

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
ENG. DE CONTROLE E AUTOMAÇÃO

MARCELO DE OLIVEIRA MONARETTO -  
00273183

**SUPERVISÓRIO INTEGRADO COM  
IOT PARA IDENTIFICAÇÃO E  
NOTIFICAÇÃO DE MUDANÇAS  
NA PLANTA DE COMPOSTAGEM  
DOMÉSTICA CONTROLADA**

Porto Alegre  
2023

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
ENG. DE CONTROLE E AUTOMAÇÃO

MARCELO DE OLIVEIRA MONARETTO -  
00273183

**SUPERVISÓRIO INTEGRADO COM  
IOT PARA IDENTIFICAÇÃO E  
NOTIFICAÇÃO DE MUDANÇAS  
NA PLANTA DE COMPOSTAGEM  
DOMÉSTICA CONTROLADA**

Trabalho de Conclusão de Curso (TCC-CCA)  
apresentado à COMGRAD-CCA da Universi-  
dade Federal do Rio Grande do Sul como parte  
dos requisitos para a obtenção do título de *Ba-  
charel em Eng. de Controle e Automação* .

ORIENTADOR(A):

Prof. Dr. Marcelo Götz

Porto Alegre  
2023

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
ENG. DE CONTROLE E AUTOMAÇÃO

MARCELO DE OLIVEIRA MONARETTO -  
00273183

**SUPERVISÓRIO INTEGRADO COM  
IOT PARA IDENTIFICAÇÃO E  
NOTIFICAÇÃO DE MUDANÇAS  
NA PLANTA DE COMPOSTAGEM  
DOMÉSTICA CONTROLADA**

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção dos créditos da Disciplina de TCC do curso *Eng. de Controle e Automação* e aprovado em sua forma final pelo(a) Orientador(a) e pela Banca Examinadora.

Orientador(a): \_\_\_\_\_

Prof. Dr. Marcelo Götz, UFRGS

Doutor pela Universität Paderborn – Paderborn, Alemanha

Banca Examinadora:

Prof. Dr. Marcelo Götz, UFRGS

Doutor pela Universität Paderborn – Paderborn, Alemanha

Profa. Dra. Lucíola Campestrini, UFRGS

Doutora pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Renato Ventura Bayan Henriques, UFRGS

Doutor pela Universidade Federal de Minas Gerais – Belo Horizonte, Brasil

\_\_\_\_\_  
Mário R. Sobczyk Sobrinho

Coordenador de Curso

Eng. de Controle e Automação

Porto Alegre, Abril 2023

## **DEDICATÓRIA**

Dedico este trabalho aos meus pais, Leandro e Valéria, e minha irmã, Marta, em especial pelo apoio e paciência que mostraram em todos os momentos difíceis, os quais foram essenciais para a finalização deste trabalho. Dedico também este trabalho para a Diana, que me apoiou e motivou para seguir minhas ideias, mesmo quando elas pareciam ser impossíveis de serem concretizadas.



## **AGRADECIMENTOS**

À Universidade Federal do Rio Grande do Sul, UFRGS, pela oportunidade de realização de meus estudos.

Aos colegas de curso pelo seu auxílio nas tarefas desenvolvidas durante o curso, possibilitando um crescimento em conjunto.

Agradeço ao meu orientador, Marcelo Götz, cuja revisão, empatia e grandes ideias foram essenciais para a confecção deste trabalho, do início ao fim.

## RESUMO

O presente trabalho tem como objetivo fornecer um sistema capaz de identificar e notificar um usuário sobre mudanças em uma planta controlada, usando uma abordagem de arquitetura *IoT*. Para tanto, juntam-se as áreas de *internet*; sensoriamento; controle, sem parâmetros de controle fixos; interfaces gráficas; banco de dados; *APIs* na nuvem. A solução é capaz de projetar e usar um controle PID de temperatura, implementado com uma placa Arduino de um sistema de compostagem doméstica simplificado, enviando dados de temperatura e tensão de saída enquanto recebe os ganhos do controlador, atualizados a cada ciclo de amostragem. A interface gráfica apresenta para o usuário, de forma compreensível e intuitiva, todos os dados de temperatura e tensão de saída, além de fornecer a capacidade de alterar os parâmetros de controle. A interface também é responsável por notificar o usuário sobre mudanças significativas na planta ao longo do tempo, caso existam. A *API* na nuvem serve como intermediadora entre os demais módulos do projeto e o banco de dados, também na nuvem, manejando as chamadas e requisições HTTP. Os resultados indicam que a solução é funcional, realizando todas as funções que o trabalho se propõe, criando uma solução simples e eficaz, mas com muitas possibilidades para otimização. Espera-se que este trabalho tenha contribuído para explorar as perspectivas do que é possível ser feito com tecnologia de baixo custo, assim como as barreiras a serem superadas para as várias áreas de conhecimento funcionarem em conjunto.

**Palavras-chave:** IoT, Controlador Digital, Automação e Controle, Interface de Usuário, Supervisório, Base de Dados.

## ABSTRACT

The present academic work aims to provide a system capable of identifying and notifying a user about changes in a controlled plant using an IoT approach. To achieve this, the project joins together the areas of the internet, sensing, controlling, without fixed control parameters, user interfaces, databases, and of APIs in the cloud. The solution is capable of using a PID control for temperature, implemented through an Arduino board, sending data about the temperature and output voltage while receiving the controller gains from the API in each sampling cycle. The user interface presents the data to the user, in a comprehensive and intuitive way, while also enabling the user to alter the gains of the controller. The UI is also capable of notifying the user when a major change has happened in the plant if they exist. The API is run in the cloud and serves as a middleman for other modules to communicate with the database, managing HTTP requests. The results indicate the solution is functional, attending all requirements the project proposes to accomplish, being effective, but rather simple, leaving space for much further optimization. This project hopefully contributes to exploring perspectives about what is possible with low-cost technology while showing the barriers to be surpassed to make use of all the knowledge areas to work together.

**Keywords:** IoT, Digital Controller, Automation and Control, User Interface, Supervisory, Database.

## LISTA DE ILUSTRAÇÕES

1	Desenho do Sistema de controle de temperatura. . . . .	16
2	Fluxograma sobre o resumo do funcionamento geral do sistema. . . . .	21
3	Placa Arduino Mega montada com o Módulo Ethernet. . . . .	23
4	Croqui de da junção do aquecedor com a planta. . . . .	25
5	Gráfico simulado das curva de temperatura esperada com o uso do relé. . . . .	25
6	Diagrama de conexões da <i>API</i> com o banco, Arduino e supervisorio . . . . .	28
7	Croqui do resultado esperado para o sistema em funcionamento normal. . . . .	29
8	Ensaio ao Salto. . . . .	31
9	Ajuste do Controlador . . . . .	32
10	Implementação básica do controle digital . . . . .	33
11	Implementação no software Simulink do sistema de controle em malha fechada . . . . .	34
12	Implementação no software Simulink do sistema de controle em malha fechada com relés . . . . .	34
13	Arduino Mega, Módulo Ethernet e o par de relés. . . . .	35
14	Caixa de Acrílico com as Lâmpadas. . . . .	35
15	Supervisorio com os gráficos de temperatura, saída do relé e ganhos do controlador. . . . .	36
16	Caixa de acrílico com terra antes das alterações manuais . . . . .	37
17	Caixa de acrílico com terra após as alterações manuais. . . . .	38
18	Supervisorio com os gráficos de temperatura, saída do relé e ganhos do controlador após a mudança manual . . . . .	38
19	Tela de monitoramento do aplicativo da <i>API</i> publicada na plataforma Azure. . . . .	40
1	Fluxograma mais profundo do funcionamento do sistema. . . . .	46

## LISTA DE TABELAS

1	Características da placa Arduino Mega . . . . .	23
2	Características das Lâmpadas . . . . .	24
3	Características de cada lâmpada . . . . .	26
4	Características da Caixa . . . . .	26

## LISTA DE ABREVIATURAS

<b>CCA</b>	Curso de Eng. em Controle e Automação
<b>IoT</b>	<i>Internet of Things</i>
<b>PID</b>	Proporcional, Integral e Derivativo
<b>API</b>	<i>Application Programming Interface</i>
<b>HTTP</b>	<i>Hypertext Transfer Protocol</i>
<b>URL</b>	<i>Uniform Resource Locators</i>

## LISTA DE SÍMBOLOS

$R$	Resistência
$T$	Temperatura ou Período de Amostragem
$T_{amb}$	Temperatura Ambiente
$K_p$	Ganho proporcional
$K_i$	Ganho Integral
$K_d$	Ganho Derivativo
$K$	Função de Transferência Discretizada da Parte Proporcional do Controlador
$I$	Função de Transferência Discretizada da Parte Integral do Controlador
$D$	Função de Transferência Discretizada da Parte Derivativa do Controlador
$G$	Função de Transferência da planta
$C$	Função de Transferência do Controlador
$k$	Iteração do Controle
$e$	Erro
$p$	Polo
$\overline{\Delta t}$	Média de Intervalos de Tempo em que a Curva de Temperatura Permaneceu acima da Referência
$\overline{U}$	Média do Sinal de Controle

# SUMÁRIO

1	INTRODUÇÃO .....	14
2	REVISÃO DA LITERATURA.....	16
2.1	Função de Transferência de Processos Térmicos.....	16
2.1.1	Modelo da Função de Transferência .....	16
2.1.2	Identificação de Parâmetros da Planta.....	17
2.2	Controle PID .....	17
2.3	Controlador PID Discreto .....	17
2.3.1	Ação Proporcional.....	18
2.3.2	Ação Integral .....	18
2.3.3	Ação Derivativa.....	18
2.3.4	PID Discretizado.....	18
2.4	Arquitetura IoT.....	19
2.4.1	O que é IoT .....	19
2.4.2	Requisições HTTP .....	19
3	METODOLOGIA .....	21
3.1	Visão Geral da Arquitetura do Sistema.....	21
3.2	Sistema Embarcado.....	22
3.3	Sistema de Aquecimento.....	23
3.4	Medição de Temperatura .....	26
3.5	Planta de Controle .....	26
3.6	Supervisório .....	26
3.7	API na Nuvem.....	27
3.8	Identificação de Mudanças na Planta.....	28
3.8.1	Identificação por Média Móvel do Sinal de Controle .....	29
3.8.2	Identificação por Média Móvel do Intervalo de Cruzamento da Referência	30
4	RESULTADOS E DISCUSSÃO.....	31
4.1	Função de Transferência da Planta.....	31
4.2	Controlador .....	32
4.2.1	Ajuste do Controlador PID .....	32
4.2.2	Implementação Digital do Controlador .....	33



4.3	Implementação Física do Controle .....	34
4.4	Temperatura, Saída do Relé e Interface Gráfica .....	36
4.5	Alteração na Planta .....	37
4.6	Identificação de Mudanças .....	39
4.7	Chamadas da API na Nuvem .....	39
5	CONCLUSÕES E TRABALHOS FUTUROS.....	41
	REFERÊNCIAS .....	43
	ANEXOS .....	45

# 1 INTRODUÇÃO

O uso de composto orgânico, húmus, na agricultura pode ser considerado uma das atividades essenciais para uma sociedade sustentável (SEQUI, 1996). A compostagem, por sua vez, é o processo gerador deste composto, sendo de igual importância.

Para realizar uma compostagem de qualidade, muito cuidado deve-se ter com parâmetros deste processo, como, por exemplo, a temperatura. Como dizem Herbets et al. (2005, p.44) “a temperatura é um fator determinante no processo, uma vez que diferentes temperaturas promovem o desenvolvimento de diferentes micro-organismos. Este também é o parâmetro mais útil para monitorar a evolução da compostagem” (HERBETS et al., 2005). A temperatura deve alcançar no mínimo 40°C para que uma compostagem seja realmente feita, de acordo com o Manual de Compostagem Doméstica (COMLURB, 2021).

Quando o processo atinge essa temperatura, ele está na fase termófila, possibilitando a proliferação de bactérias características dessa fase e responsáveis pela continuação da decomposição (BERTICELLI et al., 2016). Entretanto, um estudo feito nos EUA, em 4000 casas com compostagens domésticas de pequeno porte, mostra que as temperaturas atingidas ficaram, em sua maioria, abaixo do mínimo durante todo o período (SMITH; JASIM, 2009). Esses dados mostram que é necessário um controle melhor desse parâmetro.

Já existem muitos métodos para controle da temperatura, como resfriamentos feitos por aeração, mas são voltados para sistemas maiores, como indústrias e fazendas, onde a escala é significativamente maior. Devido à grande quantidade de matéria orgânica nestes sistemas e da natureza exotérmica do processo de decomposição, a temperatura aumenta por si só, sendo necessário um resfriamento externo e não o contrário. Grandes quantidades de matéria orgânica em decomposição podem gerar temperaturas em torno de 70 °C (BERTICELLI et al., 2016). Este trabalho, entretanto, foca em sistemas de compostagem domésticos, que possuem somente uma fração de matéria orgânica que os sistemas maiores tem. Dessa forma, o calor gerado pela decomposição não é suficiente para aumentar a temperatura para acima de 40 °C, sendo necessário um auxílio de aquecimento vindo do controle.

Este trabalho tem por objetivo geral implementar uma arquitetura *IoT* que seja capaz de permitir um usuário de monitorar o estado de uma planta controlada remotamente, assim como ter a opção de alterar os parâmetros do controlador PID. Isso concede ao usuário uma liberdade de tomar decisões ao longo do processo, além de providenciar informações relevantes para estas tomadas de decisão. O sistema também deve ser capaz de indicar quando existe uma alteração significativa na planta controlada que requer uma possível mudança no controle. Como objetivos secundários, do presente trabalho procura-se projetar um controlador digital PID que possa controlar um sistema de aquecimento, enviando os seus dados de leitura de temperatura para a nuvem, além de buscar informações dela. Espera-se possibilitar que se possa fazer a manutenção de um sistema de compostagem doméstica, mantendo-o em uma temperatura ideal para a maturação do composto (COMLURB, 2021).

Este trabalho está organizado como segue. No capítulo dois, é desenvolvida a revisão da literatura. No seguinte, aborda-se a metodologia, seguida pela análise de resultados e discussões no capítulo subsequente. Por fim, as conclusões e considerações sobre trabalhos futuros são desenvolvidas no último capítulo.

## 2 REVISÃO DA LITERATURA

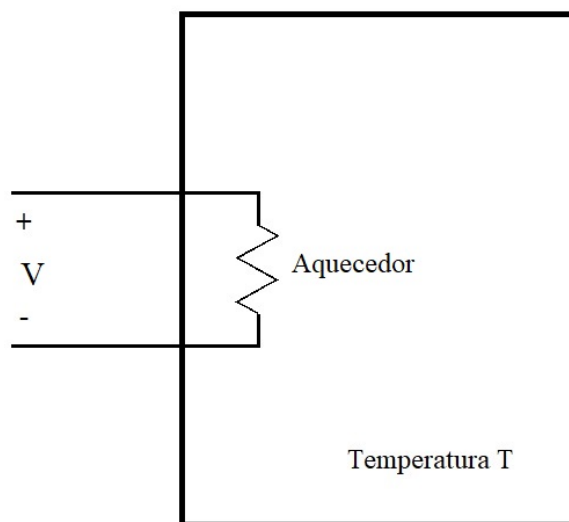
### 2.1 FUNÇÃO DE TRANSFERÊNCIA DE PROCESSOS TÉRMICOS

Para que um sistema seja considerado como um processo de troca de calor com o meio, sua variável de controle deve ser a temperatura.

#### 2.1.1 Modelo da Função de Transferência

Para sistemas simples de temperatura, a modelagem pode ser simplificada, como mostra o exemplo da Figura 1, onde as únicas interferências são vindas da resistência  $R$  e da temperatura ambiente  $T_a$ .

**Figura 1:** Desenho do Sistema de controle de temperatura.



Fonte: Adaptada de Bazanella e Silva Jr (2005, p. 99).

Como mostram Bazanella e Silva Jr (2005, p. 100), equação que representa a função de transferência entre a temperatura  $T$  e a entrada de tensão do processo  $V$  deste sistema é

$$\frac{T(s)}{V(s)} = \frac{e^{-\tau s}}{Cs + k_T} \quad (1)$$

onde  $\tau$  é o atraso de transporte, em segundos,  $C$  é a capacidade térmica do material a ser aquecido, e  $k_T$  é a constante de perdas característica das paredes do recipiente. É importante

notar que essa hipótese leva em consideração que o material possui temperatura homogênea por todo o seu volume, enquanto a heterogeneidade da temperatura é aproximada para um atraso de transporte, isto é, o tempo que demora para que o calor parta da fonte e chegue até o sensor onde sua influência é medida.

A Equação 1 caracteriza um sistema de primeira ordem, cuja equação pode ser descrita por

$$G(s) = \frac{G(0)}{Ts + 1} = \frac{G(0)/T}{s + 1/T} = \frac{K}{s + p} \quad (2)$$

onde  $T$  é a constante de tempo,  $K'$  é o ganho estático do processo ( $K = G(0)/T$ ) e  $p$  é o polo do processo, que é  $(1/T)$ . Note que as estruturas das equações de modelagem de processos de temperatura, Equação 1 e da Equação 2 são bastante similares, exceto devido ao atraso de transporte  $e^{-\tau s}$ . Dessa forma, pode-se atribuir os valores entre as duas equações, resultando em

$$\frac{T(s)}{V(s)} = \frac{(1/C)e^{-\tau s}}{s + k_T/C} = \frac{Ke^{-\tau s}}{s + p} \quad (3)$$

onde  $K$  é dado por  $1/C$  e  $p$  é  $k_T/C$ .

### 2.1.2 Identificação de Parâmetros da Planta

## 2.2 CONTROLE PID

Controle PID é um tipo de controle baseado em três ações: a proporcional; a integral; a derivativa ou PID, quando abreviadas. A ação proporcional é responsável por agir com base no erro presente (diferença entre o valor de referência e o valor atual), oferecendo uma resposta instantânea para variações de referências significativas. Já a ação integral é responsável por eliminar o erro, em sistema de primeira ordem, e seguir a referência em regime permanente (após um período de tempo suficientemente longo ter transcorrido). Enquanto isso, a ação derivativa tenta antecipar o processo e prever o erro, acelerando a convergência para o resultado em regime permanente (BAZANELLA; SILVA JR, 2005). A função de transferência que representa este tipo de controlador é

$$C(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{s + p} \quad (4)$$

onde  $K_p$  é o ganho proporcional,  $K_i$  é o ganho integral,  $K_d$  é o ganho derivativo,  $p$  é o polo rápido do sistema e  $s$  é a frequência.

## 2.3 CONTROLADOR PID DISCRETO

A discretização do controlador é o um passo essencial para que ele seja passível de ser implementado digitalmente. Como a Equação 4 mostra que o controlador é uma soma linear das ações de controle, pode-se também discretizar cada parte individualmente. O sinal de controle passa a não ser mais contínuo, mas atualizado somente a cada momento de amostragem. O instante de tempo para cada amostragem é representado por  $kT$ , onde  $T$ , neste caso, é o período de amostragem (BAZANELLA; SILVA JR, 2005).

### 2.3.1 Ação Proporcional

Como a ação proporcional não possui termo dependente da frequência, ela pode ser descrita simplesmente por

$$P(k) = K_p(e(k)) \quad (5)$$

### 2.3.2 Ação Integral

A ação integral, diferentemente da proporcional, possui dependência da frequência, mas, como mostram Astrom e Wittenmark (1997, p. 294), ela pode ser discretizada usando o método de diferenciação (Euler) em atraso, resultando em:

$$I(k) = I(k - 1) + K_i T e(k) \quad (6)$$

Note como a ação integral I depende da sua iteração passada  $k - 1$ , sendo k a iteração da amostragem atual. É possível usar os métodos de Euler por diferenciação em avanço ou a aproximação trapezoidal também (ASTROM; WITTENMARK, 1997).

### 2.3.3 Ação Derivativa

A ação derivativa, assim como a integral, possui a dependência na frequência. Usando o mesmo método de diferenciação (Euler) em atraso, pode-se chegar na seguinte equação:

$$D(k) = \frac{D(k - 1)}{1 + pT} + \frac{KT_d}{2 + pT}(e(k) - e(k - 1)) \quad (7)$$

Note como a ação derivativa também depende do seu resultado passado, mas ainda também depende do erro anterior  $e(k - 1)$ , mostrando que ela usa erros passados para melhorar o controle para a próxima iteração. Ela avalia a taxa de variação da temperatura e não somente o seu valor.

### 2.3.4 PID Discretizado

Juntando-se as equações 5, 6 e 7, que fizeram uso da aproximação de Euler, tem-se a equação final do PID discretizada:

$$C(k) = K_p e(k) + I(k - 1) + K_i T e(k) + \frac{D(k - 1)}{1 + pT} + \frac{pK_d}{1 + pT}(e(k) - e(k - 1)) \quad (8)$$

Cada ação, por ser parte de uma superposição das ações, pode ser calculada separadamente e depois somada com as outras, produzindo a resposta do controlador.

## 2.4 ARQUITETURA IOT

### 2.4.1 O que é IoT

*IoT* ou *Internet of Things* é a junção de objetos físicos que são monitorados com sensores, *software* e outras tecnologias, com o propósito de se conectar e transferir informações com a *internet* (ORACLE, 2023). Um exemplo de *IoT* pode ser visto nos sistemas de agricultura de grande porte, onde seus tratores têm sensores capazes de monitorar seu deslocamento, produção, posição e falhas e enviam sinais para o usuário e uma central de operações, onde elas são consumidas e o usuário pode tomar decisões remotamente ou deixar para um algoritmo mais inteligente tomá-las por ele.

Dessa forma, algo a ser considerado pertencente a uma arquitetura IoT deve possuir algumas características. Segundo Lea (2020, p. 10), é necessário que o dispositivo deva ser capaz de usar um *software* que utilize um protocolo de comunicação de *internet*; tenha um *hardware* que dê suporte para transmissão pela rede, que não pode ser um dispositivo tradicionalmente usado para se comunicar com a *internet*, como um computador, impressora ou *tablets* e *smartphones*. Esses quesitos indicam que um sensor, microcontrolador e outros dispositivos similares estão dentro da classificação de IoT.

### 2.4.2 Requisições HTTP

Como fazem parte do protocolo mais comum de comunicação pela internet, requisições HTTP precisam ser conhecidas para que dados sejam transportados do sistema embarcado para o outro ponto de acesso, como uma base de dados.

*URLs* (*Uniform Resource Locators*) são os locais onde os recursos são encontrados pelos navegadores de internet, para procurar, usar e compartilhar bilhões de dados (DAVID; BRIAN, 2002). Uma *URL*, em termos simplórios e gerais, é constituída da seguinte forma:

$$[Scheme].[Host]/[Path] \quad (9)$$

onde, o *Scheme*, no caso do protocolo HTTP, é o texto “http”, o *Host* é o endereço do servidor ao qual se quer se conectar, e o *Path* é o local onde está o recurso dentro do servidor ou, em outras palavras, é o que a requisição deseja encontrar (DAVID; BRIAN, 2002).

O protocolo mais utilizado para acessar as *URLs* é o HTTP, e existem quatro tipos de transações deste protocolo os pilares de toda a comunicação HTTP: *GET*, *POST*, *PUT* e *DELETE*, pertencente à arquitetura de aplicações *web* chamada *CRUD* (*Create*, *Read*, *Update*, *Delete*). Em termos gerais, as transações se diferem entre elas de acordo com o cabeçalho enviado na requisição HTTP, o que diz, para o servidor que vai recebê-las, o que ele deve esperar. Além disso, diferentes tipos de transações permitem diferentes tipos de dados a serem enviados: os métodos *GET*, por exemplo, permitem somente o envio de texto puro em suas requisições, enquanto os métodos *POST* permitem formatos de objetos mais complexos.

Segundo David e Brian (2002, p.48), as transações podem ser resumidas da seguinte forma:

- **GET** é responsável por enviar uma requisição de recursos para um servidor, esperando um documento como resposta.
- **POST** é responsável por enviar um valor ou estrutura de dados que o servidor deverá processar. Este método possui valores em seu corpo, como os objetos complexos.
- **PUT** é responsável por enviar um valor ou estrutura que o servidor deverá guardar em sua aplicação ou banco de dados. Este método possui valores em seu corpo, como os objetos complexos.
- **DELETE** é responsável por enviar uma chave para o servidor remover, da aplicação ou do banco de dados, o documento associado à ela.

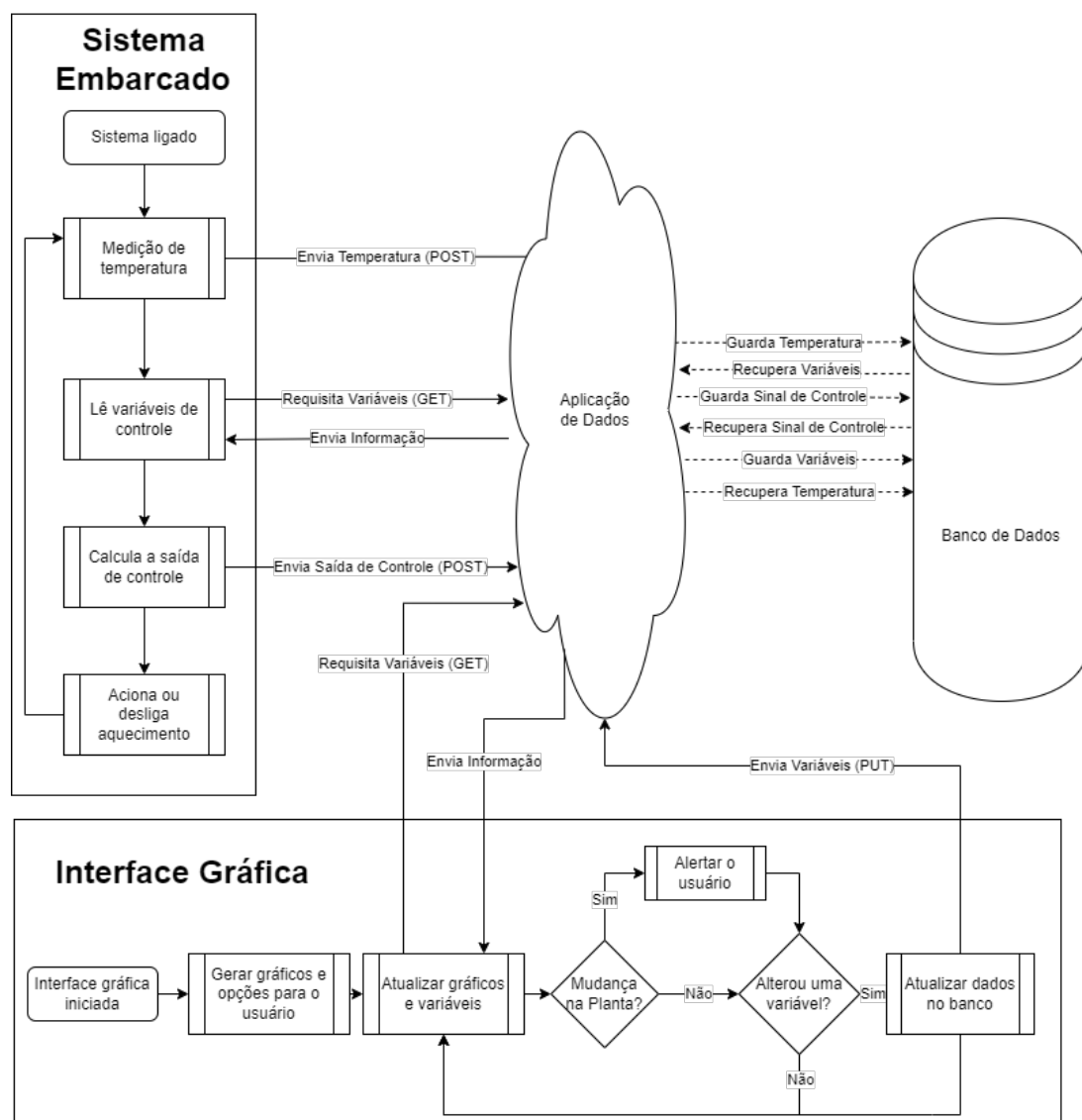


## 3 METODOLOGIA

### 3.1 VISÃO GERAL DA ARQUITETURA DO SISTEMA

A Figura 2 mostra um fluxograma resumindo o funcionamento geral da arquitetura *IoT*. Ela é composta pelo sistema Embarcado, administrado pelo novo controlador na placa Arduino; pela interface gráfica, representada por um *site* na *internet* e por uma *API*, responsável pelo intermédio das duas e pelo banco de dados, representada pela nuvem.

**Figura 2:** Fluxograma sobre o resumo do funcionamento geral do sistema.



Fonte: elaborado pelo autor.

O sistema embarcado tem por base o sistema Arduino. Quando ele é ligado, se dá o início ao controle da planta e a transmissão e recebimento de dados. O seu primeiro passo é receber a informação do sensor de temperatura sobre o estado da planta, guardando a informação e também a enviando para a nuvem, para manter ela atualizada. Após isso, ele requisita da nuvem as informações sobre as variáveis de controle  $K_p$ ,  $K_i$  e  $K_d$ , para usá-las no cálculo da saída de controle. Uma vez recebidas essas informações, o microcontrolador então calcula a saída de controle; envia ela para a nuvem e também faz o acionamento ou desligamento do aquecimento, iniciando novamente o *loop* do funcionamento do sistema embarcado.

A interface gráfica é gerida pelo *site* do supervisor na *internet*, o qual é explicado posteriormente. O seu processo começa quando o usuário acessa a página no seu endereço *web*, iniciando a geração dos gráficos e opções de interação do usuário. Após este processo de criação da interface, o *software* insere as informações no gráfico, fazendo uma requisição HTTP GET para a nuvem, a qual responde a chamada com os dados. O supervisor também é responsável por fazer a identificação de mudanças na planta, então ele procura por indícios de alteração e alerta o usuário se a resposta for afirmativa. Depois de resolver as informações da planta, o supervisor identifica se o usuário alterou os dados dos ganhos do controlador, enviando-os para a nuvem. Após esta decisão, a interface gráfica reinicia o seu processo a partir da atualização dos gráficos e variáveis, mantendo o usuário ciente do estado das variáveis em tempo real.

A nuvem é responsável pelo gerenciamento das informações utilizadas pelos outros elementos da arquitetura, recebendo e respondendo as requisições HTTP. O seu método de manutenção da informação é pela utilização banco de dados não-relacional MongoDB localizado em um servidor na nuvem, onde estas informações então são armazenadas.

Existem mais variáveis, que não estão representadas no fluxograma da Figura 2, como o que acontece quando uma requisição HTTP falha ou quando atinge o número máximo de tentativas antes de ser considerada uma falha no sistema. Entretanto, existe o diagrama com mais detalhes apresentado no Anexo A.1. Além disso, é importante notar que a decorrência dos eventos não necessariamente é feita sequencialmente, pois existem múltiplas *threads* de processamento para cada aplicação. Esse paralelismo é importante, pois, dessa forma, permite-se que a aplicação possa ter várias operações simultâneas.

## 3.2 SISTEMA EMBARCADO

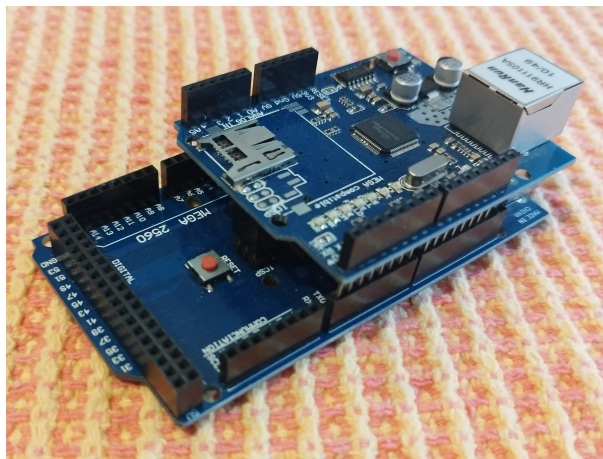
Para que seja possível medir e controlar a temperatura, além de transmitir os dados relacionados a ela para a nuvem, é necessário utilizar um sistema embarcado. Algumas possibilidades são o ESP32, o Arduino, o RaspberryPi, entre outros. Por motivos de praticidade e familiaridade de programação do autor e disponibilidade imediata, o Arduino Mega foi escolhido como o microcontrolador responsável por estas tarefas. Entretanto, essa placa não possui módulo *Ethernet* o que é necessário para a comunicação com a *internet* e subsequentemente com o supervisor.

A Figura 3 mostra a junção da placa microcontroladora Arduino Mega 2560 com o seu Módulo Ethernet. A sua ligação é simples, necessitando apenas o encaixe do módulo nos pinos de entrada e saída da placa, oferecendo a funcionalidade da conexão com o cabo e um *proxy* de *inputs* e *outputs*. A alimentação da placa é feita com uma fonte externa de 9 V, permitindo uma localização mais distante do computador.

**Tabela 1:** Características da placa Arduino Mega

Parâmetro	Valor
Microprocessador	ATmega2560
Pinagem	Pinos de LED: 13 I/O digitais: 54 Entradas analógicas: 16 Saídas PWM: 15
Tensão de Entrada	7-12 V
Tensão dos Pinos	0-5 V
Corrente máxima por pino	20 mA

Fonte: Arduino (2023).

**Figura 3:** Placa Arduino Mega montada com o Módulo Ethernet.

Fonte: o autor.

A Tabela 1 mostra os parâmetros da placa Arduino Mega que é utilizada no projeto.

### 3.3 SISTEMA DE AQUECIMENTO

Como parte dos objetivos do trabalho é a manutenção da temperatura da planta em 40°C, é necessária a presença de um sistema de aquecimento, pois a temperatura ambiente, em Porto Alegre, por exemplo, fica em torno de 20 e 32 °C no verão, o que não proporciona a temperatura necessária mencionada por COMLURB (2021). Em climas mais frios, a necessidade de aquecimento aumenta proporcionalmente. Para o aquecedor ser considerado suficiente, mantendo-se um temperatura de 40°C, ele deve ser capaz de esquentar o conteúdo do recipiente a no mínimo 45°C, pois uma margem de tolerância é importante para que o sinal de controle não fique sempre ativo e para que o sistema seja capaz de superar as barreiras de temperatura de ambientes mais frios.

O sistema de aquecimento escolhido para a planta é um conjunto de cinco lâmpadas de 21 W. Os motivos para ser usado esse conjunto de lâmpadas são: aquecimento rápido; possibilidade de serem desligadas e ligadas com frequência; não exalar odores como as resistências comuns (evitando a impregnação do cheiro no domicílio); ter uma temperatura máxima alcançada suficiente; possibilitar uma melhor distribuição do aquecimento, providenciando um aquecimento mais homogêneo.

A Tabela 2 a seguir mostra as características das lâmpadas automotivas incandescentes que são usadas para o aquecimento do material. É importante levar em consideração que essas medidas foram encontradas a partir de experimentos quando a temperatura do dia estava em torno de 30°C. Isso significa que a temperatura máxima atingida, por exemplo, é 70° C, mas as lâmpadas em si só conseguem elevar a temperatura até 40 °C acima da temperatura ambiente.

**Tabela 2:** *Características das Lâmpadas*

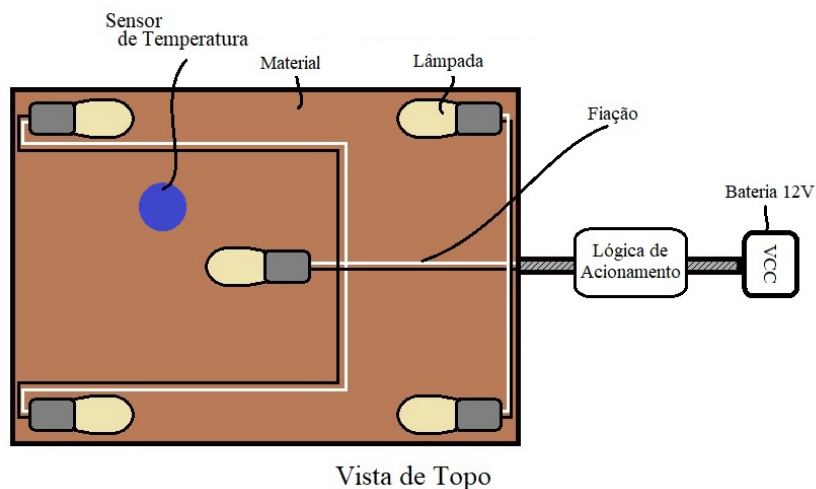
Parâmetro	Valor
Potência	21 W/Lâmpada
Dimensões com o <i>socket</i>	26.5x45x7.5 mm
Tensão de operação	12 V
Corrente de operação	1,75 A
Temperatura máxima atingida	70 °C ( $T_{amb} = 30^{\circ}\text{C}$ )
Tempo até a temperatura máxima	2 min

Fonte: elaborado pelo autor.

A Figura 4 mostra um croqui da instalação das lâmpadas. Elas são colocadas dentro do material, espalhadas pelos cantos e pelo meio do aquecedor, estando todas no mesmo plano. Com o espalhamento simétrico delas ao longo do material, espera-se que a distribuição da temperatura fique mais homogênea em toda a caixa, além de diminuir o atraso de transporte que haveria com somente uma lâmpada? O conjunto de todas as lâmpadas fornece uma potência de 105 W, distribuída pelo material, com um consumo de corrente total de 8,75 A. Este valor de consumo de corrente é muito superior àquele capaz de ser fornecido por uma saída do Arduino (Tabela 1), mostrando-se necessária o acoplamento de uma bateria externa.

A bateria externa utilizada deve fornecer uma tensão de 12 V e uma corrente maior do que 8,75 A. É possível suprir essas necessidades utilizando múltiplas baterias pequenas ou um grande transformador, mas existe também a possibilidade de se utilizar uma bateria automotiva. Essas baterias possuem exatamente a tensão de saída esperada de 12V e esta saída se mantém ao longo do consumo de sua carga. Além disso, como as baterias automotivas são usadas para alimentar múltiplas acessórios do automóvel, elas têm um fornecimento máximo de corrente significativamente superior aos 8,75A necessários para sustentar as lâmpadas de aquecimento.

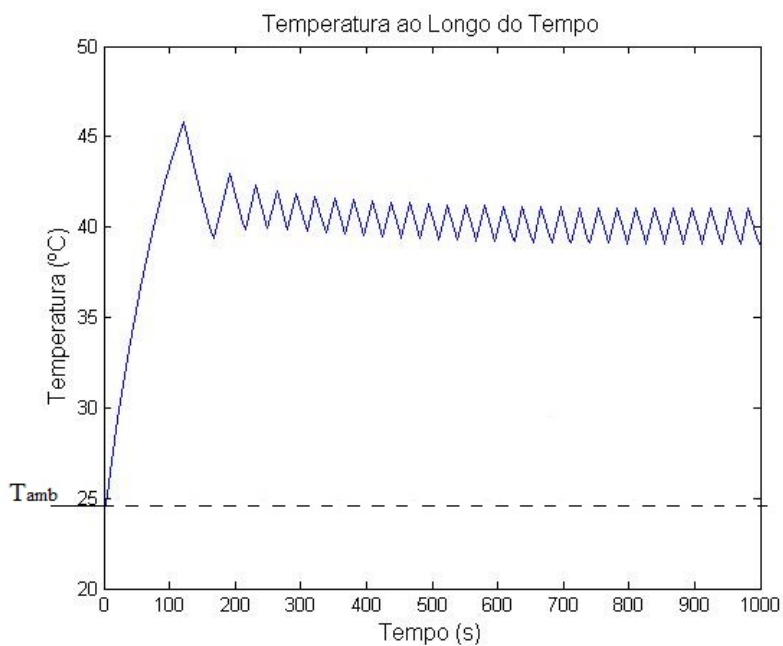
**Figura 4:** Croqui de da junção do aquecedor com a planta.



Fonte: elaborado pelo autor.

É possível ver na Figura 4 que a lógica de acionamento é uma parte essencial do sistema de aquecimento, pois ele é responsável por ligar e desligar as lâmpadas. Como as lâmpadas são abastecidas por uma bateria 12 V que não possui uma regulagem de tensão, a lógica de acionamento pode ser feita com relés. Eles são capazes de ligar e desligar as lâmpadas, mantendo a temperatura controlada, mesmo que oscilando. Dessa forma, o comportamento esperado da temperatura seria uma oscilação ao redor da referência de 40 °C, como mostra a Figura 5, onde  $T_{amb}$  é a temperatura ambiente.

**Figura 5:** Gráfico simulado das curva de temperatura esperada com o uso do relé.



Fonte: elaborado pelo autor.

### 3.4 MEDIÇÃO DE TEMPERATURA

A medição de temperatura é um fator essencial para o controle, pois relaciona o estado da variável de interesse com a referência. O sensor DHT22 é escolhido para ser o responsável pela medição, visto que pode trabalhar em faixas de temperatura de até 80 °C e por sua facilidade de uso. A Tabela 3 mostra mais características deste sensor.

**Tabela 3:** *Características de cada lâmpada*

Parâmetro	Valor
Tensão de Operação	5 V
Faixa de medição de temperatura	-40 a 80 °C
Tempo de resposta mínimo	2 s
Resolução	0,1 °C

Fonte: Liu (2023).

A medição de temperatura é feita na posição indicada pelo ponto azul na Figura 4, onde o sensor está colocado entre a lâmpada superior esquerda e a lâmpada central.

### 3.5 PLANTA DE CONTROLE

Tendo em vista que o estudo de caso do trabalho é o controle da temperatura de sistemas de compostagem domésticos e de pequena escala, a planta de controle é um modelo simplificada desse tipo de ambiente: uma caixa de acrílico com terra no seu interior, procurando se aproximar do que seria uma situação com matéria orgânica nos casos reais. As dimensões dessa caixa apresentam as seguintes medidas representadas na Tabela 4.

**Tabela 4:** *Características da Caixa*

Parâmetro	Valor (cm)
Comprimento	35
Largura	25
Altura	20

Fonte: Elaborado pelo autor.

Conforme Bazanella e Silva Jr (2005, p. 116), os parâmetros de uma planta de controle de primeiro grau (Equação 2) podem ser dados pelo tempo de acomodação  $t_s$ , medido quando a planta atinge 98% do seu valor de regime permanente, e  $K$ , dado pelo valor de regime permanente da planta dividido pela entrada (neste caso em V). Essa simplificação pode ser feita se o atraso da planta for desprezível.

### 3.6 SUPERVISÓRIO

O supervisório é responsável por mostrar para o usuário, usando uma interface gráfica intuitiva, o comportamento das duas variáveis principais ao longo do tempo; a temperatura, os valores dos ganhos do controlador,  $K_p$ ,  $K_i$  e  $K_d$ , e o valor da saída de tensão

para as lâmpadas. O método para fazer isso é usando uma comunicação via *internet* com a *API* na nuvem, usando o protocolo HTTP. Para isso, o supervisor tem três métodos GET: um para requisitar dados para a nuvem sobre a temperatura em um determinado intervalo de tempo; um para requisitar dados sobre a saída de tensão que entra nas lâmpadas; um para requisitar os valores dos ganhos  $K_p$ ,  $K_i$  e  $K_d$ . Estes métodos requisitam informações da nuvem, a as busca no banco de dados MongoDB.

Além disso, o supervisor deve permitir a mudança parâmetros dos ganhos do controlador livremente, habilitando a possibilidade de mudar o desempenho do controle remotamente. Para isso, a aplicação usa uma requisição HTTP do tipo POST para enviar os dados para a *API* na nuvem, a qual processa as informações e as armazena no banco de dados.

A aplicação do supervisor é produzida com a tecnologia da plataforma .NET, da Microsoft, mais especificamente usando Blazor e a linguagem C#. A arquitetura do sistema é de micro-serviços, tendo o supervisor como interface gráfica na *web*, enviando requisições HTTP para a *API* na nuvem, a qual é responsável por se comunicar com o banco de dados não relacional MongoDB, sendo uma *API* (*Application Programming Interface* ou Interface de Programável para Aplicação).

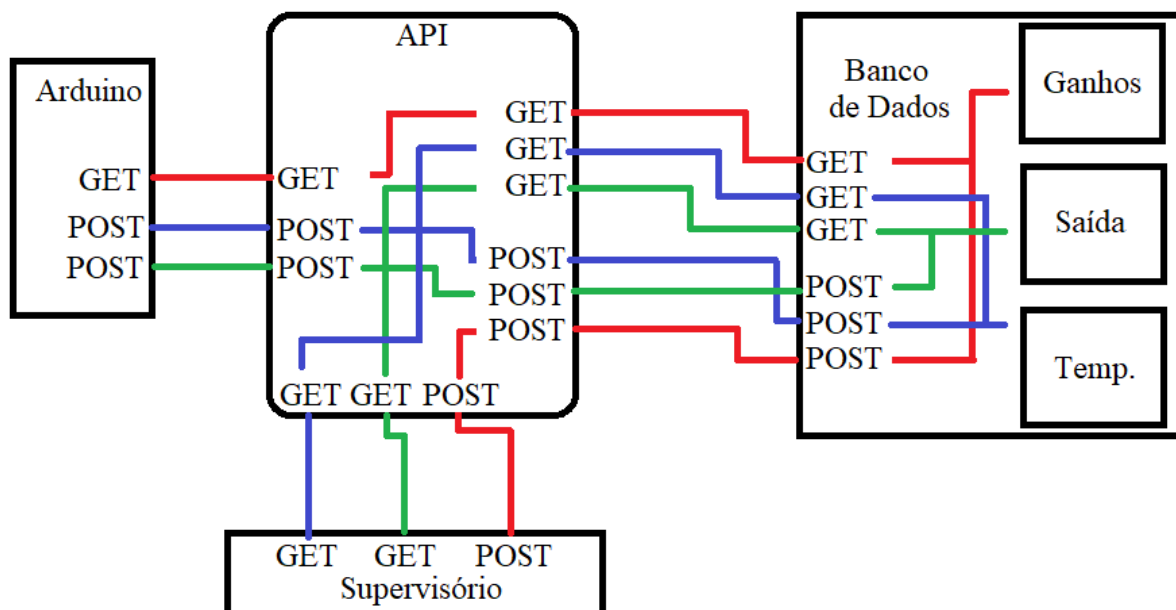
### 3.7 API NA NUVEM

Enquanto o Arduino e o supervisor são responsáveis por gerir o sistema embarcado e mostrar para o usuário os resultados, respectivamente, a *API* é responsável por gerir todas as informações referentes às necessidades da arquitetura. Esta aplicação tem um canal de comunicação com o banco de dados MongoDB, também situado na nuvem, mas em servidores localizados em partes diferentes do planeta. Este banco recebe as requisições da *API* para enviar informações, que são então entregues para o supervisor e para o Arduino, e também para armazenar informações, como os dados de temperatura, tensão de saída e ganhos do controlador, vindos também das outras frentes da arquitetura.

Os métodos de comunicação da *API* com os módulos externos também é feito com o protocolo HTTP pela internet, mas ela exige uma autenticação antes de permitir que a comunicação estabelecida. Apesar de ser um sistema simples de proteção que usa usuário e senha, somente o Arduino e o supervisor possuem tais credenciais.

A Figura 6 mostra um diagrama que representa todas as conexões que a *API* gerencia, representando por cores sobre qual informação está sendo transmitida e de quem vem. As cores em vermelho representam informações sobre os ganhos do controlador, em verde são as informações sobre a saída de tensão vinda da bateria e em azul são as informações sobre a temperatura medida pelo sensor. É importante notar que a *API* faz o encaminhamento das requisições, manejando alguns dados pertinentes à composição da chamada e expectativa de resultados.

**Figura 6:** Diagrama de conexões da API com o banco, Arduino e supervisor



Fonte: elaborado pelo autor.

Um exemplo de funcionamento é quando o Arduino faz uma requisição HTTP GET para receber os dados dos ganhos do controlador. A API recebe esta requisição e a ela faz sua própria requisição para o banco, que acessa os dados dos ganhos, retorna para a API que, por fim, retorna para o Arduino. O mesmo ciclo é feito para todas as requisições HTTP vindas externamente. O motivo de não haver uma reutilização de requisições GET é para manter uma consistência de dados, de segurança e velocidade para a API, além de não necessitar que a escolha de caminhos seja feita via os dados enviados pelos remetentes.

Esta separação de trabalhos é característica de uma arquitetura preparada para micro-serviços, pois cada tipo de serviço é delegado para uma aplicação específica. Neste caso, o Arduino é responsável somente pelo controlador e medição de variáveis, enquanto o supervisor só se importa com mostrar as informações para o usuário. O manejo dos dados e a complexidade de comunicação com o banco de dados não deve ser responsabilidade nem do Arduino e nem da interface gráfica, sendo a principal razão para o desenvolvimento da API.

### 3.8 IDENTIFICAÇÃO DE MUDANÇAS NA PLANTA

Existem dois métodos possíveis para encontrar mudanças drásticas na planta: o método de verificação da média móvel do sinal de controle e o método de verificação por média movel do intervalo de quando a curva cruza a linha de referência. Esse método de identificação é específico para este sistema e foi pensado para este trabalho, pois o foco do projeto não é exclusivamente encontrar uma maneira ótima de identificar estas alterações.

Para este trabalho, a média móvel faz uso da média dos valores dentro de um intervalo de 1h de operação por dia durante os últimos 5 dias. Isso significa que, quando o sistema avalia os dados em relação ao seus valores iniciais, ele está fazendo uma média das

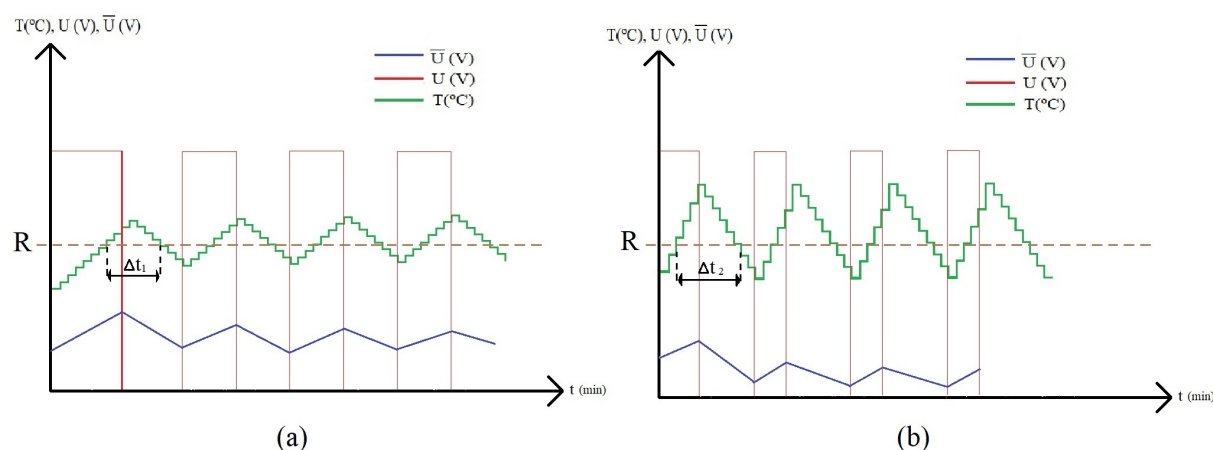


últimas 5 coletas diárias de dados e comparando com os valores iniciais e suas tolerâncias estipuladas.

### 3.8.1 Identificação por Média Móvel do Sinal de Controle

Como o Arduino controla um relé, o sinal de controle efetivo é um pulso. A resposta esperada do sistema em regime permanente, antes de haver qualquer alteração nele, está representada na Figura 7.a, a qual apresenta um croqui das curvas de temperatura,  $T$ , sinal de controle,  $U$ , e média móvel do sinal de controle,  $\bar{U}$ . Quando a planta está em condições normais, a temperatura se mantém fluando ao redor de  $40^\circ\text{C}$ . A média móvel do sinal de controle aumenta e diminui, conforme a ativação do relé. Esta média móvel específica do sistema é então salva para uso posterior, pois ela representa o indicador de alteração da planta.

**Figura 7:** Croqui do resultado esperado para o sistema em funcionamento normal.



Fonte: elaborado pelo autor.

Já a Figura 7.b representa o croqui da saída esperada, também em regime permanente, do sistema quando houver a compactação de material, isto é, alteração na planta. O resultado disso é que o sistema de aquecimento, tendo sido projetado para um material com propriedades diferentes, irá aquecer muito mais rápido do que antes, visto que as lâmpadas usadas têm um aquecimento rápido. Além disso, o resfriamento do material tem um aumento menos significativos quando comparado com o aumento do aquecimento. O motivo desta discrepância é porque, segundo Xu et al. (2020), a compactação do material cria uma camada de isolamento térmica que pode diminuir significativamente a condutividade e difusividade térmica do material. Dessa forma, as ativações do relé estarão mais espaçadas do que antes, resultando em uma diminuição da média móvel de sinal de controle. Assim, quando comparada com a média de funcionamento normal, aponta uma diferença acima de uma tolerância arbitrária escolhida com base nos testes. Esse é o indicador medido para sinalizar posteriormente para o usuário que a planta sofreu uma alteração significativa e o controlador pode ou não ser alterado.

É importante notar que esse procedimento leva em consideração que o aumento no ganho da temperatura pelo sistema provocado pelas lâmpadas é maior do que o aumento das perdas de calor e temperatura por ter menos material. Esse método também faz um

uso simplório do que seria uma variação radical na planta, pois usa tolerâncias arbitrárias e também é totalmente dependente de testes para ser posto em prática.

### **3.8.2 Identificação por Média Móvel do Intervalo de Cruzamento da Referência**

O método de identificação por intervalo de cruzamento da referência faz uso da diferença de intervalos de tempo em que a curva da temperatura está acima (ou abaixo) da linha de referência. Na Figura 7.a é possível ver que o intervalo de tempo em que a curva está acima da linha de referência é  $\Delta t_1$ . Já na Figura 7.b, o intervalo é denominado por  $\Delta t_2$ . Entretanto, espera-se que o sistema esquite mais rápido quando há uma alteração substancial na planta, como a sua compactação. Entretanto, a perda de calor para o ambiente também aumenta, mas, como mostrado por Xu et al. (2020), devido a compactação da superfície e menor quantidade de ar dentro do material, é relativamente menor do que o aumento da temperatura.

Com a diferença na compactação do material, é possível inferir que  $\Delta t_2 > \Delta t_1$ , pois a subida da temperatura é maior, enquanto a descida dela é proporcionalmente menor. Quando o material está suficientemente compactado, o sistema ficara menos tempo ligado esquentando e mais tempo esfriando, o que resulta em um intervalo de tempo maior em que a curva de temperatura fica acima da referência.

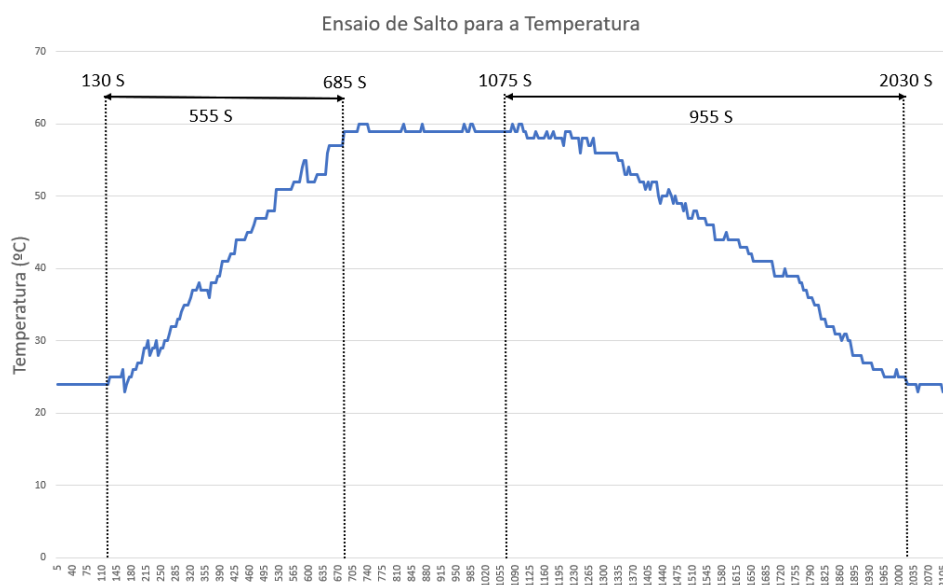
## 4 RESULTADOS E DISCUSSÃO

### 4.1 FUNÇÃO DE TRANSFERÊNCIA DA PLANTA

De acordo com Bazanella e Silva Jr (2005, p. 116), a função de transferência de primeira ordem pode ser extraída com a identificação dos parâmetros vindos de um ensaio ao salto. Este ensaio consiste em aplicar uma saída específica instantânea no atuador e observando o comportamento da variável de controle que é, neste caso, a temperatura. Após isso, é possível extrair o ganho estático, polo e atraso, obtendo-se a função de transferência da planta.

A Figura 8 mostra o ensaio ao salto feito no sistema. Foram aplicados 12 V nas lâmpadas no instante 130 s, usando o tempo de amostragem de 5 s. Observando-se o comportamento da temperatura, é possível verificar que o tempo de acomodação  $t_s$ , encontrado quando o valor da temperatura chega em 98% do valor final (aproximadamente 59 °C) é  $t_s = 685 - 95 = 555$  s. O ganho,  $K$ , é  $K = \frac{60-24}{12V} = 3$  °C/V. Enquanto isso, o atraso,  $\tau$ , foi medido em um teste separado, usando um tempo de amostragem de 2 s e valores de temperatura com resolução de 0,1 °C, resultando em um valor de 2 s.

**Figura 8:** *Ensaio ao Salto.*



Fonte: o autor.

Com estes resultados, é possível reescrever a Equação 2 com os novos valores:

$$G(s) = \frac{K}{s + p} = \frac{3e^{-2s}}{1 + 130s} \approx \frac{3}{1 + 130s} \quad (10)$$

O atraso no sistema de 2 s representa aproximadamente 0,3% do tempo de acomodação e pode ser desconsiderado. Entretanto, é importante notar que o tempo de amostragem deve ser superior a 2s para que ele realmente seja desconsiderado e, por essa necessidade, o tempo de amostragem escolhido foi de 5 s.

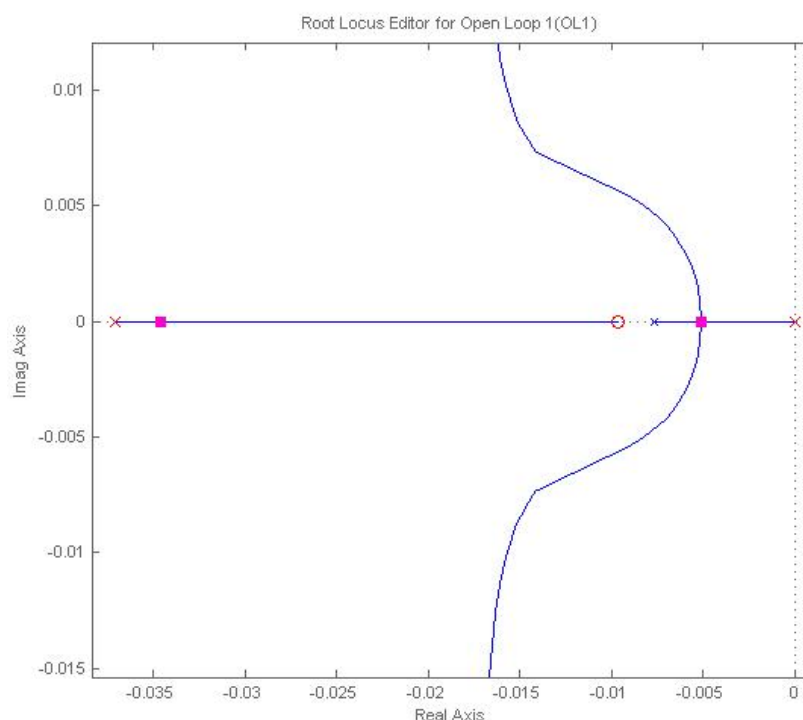
É importante notar que a temperatura da planta decresce mais lentamente do que sua velocidade de subida. Um dos motivos disso é o próprio aquecimento das lâmpadas responsáveis por aquecer a planta. Quando as lâmpadas param de aquecer, elas ainda estão transmitindo calor e estão com a temperatura alta e isso faz com que o sistema como um todo preserve a temperatura por um tempo mais prolongado. Isso também significa que o processo de resfriamento da planta é gerido por uma função de transferência diferente da calculada no início. O segundo motivo para que o decrescimento mais lento da temperatura é o fato de haver menos ar dentro do composto devido a compactação, além da superfície estar também com menor capacidade térmica.

## 4.2 CONTROLADOR

### 4.2.1 Ajuste do Controlador PID

O ajuste do controlador pode ser feito uma vez que a função de transferência da planta é conhecida, pois pode ser realizado graficamente com o auxílio da ferramenta *rltool* fornecida pelo *software* Matlab. A Figura 9 mostra o diagrama de polos e zeros da planta quando afetado pelo controlador PID. Os polos do controlador foram ajustados para não conterem parte imaginária em malha fechada com a planta, o que resultaria em uma resposta oscilatória em regime permanente.

**Figura 9:** Ajuste do Controlador



A partir do ajuste do controlador usando a ferramenta, encontra-se a função de transferência do controlador:

$$C(s) = \frac{0,00105(1 + 200s)}{s(1 + 27s)} \quad (11)$$

da qual é possível extrair  $K_p = 0,182$ ,  $K_i = 0,00105$  e  $K_d = \frac{-4,9}{27} = -0,1815$ . Mantendo o período de amostragem  $T = 5$  s e o polo  $p = \frac{1}{27} = 0,037$  o que permite a reescrita da Equação 8, encontrando-se

$$C(k) = 0,182e(k) + I(k-1) + 0,00525e(k) + \frac{D(k-1)}{1,185} + 0,0283(e(k) - e(k-1)) \quad (12)$$

#### 4.2.2 Implementação Digital do Controlador

Com a equação da função de transferência discretizada do controlador da Equação 8, é possível fazer a implementação deste controlador digitalmente no Arduino, como mostra a Figura 10.

**Figura 10:** Implementação básica do controle digital

```
void loop() {
  y = dht.readTemperature(); // Temperature Reading
  e = ref - y;
  P = 0.182*e;
  I = last_I + 0.00525*(e+last_e);
  D = last_D/1.185 + 0.0283*(e-last_e);
  u = P + I + D;
  last_I = I;
  last_D = D;

  // Saturation
  if (u > 12)
    u = 12;
  else if (u < 0)
    u = 0;

  delay(5000); // 5 s delay between each sampling
}
```

Fonte: o autor.

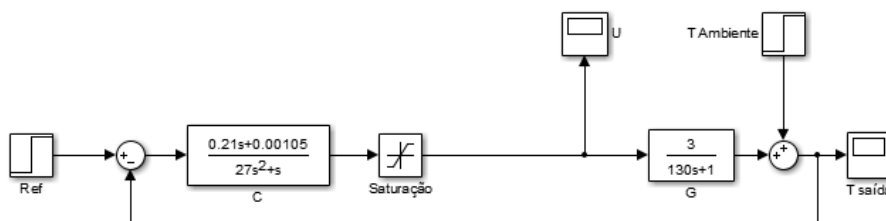
Todo o funcionamento do controlador está dentro de uma função *loop*, que é própria da linguagem de programação do Arduino e é responsável por executar o código infinitamente em repetição, onde  $y$  é a medição da temperatura,  $e$  é o erro,  $P$ ,  $I$  e  $D$  são as ações proporcional, integral e derivativa, respectivamente,  $u$  é o sinal de controle,  $last\_I$  e  $last\_D$  são os resultados da última iteração das ações integral e derivativa. Nesta figura também é possível observar que uma lógica está imposta por meio das cláusulas *IF* e

*ELSE IF*, respectivamente. Essas cláusulas são responsáveis por representar digitalmente a saturação da saída de controle, pois ela é limitada a estar entre 0 e 12 V, pois a bateria usada para o acionamento das lâmpadas não pode ser sua polaridade revertida e as próprias lâmpadas não podem ter o seu calor retirado.

### 4.3 IMPLEMENTAÇÃO FÍSICA DO CONTROLE

A Figura 11 mostra como fica o sistema de controle em malha fechada. A implementação do controle (neste caso, contínuo e não discreto), tem um bloco de saturação que representa as limitações físicas da bateria e das lâmpadas que o controle apresenta na realidade.

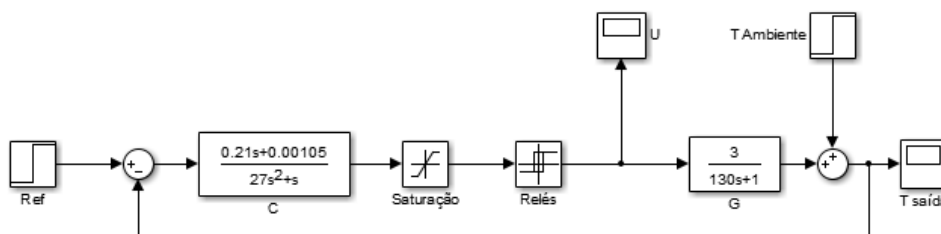
**Figura 11:** Implementação no software Simulink do sistema de controle em malha fechada



Fonte: o autor.

Existem ainda outros fatores que impedem a implementação exatamente como está mostrada na Figura 11, como a necessidade de utilizado um relé para o acionamento das lâmpadas. Isso é importante, pois significa que o controle não é obedecido totalmente pela magnitude do sinal de controle e oscila por natureza, dado que o relé só possui dois estados, *on-off*. Dessa forma, uma representação mais fiel do funcionamento está representada na Figura 12, onde o bloco do relé está também inserido.

**Figura 12:** Implementação no software Simulink do sistema de controle em malha fechada com relés

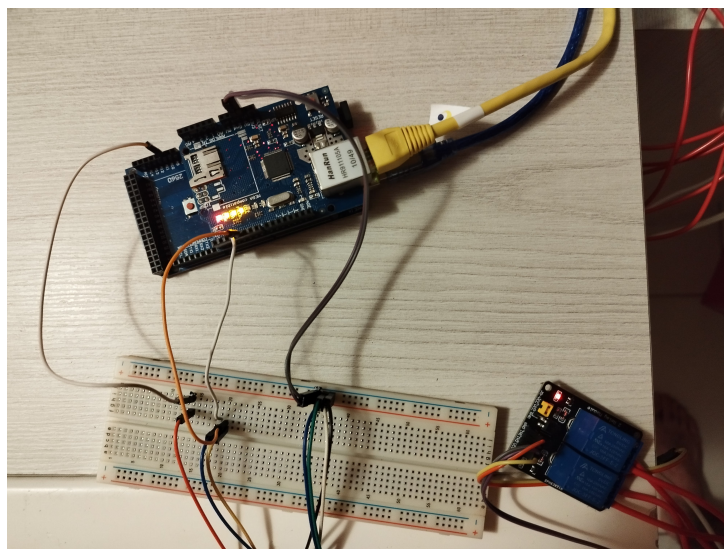


Fonte: o autor.

A implementação real pode ser vista na Figura 13, onde está o Arduino Mega acoplado ao módulo *ethernet*, a placa *protoboard* e dois relés. O Arduino se conecta a *internet* e faz as requisições HTTP usando o módulo *ethernet* enquanto recebe informações do sensor de temperatura DHT22 e envia os sinais de acionamento para o relé utilizando

suas entradas e saídas digitais. O motivo de serem usados dois relés é para dividir a passagem de corrente por cada um deles, visto que estes são capazes de conduzirem somente 5 A por unidade. Com a utilização de dois, cada relé conduz aproximadamente 4,4 A.

**Figura 13:** *Arduino Mega, Módulo Ethernet e o par de relés*

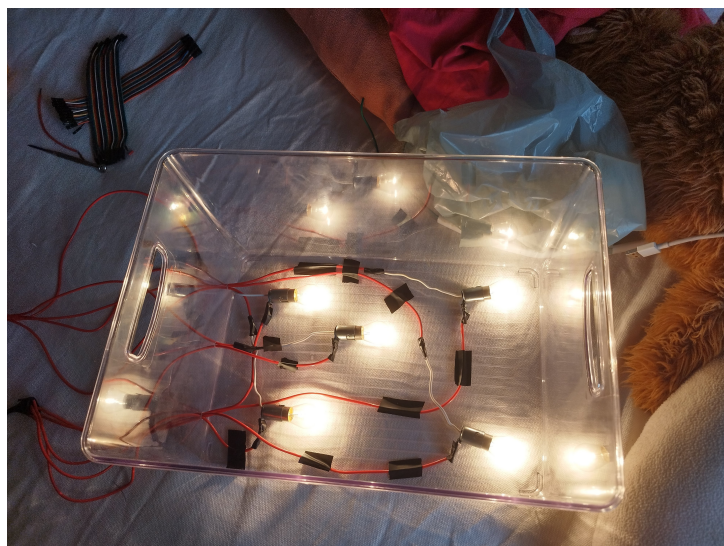


Fonte: o autor.

A caixa de acrílico foi acoplada às lâmpadas, como mostra a Figura 14. As lâmpadas foram presas à caixa com auxílio de fita isolante, mas mantendo uma certa folga perto do bulbo. O motivo da folga é poder levantar as lâmpadas e permitir que terra abaixo, prevenindo o aquecimento demasiado da caixa, que poderia provocar o derretimento dela.

Com todas as lâmpadas funcionando, ao ar livre e com temperatura ambiente estando em torno de 20 °C, a temperatura atingida máxima é, de maneira próxima à homogênea, de aproximadamente 85°C. Mesmo se a eficiência das lâmpadas decaia para a metade, ainda é possível fazer o material atingir a referência de 40 °C.

**Figura 14:** *Caixa de Acrílico com as Lâmpadas*

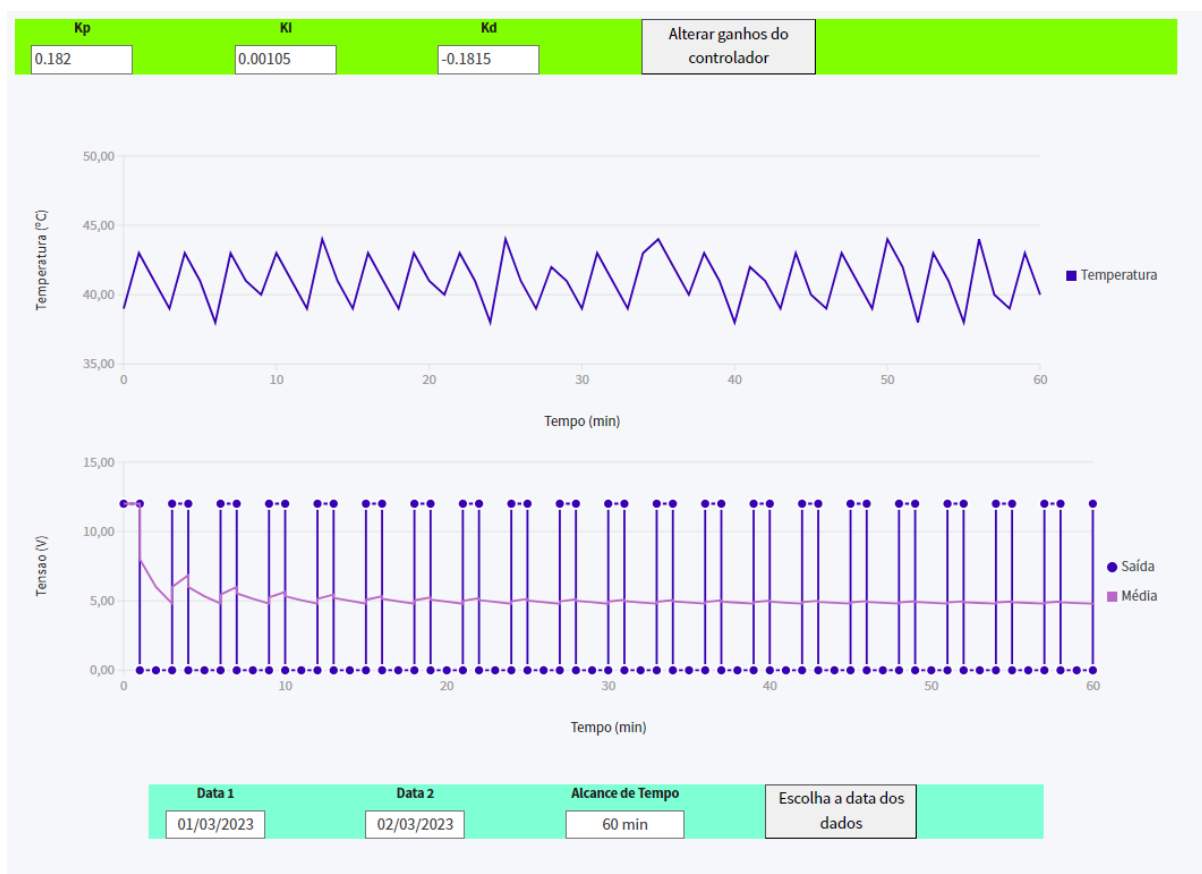


Fonte: o autor.

#### 4.4 TEMPERATURA, SAÍDA DO RELÉ E INTERFACE GRÁFICA

A Figura 15 mostra a visão do supervisor, onde é possível ver que existem caixas de texto com os ganhos  $K_p$ ,  $K_i$  e  $K_d$  do controlador, um botão para alterar os ganhos no Arduino, um gráfico de temperatura pelo tempo, um gráfico do saída de tensão dos relés e de sua média pelo tempo e um seletor de alcance de tempo e datas.

**Figura 15:** *Supervisor com os gráficos de temperatura, saída do relé e ganhos do controlador*



Fonte: o autor.

Os valores e curvas apresentados na Figura 15 foram extraídos durante o experimento quando a caixa estava com maior quantidade de material, ou seja, antes da retirada e compactação de terra. A Figura 16 mostra o estado da caixa, tendo uma quantidade de terra fofa que se aproxima do topo.



**Figura 16:** Caixa de acrílico com terra antes das alterações manuais



Fonte: o autor.

Os gráficos de temperatura e saída do relé estão com um alcance de tempo de 60 minutos, o que pode ser modificado através dos seletores de tempo abaixo dos gráficos. Entretanto, não é possível alterar o intervalo de tempo entre as medidas, tendo o tempo de amostragem para os gráficos (o que difere do controlador) de 1 minuto, facilitando a visualização do geral. Caso o usuário venha a escolher um alcance de tempo muito grande, o gráfico fica poluído. Caso o usuário escolha uma data para a qual não exista valores, nada é mostrado nos gráficos.

A temperatura, em regime permanente, permaneceu predominantemente entre  $39\text{ }^{\circ}\text{C}$  e  $43\text{ }^{\circ}\text{C}$ , oscilando em torno  $41\text{ }^{\circ}\text{C}$ . A Figura 15 também mostra que houve valores de temperatura superiores a  $43\text{ }^{\circ}\text{C}$  e inferiores a  $39\text{ }^{\circ}\text{C}$  em certos momentos. Enquanto isso, a saída de tensão dos relés oscila entre os valores de 0 e 12 V, que é a tensão da bateria ao qual os relés estão conectados. A média da tensão de saída, em regime permanente, convergiu para 5 V, o que é abaixo da metade de 12 V. A média dos intervalos de tempo que a curva de temperatura ficou acima da referência, com base em 30 amostras, em regime permanente, é de 2 minutos e 38 s.

Os valores dos ganhos do controlador na Figura 15 são os valores projetados na seção 4.2.1, mas estes podem ser alterados a qualquer momento. Entretanto, os valores só são atualizados na placa Arduino a cada ciclo de amostragem, isto é, somente a cada 5 s o Arduino busca por novos parâmetros. Não existe validação quanto a qualidade dos ganhos, permitindo que o usuário potencialmente deixe o sistema em malha fechada longe do ideal.

## 4.5 ALTERAÇÃO NA PLANTA

A Figura 17 mostra como ficou o material dentro da caixa de acrílico após a retirada de um pouco de terra, seguido de uma compactação manual do restante.

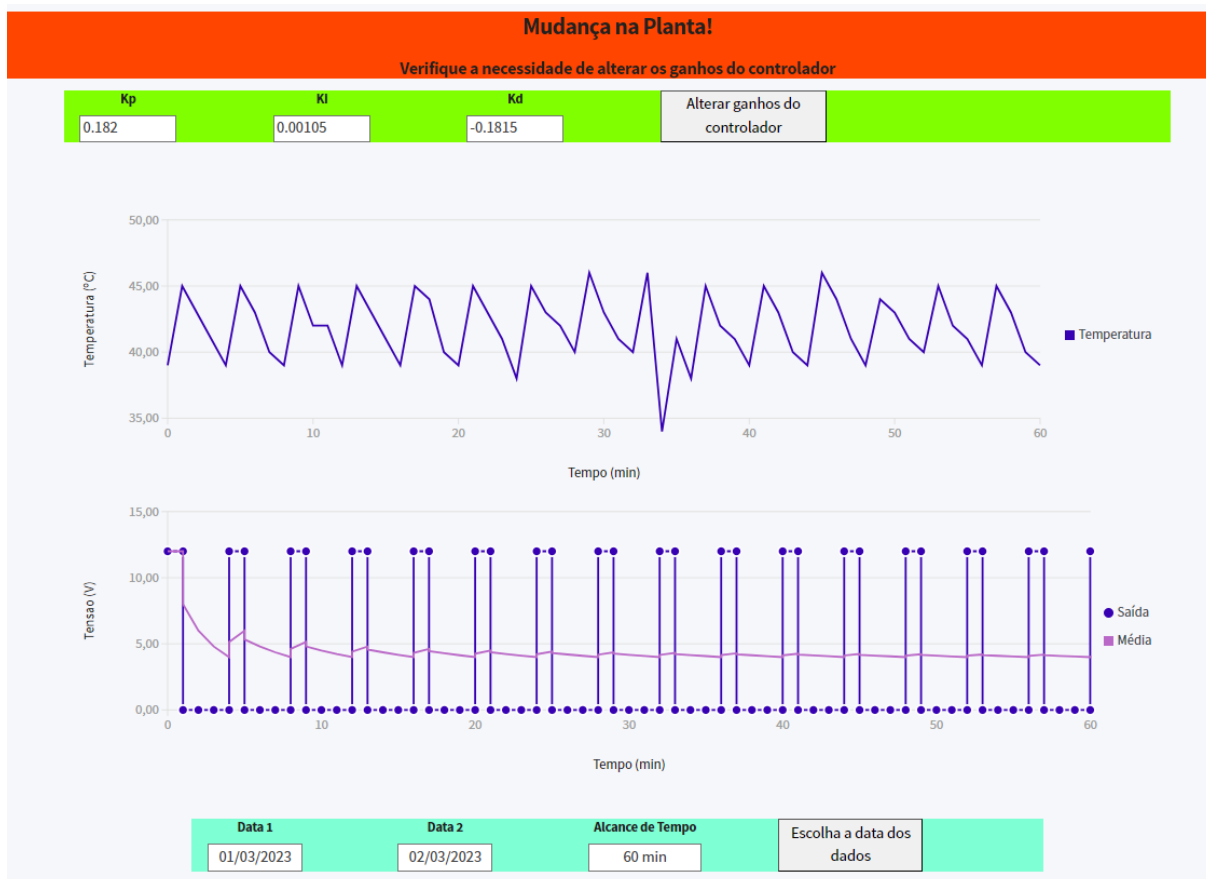
**Figura 17:** Caixa de acrílico com terra após as alterações manuais



Fonte: o autor.

Aplicando o mesmo controle sobre a planta, com os mesmos ganhos, resulta nos dados mostrados na tela do supervisor, apresentada pela Figura 18.

**Figura 18:** Supervisor com os gráficos de temperatura, saída do relé e ganhos do controlador após a mudança manual



Fonte: o autor.

A temperatura, em regime permanente, se manteve predominantemente em torno de 39 °C e 45 °C, com ocasionais altos para 46 °C e baixos de 34 °C. A saída de tensão dos relés novamente oscilou entre 0 e 12 V, mas tendo um minuto a mais de espaçamento entre as ativações. A média de tensão de saída, em regime permanente, convergiu para aproximadamente 4,10 V. A média dos intervalos de tempo em que a curva de temperatura ficou acima da referência de 40 °C foi de aproximadamente 3 minutos e 25 s.

## 4.6 IDENTIFICAÇÃO DE MUDANÇAS

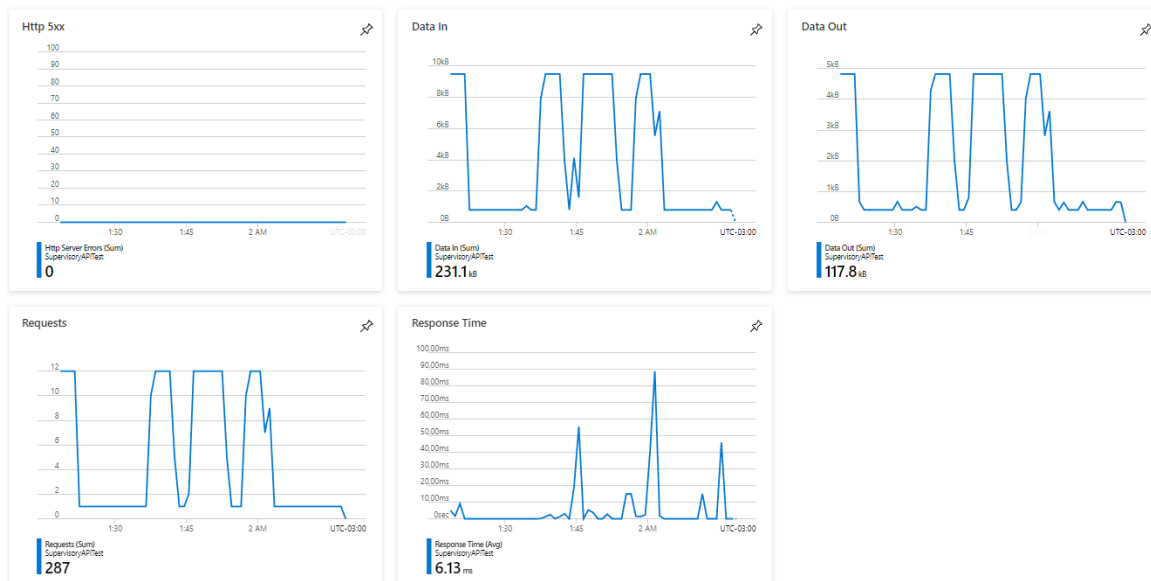
A Figura 15, assim com o restante dos dados adquiridos durante o experimento, permitiu a descoberta dos padrões e limites de temperatura e saída de tensão para os relés. Com base na convergência aproximada da média da saída dos relés,  $\bar{U}$ , de 5 V, arbitrariamente foi estipulada uma margem de  $\pm 0,2$  V para a identificação de uma mudança por parte da tensão de ativação das lâmpadas. Já para a média dos intervalos de tempo em que a curva de temperatura fica acima da referência,  $\bar{\Delta t}$ , a média, mesmo ficando em torno de 2 minutos e 38 s minutos, apresentou maior propensão à ocorrência de desvios mais significativos de tempo, como um intervalo acima de 2 minutos e 55 s. Dessa forma, a margem arbitrária utilizada foi de 2m38s  $\pm$  40 s.

A Figura 18 mostra alteração em ambos os parâmetros  $\bar{U}$  e  $\bar{\Delta t}$ , caracterizando uma diferença entre os valores anteriores e os novos de magnitude suficiente para o usuário ser alertado. Dessa forma, o aviso “Mudança na Planta” foi disparado e só desaparecerá uma vez que o usuário mudar os ganhos de controle. O alarme só poderá ser disparado novamente depois de um dia transcorrer, que é o marco para o sistema começar a procurar por mudanças novamente.

## 4.7 CHAMADAS DA API NA NUVEM

A Figura 19 mostra os gráficos de monitoramento da plataforma Azure, a qual serve de servidor para a API. Os dois gráficos mais relevantes são o “*Requests*” e “*Response Time*”, pois tratam do número de requisições feitas e o tempo de resposta para elas, respectivamente. O número de requisições feitas comprova que o aplicativo de fato recebeu chamadas, enquanto o tempo de resposta indica a velocidade em que a *consegue responder*. É possível notar que, no gráfico “*Requests*” (requisições), existe oscilações no número de chamadas ao longo de uma hora, não tendo um comportamento estável ao longo do processo, com os picos sendo de 12 chamadas e os mínimos de duas. O gráfico “*Response Time*” (tempo de resposta) apresenta variações no tempo de resposta que a API tem para as chamadas, apresentando um máximo de 88 ms e um mínimo de 1,25 ms.

**Figura 19:** Tela de monitoramento do aplicativo da API publicada na plataforma Azure



Fonte: Microsoft (2023)

Um fato importante é que os valores apresentados pelos gráficos de monitoramento do Azure são uma estimativa, pois o plano de uso não é o com maior capacidade de monitoramento. Além disso, os tempos não dependem somente do Azure, mas da combinação dele com a plataforma e servidor do banco de dados MongoDB, que também usa a versão gratuita. Os tempos experienciados pelo Arduino e pelo supervisor variaram de 200 ms até 4 s. Por causa deste fato, um tempo de *timeout* foi estabelecido como 1 segundo e o número de tentativas máximo para conexão com a *API* é de 3. Apesar da maioria das conexões serem próximas de 200 ms, no pior caso, onde o tempo de *timeout* é atingido todas as três vezes, a temporização do *loop* feito pelo Arduino acaba não sendo de 5 s, mas sim de  $5 + 1 * 3 = 8$  s, o que muda o tempo de amostragem efetivo do controlador.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho teve como objetivo a capacidade de fornecer para o usuário um sistema com arquitetura *IoT* que permitisse controlar a temperatura de um sistema de compostagem doméstico com um controlador PID. Além disso, e mais importante, é a capacidade do operador de visualizar o estado da planta controlada remotamente e de alterar os ganhos do controlador pela mesma interface a qualquer momento. Dadas estas premissas, o trabalho desenvolvido, atingiu os requerimentos, mas com muito espaço para melhorar.

Apesar do trabalho ter realizado o que ele se propôs, existem muitas variáveis que diminuíram sua performance e, discutivelmente, podem ter impedido certas ações de serem executadas com sucesso. Um exemplo disso foi a limitação física do controlador, pois, mesmo tendo um controle digital PID implementado na placa Arduino, o atuador responsável pelo acionamento do sistema de aquecimento é um relé. Esse fato diminui a eficácia do controlador significativamente, podendo ser um quesito importante para reavaliação futuramente. Outro aspecto relevante para a performance da solução do problema de comunicação pela *internet* é o servidor Azure e MongoDB utilizados. Estes servidores foram escolhidos por familiaridade do autor com sua utilização e peculiaridades, mas a licença utilizada foi uma gratuita, o que limita com grande impacto na performance de recepção e resposta, além do monitoramento de ambas aplicações na nuvem.

O sensor escolhido para o trabalho de medição de temperatura, DHT22, se mostrou um desempenho pouco eficiente quando inserido na terra. Suas medições variaram bastante na precisão, apresentando saltos de erro de até 6 °C. Um erro grande assim pode colocar em risco a confiabilidade do trabalho e, por isso, todos os valores usados nas comparações foram os valores de média móvel. A inconsistência na resolução do sensor também foi baixa, obrigando o autor a reduzir esta resolução de medição para 1 °C em vez de 0,2 °C. Considerando os resultados deste sensor, é notória a necessidade de um sensor diferente esse tipo de aplicações.

O supervisor atingiu os objetivos que se propôs, mostrando os dados e notificando o usuário quando necessário. Entretanto, sua estilização é básica, deixando bastante espaço para exploração e inovação. A metodologia utilizada para reconhecimento das alterações na planta também é simples e específica para este tipo de sistema térmico, sendo um dos métodos específico para o uso de relés. Essa metodologia necessita de um grande número de testes para identificar as margens de tolerância a serem utilizadas no fator de decisão, tendo o requerimento então de uma instalação mais trabalhosa e que demanda tempo para ser efetivada.

Considerando que nenhuma das áreas é o foco específico do trabalho e sim o funcionamento de todas em conjunto, o trabalho atinge as expectativas. Entretanto, o projeto possibilita que vários aspectos possam ainda ser explorados, em termos de otimização e confiança em seu desempenho. É necessário maior estudo em relação ao acionamento, aos atuadores e ao método de identificação de mudanças, maior investimento

nos servidores e sensores utilizados e maior tempo para o desenvolvimento da interface gráfica do supervisor.

## REFERÊNCIAS

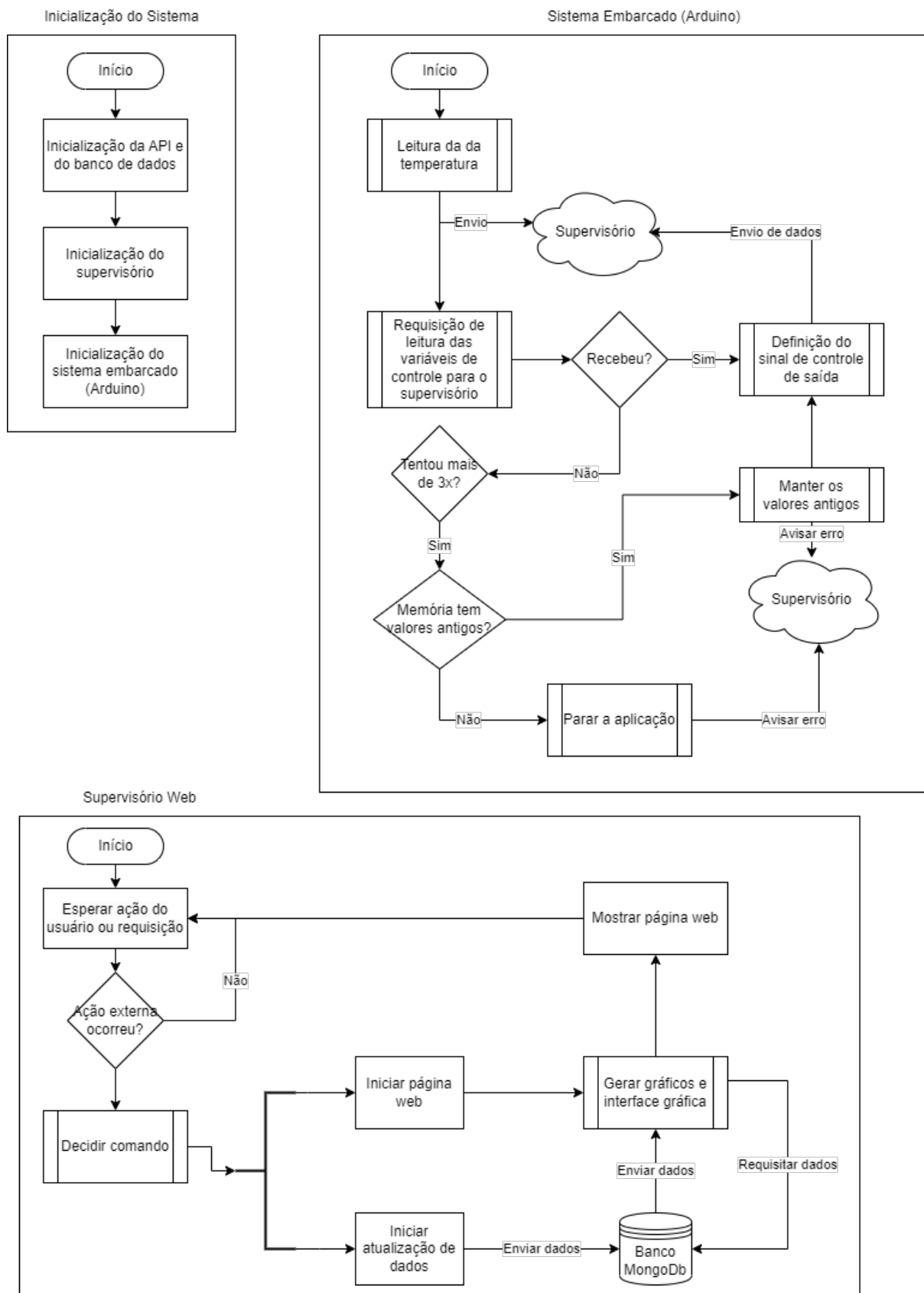
- ARDUINO. *Arduino® Mega Rev3*. [S.l.], Fevereiro 2023. Disponível em: <<https://docs.arduino.cc/static/72be0dec11bfaddb42da1c09e0d02ac5/A000067-datasheet.pdf>>.
- ASTROM, K.; WITTENMARK, B. *Computer-Controlled Systems - Theory and Design*. Edição: Thomas Kailath. 3. ed. [S.l.]: Prentice Hall, 1997.
- BAZANELLA, A.; SILVA JR, J. M. G. DA. *Sistemas de Controle princípios e métodos de projeto*. Porto Alegre: UFRGS Editora, 2005. P. 306. ISBN 978-85-7025-849-6.
- BERTICELLI, R. et al. COMPOSTAGEM COMO ALTERNATIVA DE BIORREMEDIÇÃO DE ÁREAS CONTAMINADAS. *CIATEC*, v. 8, p. 27, jan. 2016.
- COMLURB. *Manual de Compostagem Doméstica*. [S.l.], 2021. Disponível em: <<http://www.rio.rj.gov.br/documents/91370/1017211/compostagem-manual2021.pdf>>.
- DAVID, G.; BRIAN, T. *HTTP: The Definitive Edition*. Edição: Linda Mui. 1st. Gravenstein Highway North, Sebastopol: O'Reilly Media, 2002.
- HERBETS, R. A. et al. Compostagem de Resíduos Sólidos Orgânicos: Aspectos Biotecnológicos, p. 9, jan. 2005. Disponível em: <[https://www.researchgate.net/profile/Luiz-Claudio-Miletti/publication/274564974\\_Compostagem\\_de\\_Residuos\\_solidos\\_organicos\\_aspectos\\_biotecnologicos/links/552273f30cf2a2d9e1454bb6/Compostagem-de-Residuos-solidos-organicos-aspectos-biotecnologicos.pdf](https://www.researchgate.net/profile/Luiz-Claudio-Miletti/publication/274564974_Compostagem_de_Residuos_solidos_organicos_aspectos_biotecnologicos/links/552273f30cf2a2d9e1454bb6/Compostagem-de-Residuos-solidos-organicos-aspectos-biotecnologicos.pdf)>.
- LEA, P. *IoT and Edge Computing for Architects*. Edição: Janice Gonsalves. 2. ed. 35 Livery Street, Birmingham, Inglaterra: Packt Publishing Ltd., 2020.
- LIU, T. *Digital-output relative humidity temperature sensor/module DHT22 (DHT22 also named as AM2302)*. [S.l.: s.n.], 2023. Disponível em: <<https://pdf1.alldatasheet.com/datasheet-pdf/view/1132459/ETC2/DHT22.html>>.
- MICROSOFT. *Key Metrics*. [S.l.: s.n.], mar. 2023. Disponível em: <<https://learn.microsoft.com/en-us/azure/azure-monitor/essentials/data-platform-metrics>>.
- ORACLE. *What is IoT?* [S.l.: s.n.], 2023. Disponível em: <<https://www.oracle.com/internet-of-things/what-is-iot/>>.
- SEQUI, P. The Role of Composting in Sustainable Agriculture. In: *THE Science of Composting*. [S.l.]: Springer Netherlands, 1996. P. 23-29. DOI: 10.1007/978-94-009-1569-5\_3.

- SMITH, S.; JASIM, S. Small-scale home composting of biodegradable household waste: overview of key results from a 3-year research programme in West London. *Waste Management & Research: The Journal for a Sustainable Circular Economy*, SAGE Publications, v. 27, n. 10, p. 941–950, 1 jun. 2009. DOI: 10.1177/0734242x09103828.
- XU, Y. et al. Comparative study on thermal properties of undisturbed and compacted lateritic soils subjected to drying and wetting. *Engineering Geology*, Elsevier BV, v. 277, p. 105800, nov. 2020. DOI: 10.1016/j.enggeo.2020.105800.



# Anexos

**Anexo A. 1:** Fluxograma mais profundo do funcionamento do sistema



Fonte: O autor