temperatura ou umidade para determinada temporada, seja ela de pré ou pós-colheita.

Além dessas funcionalidades citadas acima, pode-se incluir o desenvolvimento focado na utilização de técnicas de otimização para determinar e ajustar de maneira autônoma quais os melhores parâmetros do PID, para determinado SP informado pelo usuário, nesse tipo de controle. Estudos sobre o vigor dos grãos armazenados a partir da leitura de alguns parâmetros físicos, explorando os conceitos de modelos incorporados a sensores inteligentes (*smart sensors*) no contexto de IoT. Segue-se ainda, contribuir para o CESAR com a inclusão do ESP8266/ESP32, como uma segunda opção ao Arduino UNO, para atuar também como *thing* e o protocolo MQTT.



## REFERÊNCIAS

- ADD-THERM. **Termistores NTC**. ADD-THERM, 2018. Disponível em: <<http://www.addtherm.com.br/produto/termistores-ntc/>>. Acesso em: 10 nov. 2018.
- ANDROID. **Conheça o Android Studio**. Android Developers, 2018. Disponível em: <<https://developer.android.com/studio/intro/?hl=pt-br>>. Acesso em: 10 nov. 2018.
- ARDUINO. **Arduino, About Us**. ARDUINO, 2018. Disponível em: <<https://www.arduino.cc/en/Main/AboutUs>>. Acesso em: 10 nov. 2018.
- AVRAM, Abel. **Google Firebase: back-end completo para aplicações web e mobile**. InfoQ Brasil, 2018. Disponível em: <<https://www.infoq.com/br/news/2016/07/google-firebase>>. Acesso em: 10 nov. 2018.
- BAYER, Fernando Mariano; ARAÚJO, Olinto César Bassi de. **Controle Automático de Processos**. Rio Grande do Sul: Colégio Técnico Industrial de Santa Maria - UFSM, 2011.
- BAZANELLA, Alexandre Sanfelice; SILVA JR., João Manoel Gomes da. **Sistemas de Controle: princípios e métodos de projeto**. Rio Grande do Sul: UFRGS Editora, 2005.
- BRASIL. Lei nº 9.973, 29 maio. 2000.
- CARNEIRO, Orlando. Silos e sua construção. p. 4, 1948.
- CILO, Hugo. **O desafio da armazenagem**. Revista Dinheiro Rural, 2016. Disponível em: <<https://www.dinheirorural.com.br/secao/agroeconomia/o-desafio-da-armazenagem>>. Acesso em: 02 nov. 2018.
- DAVIS, ROD; STAFFORD, ROSS. Grain storage and handling. p. 6, 2016.
- DORNELAS, Everton; OLIVEIRA, Sérgio Campello. Monitoramento de consumo doméstico de Água utilizando uma meta-plataforma de iot. **Revista de Engenharia e Pesquisa Aplicada**, v. 2, n. 2, 2017.
- FARONI, Lêda Rita D'Antonino; SOUSA E SILVA, Juarez de. Manejo de pragas no ecossistema de grãos armazenados. In: \_\_\_\_\_. **Secagem e armazenagem de produtos agrícola**. Viçosa – MG: Editora Aprenda Fácil, 2009. cap. 15.
- GORAN, Julie; LABERGE, Laura; SRINIVASAN, Ramesh. Culture for a digital age. 2017. Disponível em: <<https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/culture-for-a-digital-age>>. Acesso em: 05 nov. 2018.
- GRADLE. **Gradle Build Tool**. Gradle Build Tool, 2018. Disponível em: <<https://gradle.org/>>. Acesso em: 19 nov. 2018.
- GUERRA, Wladimir de Andrade. Implementação de controle proporcional, integral e derivativo digital em controladores lógico programáveis. 2009.
- HARA, Tetuo. **Aeração com qualidade**. REVISTA CULTIVAR, 2003. Disponível em: <<https://www.grupocultivar.com.br/artigos/aeracao-com-qualidade>>. Acesso em: 13 nov. 2018.

JAVA. **Go Java**. ORACLE, 2018. Disponível em: <<https://go.java/index.html?intcmp=gojava-banner-java-com>>. Acesso em: 19 nov. 2018.

KARIMI, Kaivan. **The Role of Sensor Fusion in the Internet of Things**. MOUSER ELECTRONICS, 2018. Disponível em: <<https://br.mouser.com/applications/sensor-fusion-iot/>>. Acesso em: 4 dez. 2018.

KNOT. **KNoT Network of things**. CESAR, 2015. Disponível em: <<https://www.knot-cesar.org.br/>>. Acesso em: 07 nov. 2018.

MASSRUHÁ, Silvia Maria Fonseca Silveira; LEITE, Maria Angelica de Andrade. Agro 4.0 - rumo à agricultura digital. 2016.

MESHLBLUE. **Welcome to Meshblu**. Meshblue, 2017. Disponível em: <<https://meshblu.readme.io/>>. Acesso em: 4 nov. 2018.

MONGODB. **MongoDB, Inc**. MongoDB, 2018. Disponível em: <<https://goo.gl/PnTynB>>. Acesso em: 07 nov. 2018.

MYSENSE. **mySense ioT++**. Uma plataforma para Agricultura de Precisão. mySense ioT++, 2018. Disponível em: <<https://mysense.utad.pt/>>. Acesso em: 16 nov. 2018.

NORDIC. **Single chip 2.4 GHz Transceiver nRF24L01**. Nordic Semiconductor, 2006. Disponível em: <[https://www.sparkfun.com/datasheets/Components/nRF24L01\\_prelim\\_prod\\_spec\\_1\\_2.pdf](https://www.sparkfun.com/datasheets/Components/nRF24L01_prelim_prod_spec_1_2.pdf)>. Acesso em: 10 nov. 2018.

NOVUS. Controle pid básico. Novus Produtos Eletrônicos Ltda - Todos os direitos reservados, 2003.

OCTAVE, GNU. **About**. GNU Octave, 2018. Disponível em: <<https://www.gnu.org/software/octave/about.html>>. Acesso em: 10 nov. 2018.

OLIVEIRA, J.B.; BOAVENTURA-CUNHA, J.; OLIVEIRA, P.B. Moura; FREIRE, H. A swarm intelligence-based tuning method for the sliding mode generalized predictive control. **ISA Transactions**, v. 53, n. 5, p. 1501 – 1515, 2014. ISSN 0019-0578. ICCA 2013. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S001905781400127X>>.

OLIVEIRA, Marcelo Alvares de. **Evolução da armazenagem de grãos no Brasil**. BLOG CANAL RURAL, 2017. Disponível em: <<https://blogs.canalrural.uol.com.br/embrapasoja/2017/02/14/evolucao-da-armazenagem-de-graos-no-brasil/>>. Acesso em: 12 nov. 2018.

PEREIRA JR., Modesto. Aplicação de controlador fuzzy supervisor em um secador na indústria de fertilizantes. 2018.

PYSERIAL. **PySerial**. PySerial, 2015. Disponível em: <<https://pythonhosted.org/pyserial/pyserial.html>>. Acesso em: 10 nov. 2018.

RASPBERRY. **Raspberry Pi, About Us**. RASPBERRYPI, 2018. Disponível em: <<https://www.raspberrypi.org/about/>>. Acesso em: 10 nov. 2018.

RURAL, REVISTA GLOBO. **Brasil perdeu espaço no mercado agrícola mundial, afirma OMC**. 2017. Disponível em: <<https://revistagloborural.globo.com/Noticias/Economia/noticia/2017/07/globo-rural-brasil-perdeu-espaco-no-mercado-agricola-mundial-afirma-omc.html>>. Acesso em: 02 nov. 2018.

SANCHES, Walter. Indústria 4.0 e IoT: uma transformação muito mais do que digital. 2018. Disponível em: <<http://cio.com.br/opinioao/2018/03/28/industria-4-0-e-iot-uma-transformacao-muito-mais-do-que-digital/>>. Acesso em: 02 nov. 2018.

SENAI, INSTITUTO. **KNoT: a meta plataforma de código aberto para IoT**. INSTITUTO SENAI DE INOVAÇÃO PARA TECNOLOGIA DA INFORMAÇÃO, 2018. Disponível em: <<https://isitics.trixapp.net/tic/knot-a-meta-plataforma-de-codigo-aberto-para-iot/>>. Acesso em: 15 nov. 2018.

SILVA, Juarez de Sousa e; LACERDA FILHO, Adílio Flauzino; DEVILLA, Ivano Alessandro. Secagem e armazenagem de produtos agrícolas. p. 262, 2008.

SILVA JR. *et al.* Eficiência energética em sistemas de aeração de silos de armazenagem de grãos, utilizando inversores de frequência. Universidade Estadual Paulista (UNESP), 2004.

SIVIERO, Luiz Fernando Conte. Monitoramento via ultrassom, temperatura e umidade no armazenamento de grãos com acionamento do sistema de aeração. 2017.

SOUZA, Danilo Alfredo Marinho de. Estudo de lacunas da meta-plataforma knot para iot. 2017.

SOUZA, Fábio. **Usando as saídas PWM do Arduino**. Embarcados, 2014. Disponível em: <<https://www.embarcados.com.br/pwm-do-arduino/>>. Acesso em: 16 nov. 2018.

T&S. **MANUAL DE INSTRUÇÕES, PLANTA DE CONTROLE TÉRMICO, MODELO: TSPT05**. T&S Equipamentos Eletrônicos, 2005. Disponível em: <<http://tesequipamentos.com.br/arq/MA-TS-1029.PDF>>. Acesso em: 16 nov. 2018.

XIAODONG, Zhang *et al.* Environmental monitoring in grain granary based on embedded system. In: **2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)**. [S.l.: s.n.], 2016. p. 1051–1054. ISSN 2327-0594.

ZHAO, Y. Monitor and control system design of grain depot temperature based on wireless communication technology. In: **2009 First International Workshop on Education Technology and Computer Science**. [S.l.: s.n.], 2009. v. 1, p. 983–986.