

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

**BRUNO GABARDO PALLÚ
KALONI MACIEL DE MORAES
TIAGO ALEXANDRE DONATO**

**SISTEMA DE CONTROLE DE PROCESSOS PARA OTIMIZAÇÃO DE
CARACTERÍSTICAS FISIOLÓGICAS DE PLANTAS DE PEQUENO PORTE**

**CURITIBA
2022**

**BRUNO GABARDO PALLÚ
KALONI MACIEL DE MORAES
TIAGO ALEXANDRE DONATO**

**SISTEMA DE CONTROLE DE PROCESSOS PARA OTIMIZAÇÃO DE
CARACTERÍSTICAS FISIOLÓGICAS DE PLANTAS DE PEQUENO PORTE**

**PROCESS CONTROL SYSTEM FOR OPTIMIZING THE PHYSIOLOGICAL
CHARACTERISTICS OF SMALL PLANTS**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Controle e Automação do Curso de Engenharia de Controle e Automação da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Alexandre José Tuoto
Silveira Mello

Coorientador: Prof. João Lauro de Almeida
Silva

CURITIBA

2021



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**BRUNO GABARDO PALLÚ
KALONI MACIEL DE MORAES
TIAGO ALEXANDRE DONATO**

**SISTEMA DE CONTROLE DE PROCESSOS PARA OTIMIZAÇÃO DE
CARACTERÍSTICAS FISIOLÓGICAS DE PLANTAS DE PEQUENO PORTE**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Engenharia de Controle e
Automação do Curso de Engenharia de Controle
e Automação da Universidade Tecnológica
Federal do Paraná (UTFPR).

Data de aprovação: 24 de Novembro de 2022

Prof. Alexandre José Tuoto Silveira Mello
Doutor
Universidade Tecnológica Federal do Paraná

Prof. João Lauro de Almeida Silva
Professor
Centro Universitário Campos de Andrade

Prof. Adalberto dos Santos
Mestre
Universidade Federal do Paraná

**CURITIBA
2021**

Dedicamos este trabalho aos que caíram durante
esta caminhada por muitos motivos,
as pessoas que estão seguindo este caminho
recompensador e aqueles que virão até nós
e sempre encontrarão um porto seguro para o
aprendizado.

AGRADECIMENTOS

Aos nossos pais, familiares e amigos que nos auxiliaram ao longo da graduação. Agradecemos pela ajuda de Breno Gabardo Pallú, em todo o auxílio na montagem mecânica do dispositivo e do Prof. João Lauro de Almeida Silva no suporte sobre cultivo de plantas, e também do Prof. Dr. Alexandre José Tuoto Silveira Mello por aceitar ser nosso orientador e confiar em nosso projeto.

Também gostaríamos de agradecer a todos que não acreditaram em nossa capacidade, e que o descrédito dos mesmos formaram os degraus da nossa conquista.

Não importa o quanto você vá devagar desde
que não pare.
(NATÁRIO, Maria Celeste, 2019)

RESUMO

Devido a elevação da qualidade exigida no cultivo de plantas, tanto para alimentação como para ornamentação e outros fins, este projeto foca o nicho de cultivo de precisão, visando a melhoria quantitativa de atributos fisiológicos da planta com um controle fino dos insumos cedidos à mesma. Com a aplicação de ferramentas de controle de variáveis físicas como temperatura, pH e umidade, foi desenvolvida uma estufa que cultiva a planta de maneira inteligente cedendo os insumos necessários para a planta se desenvolver mais efetivamente. A programação para tal dispositivo foi desenvolvida na plataforma de microcontrolador de código aberto Arduino utilizando código C no desenvolvimento da lógica do sistema. O microcontrolador utiliza sensores periféricos para identificar o valor das variáveis em tempo real e acionar atuadores que regularão instantaneamente o ambiente interno da estufa e a quantidade de insumos cedidos à planta no decorrer do seu desenvolvimento.

Palavras-chave: Cultivo. Melhoria. Controle

ABSTRACT

Due to the increase in the quality required in the cultivation of plants, whether for food, ornamentation or other purposes, this project focuses on the niche of Precision Cultivation, aiming at the quantitative improvement of the plant's physiological attributes with fine control of the inputs given to it. Using control tools to measure physical variables such as temperature, pH and humidity, a greenhouse was developed to cultivate the plants in an intelligent way, providing the necessary inputs for the plant to develop more effectively. The programming for such device was developed on the Arduino open-source microcontroller platform and using C code to develop the system logic. The microcontroller uses peripheral sensors to identify the value of variables in real-time and activate actuators that will instantly regulate the internal environment of the greenhouse and the amount of inputs given to the plant during its development.

Keywords: Cultivation. Improvement. Control

LISTA DE ILUSTRAÇÕES

Figura 1 – Sistema de controle em malha fechada	15
Figura 2 – Cronograma	18
Figura 3 – Exemplos de plantas do grupo Angiospermas	19
Figura 4 – Exemplos de plantas do grupo Gimnospermas	20
Figura 5 – Exemplo de plantio de cebolinha	21
Figura 6 – Mudanças de cebolinha em bandejas de isopor	22
Figura 7 – Cebolinha germinando	23
Figura 8 – Informações gerais sobre o cultivo de hortaliças	25
Figura 9 – Amostragem de Luz Par	26
Figura 10 – Fotoperíodo das plantas	27
Figura 11 – Lesões causadas na cebola pelo fungo	28
Figura 12 – Sintomas de clorose nas plantas	28
Figura 13 – Exemplo de tela de sistema supervisorio	30
Figura 14 – Exemplo de banco de dados relacional	31
Figura 15 – Exemplo de sistema de controle de malha fechada	32
Figura 16 – Temperatura controlada por liga-desliga com histerese	33
Figura 17 – Especificações da resposta ao degrau	34
Figura 18 – Exemplo de arquitetura de microcontrolador	37
Figura 19 – Arduino MEGA	38
Figura 20 – Especificações técnicas Arduino MEGA	39
Figura 21 – Memória Flash	40
Figura 22 – Bluetooth	41
Figura 23 – Nobreak	42
Figura 24 – Tipos de Nobreak	42
Figura 25 – Permissividade elétrica dos componentes do solo	44
Figura 26 – Sensor capacitivo de umidade do solo	45
Figura 27 – Sensor DS18B20	46
Figura 28 – Escala de ph	47
Figura 29 – Sensor de ph para soluções	48
Figura 30 – Pinagem do módulo do sensor de ph BNC PH4502C	49
Figura 31 – Diagrama do processo de cultivo da planta	53
Figura 32 – Diagrama do sistema eletrônico	54
Figura 33 – Desenho do esquema elétrico de acionamento da placa de 12V	55
Figura 34 – Esquema elétrico fonte 127V/12V	55
Figura 35 – Conexão do sensor de umidade do solo no arduino	56
Figura 36 – Conexão do sensor DS18B20 com o Arduino(2021)	57
Figura 37 – Esquema elétrico de conexão do kit de sensoriamento de ph	58

Figura 38 – Bomba Hidráulica na estufa	59
Figura 39 – Servomotor que desloca o sensor de pH para o recipiente de água.	60
Figura 40 – Misturador	61
Figura 41 – Recipiente de vinagre	61
Figura 42 – Vista frontal do <i>cooler</i> e da porta	62
Figura 43 – Vista traseira do <i>cooler</i> e da porta	63
Figura 44 – Lâmpada <i>full</i> espectro	64
Figura 45 – Serial bluetooth	65
Figura 46 – Módulo Bluetooth HC-06	66
Figura 47 – Módulo Cartão SD	67
Figura 48 – Gráfico com todos os acionamentos completos no dia específico	69
Figura 49 – Amostra de como os dados são registrados no cartão SD	70
Figura 50 – Tela de seleção de planta	71
Figura 51 – Tela de Amostra das variáveis da planta	71
Figura 52 – Tela de solicitação de adubo	72
Figura 53 – Tela de definição da área de cultivo	72
Figura 54 – Datalog do sensor e do sistema de controle de pH	74
Figura 55 – Datalog do sensor e do sistema de controle de temperatura	76
Figura 56 – Datalog do atuador de luminosidade	77
Figura 57 – Datalog do sensor e do sistema de controle de umidade	79
Figura 58 – Projeção de crescimento da planta	80

LISTA DE TABELAS

Tabela 1 – Classificação científica da cebolinha	21
Tabela 2 – Permissividade elétrica relativa dos elementos do solo	44
Tabela 3 – Saída digital do sensor DS18B20	46

LISTA DE ABREVIATURAS E SIGLAS

CI	Integrated Circuit
EEPROM	Electrically-Erasable Programmable Read-Only Memory
EMBRAPA	Empresa Brasileira de Pesquisa Agropecuária
GND	Ground (Terra)
GPS	Global Position System (USA)
ICSP	In Circuit Serial Programming (Programação Serial em Circuito)
IHM	Interface Homem-Máquina
INPUT	Entrada
LED	Light Emitting Diode
NOR	Novel Object Recognition - Reconhecimento de Novo Objeto
NPK	nitrogênio (N), fósforo (P) e potássio (K)
NTC	Negative Temperature Coefficient
OUTPUT	Saída
PID	Parameter ID
PV	Process Variable (Variável de processo)
PWM	Pulse Width Modulation
RAM	Random Access Memory - Memória de Acesso Aleatorio
SD	Secure Digital Card
SRAM	Static Random Access Memory
TCC	Trabalho de Conclusão de Curso
UNESP	Universidade Estadual Paulista
USB	Universal Serial Bus
VCC	Tensão Corrente Contínua
VDC	Voltage Direct Current
°C	Graus Célsius

SUMÁRIO

1 INTRODUÇÃO	14
1.1 Delimitação do tema	14
1.2 Problemas e premissas	15
1.3 Objetivos	16
1.3.1 Objetivo Geral	16
1.3.2 Objetivos Específicos	16
1.4 Justificativa	17
1.5 Metodologia da pesquisa	17
1.6 Estrutura do trabalho	18
1.7 Cronograma	18
2 REFERENCIAL TEÓRICO	19
2.1 Grupos de plantas	19
2.2 Escolha das plantas	20
2.3 Métodos de cultivo	22
2.3.1 Adubação	23
2.3.2 Área de Cultivo	24
2.3.3 Ambiente controlado	25
2.3.4 Influência da luz no desenvolvimento das plantas	26
2.3.5 Dosagem dos Insumos	27
2.4 Sistema supervisorio	29
2.5 Banco de dados	30
2.6 Controle	31
2.7 Métodos de controle	32
2.7.1 Bang-Bang	32
2.7.2 PID	33
2.8 Eletrônica	35
2.9 Microcontrolador	36
2.9.1 Arduino Mega	37
2.9.2 Memória Flash	39
2.9.3 Bluetooth	40
2.9.4 Nobreak	41
2.10 Sensores	43
2.10.1 Sensor de umidade	43
2.10.2 Sensor de temperatura	45
2.10.3 Sensor de pH	47
2.11 Atuadores	49
2.11.1 Bomba hidráulico	49

2.11.2 Sistema de controle de temperatura	50
2.11.2.1 Cooler de ventilação	50
2.11.2.2 Sistema de aquecimento	50
3 DESENVOLVIMENTO DO PROJETO	51
3.1 Modelagem	52
3.2 Ligação dos dispositivos periféricos	54
3.2.1 Esquema de ligação dos sensores	56
3.2.1.1 Sensor de umidade	56
3.2.1.2 Sensor de temperatura	57
3.2.1.3 Sensor de pH	58
3.2.2 Atuadores do Dispositivo	59
3.2.2.1 Irrigação	59
3.2.2.2 Atuadores referentes ao sensor de pH	59
3.2.2.3 Cooler	61
3.2.2.4 Lâmpada full espectro	63
3.2.3 Interfaces Homem Máquina (IHM's)	64
3.2.3.1 Aplicativo de comunicação serial bluetooth	64
3.2.3.1.1 <i>Módulo Bluetooth HC-06</i>	65
3.2.3.2 Datalogger	66
3.2.3.2.1 <i>Módulo Cartão SD</i>	66
3.2.4 Dados das plantas armazenadas no dispositivo	67
4 APLICAÇÃO DO PROJETO	69
4.1 Inicialização do procedimento de cultivo	70
4.2 Controle de pH	73
4.3 Controle de temperatura	74
4.4 Controle de luminosidade	76
4.5 Controle de umidade	78
4.6 Resultado do desenvolvimento da planta	79
5 CONCLUSÕES E CONSIDERAÇÕES FINAIS	81
5.1 Microcontrolador	81
5.2 Estrutura mecânica	81
5.3 Sugestões para trabalhos futuros	83
REFERÊNCIAS	84
ANEXOS	86
Anexo 1 - Código do arduino em C++	87

1 INTRODUÇÃO

A automação agregada a técnicas de controle exerce forte influência em diversas áreas de aplicação, ajudando na pesquisa e no desenvolvimento de novos métodos e dispositivos.(NATÁRIO, 2019)

Com o crescimento exponencial do agronegócio, a tecnologia também agregou valor a este campo de produção, com novos dispositivos e equipamentos para trabalhos antes executados por pessoas, aumentando em muito o desempenho e o custo benefício nessa área de trabalho.

Como exemplo, para trazer mais praticidade aos produtores rurais, muitas empresas têm buscado investir em maquinários autônomos, como colheitadeiras e tratores, com seus próprios sensores, comandados a distância por smartphones ou computadores, guiados por GPS, reduzindo custos e aumentando a produtividade, pois melhora a qualidade e otimiza os processos (SANTOS, 2020).

Na agricultura, para a planta se desenvolver em sua melhor forma, ela precisa de cuidados específicos, assim como água e luminosidade na quantidade correta e solo contendo os nutrientes necessários à planta.

O projeto deste TCC foi desenvolvido abrangendo as necessidades básicas de plantas de pequeno porte a fim de estimular ao máximo suas características fisiológicas de interesse, sejam plantas decorativas ou destinadas à alimentação de humanos, ou animais.

1.1 Delimitação do tema

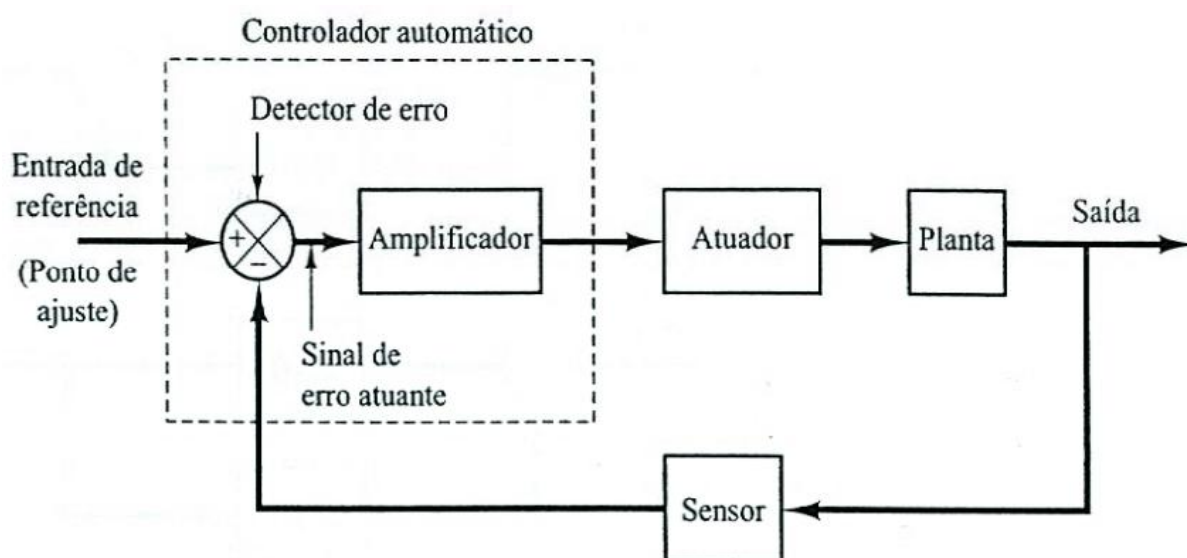
O projeto consiste em um sistema supervisorio, baseado em sensores, que aplica funções vigentes em um banco de dados específico para cada planta, com características como controle de umidade, nutrientes, temperatura e outros. Contemplando um sistema de controle onde o mesmo irá verificar as necessidades das plantas em tempo real e executar de forma automática as correções das variáveis necessárias para a planta atingir um desenvolvimento satisfatório. Utilizando um sistema de controle de malha aberta e um banco de dados para cada planta, o projeto pode ser direcionado para inúmeros tipos de plantas visando cultivo, produção ou pesquisa e desenvolvimento.

A irrigação é uma prática indispensável para que o cultivo atinja altos níveis de produtividade e qualidade. A água faz parte de praticamente todos os processos de bioquímicos e fisiológicos das plantas, na divisão celular; a falta de água pode prejudicar essa etapa de desenvolvimento, acarretando diminuição da célula levando redução número de folhas e o tamanho dos frutos, caule da planta mais espesso e seco e raízes menores. Em contrapartida, excedendo a irrigação de água, a absorção de oxigênio da planta é prejudicada, assim como a capacidade de absorver nutrientes da terra e de levar água para as folhas superiores, provocando apodrecimento e até mesmo a morte, este excesso de

umidade também acarreta proliferação de fungos (COELHO; COELHO FILHO; OLIVEIRA, 2005).

A aplicação de controle de realimentação nestes processos visa a controlar a quantidade de água da planta de maneira correta sem excesso ou perda. Logo, o controle é fundamental para todo o processo. O controle com realimentação, representado na Figura 1, refere-se a uma operação que, na presença de distúrbios, tende a diminuir a diferença entre a saída de um sistema e alguma entrada de referência, e atua com base nessa diferença (OGATA, 2010).

Figura 1 – Sistema de controle em malha fechada



OGATA (2010, p. 18)

1.2 Problemas e premissas

Quando se trata de cultivo de plantas há muitas variações quanto às necessidades de cada planta. Por exemplo, no plantio de morangos precisa-se controlar o ambiente, onde as principais características são: controle de irrigação em função da umidade do solo, pH da água a ser irrigada em função do pH do solo e temperatura.

Outro exemplo a citar seria o cultivo de girassóis, plantas ornamentais e fonte de grãos para diversas utilidades, necessitam de um controle do pH e da umidade do solo.

Será desenvolvido um sistema supervisorio em que o mesmo colhe dados da planta e que por meio de sensores e um sistema de controle alimentado com dados obtidos de um banco de dados pré-programado, determina as ações necessárias para o cultivo da planta.

No entanto, que tal fim seja alcançado é necessário responder às perguntas, tais como: quais métodos serão aplicados para irrigação? Quais sensores utilizaremos para medir o pH e outras características do ambiente, e quais características precisam ser consideradas? Qual microcontrolador será adotado? E qual das opções de interface ao

usuário baseada nesse microcontrolador será escolhida?

Tendo essas questões em vista, pretende-se desenvolver um sistema supervisório baseado em microcontroladores para tornar automático este processo de cultivo e assim, melhor o desenvolvimento das plantas escolhidas para cultivo.

1.3 Objetivos

Este TCC visa a análise de aplicação de diferentes dispositivos do segmento da eletrônica, da automação e do controle de processos. Aplicando tais processos em conjunto, será elaborado um processo automático de cultivo de uma planta de pequeno porte, a fim de auxiliá-la a se desenvolver de maneira otimizada.

1.3.1 Objetivo Geral

Desenvolver um sistema supervisório baseado em microcontroladores que possa automatizar atividades e necessidades de uma planta, a fim de que ela se desenvolva com excelência em desempenho de produção e/ou crescimento.

1.3.2 Objetivos Específicos

As etapas mais importantes deste projeto são:

- Desenvolver um sistema supervisório baseado em microcontroladores;
- Desenvolver um código em linguagem C para atuar no microcontrolador;
- Criar e manter um banco de dados com as necessidades fundamentais para o desenvolvimento das plantas;
- Definir quais sensores e atuadores serão utilizados no projeto;
- Aplicar, testar e ajustar um sistema de controle para manipular os atuadores baseados nas medições dos sensores;
- Definir dimensão do projeto, para qual escala (canteiro/plantação);
- Escolher uma interface com o usuário baseada no microcontrolador;
- Auxiliar às pessoas que farão cultivo de plantas de pequeno porte, para evitar a necessidade de checar pessoalmente as características do solo e aspectos e variações da planta.

1.4 Justificativa

Para o desenvolvimento das plantas de maneira maximizada serão utilizados preceitos de controle e automação, com o sistema supervisor pretende-se que as plantas cresçam com valores de cedidos de pH, umidade e outros insumos próximo dos ideais, isso contribuirá para o melhor desenvolvimento possível.

Devido à grande dificuldade das pessoas cuidarem de plantas, seja por falta de informações ou até mesmo por falta de tempo, este projeto contemplará um banco de dados contendo as informações necessárias para o pleno desenvolvimento das plantas, que pode ser enriquecido conforme necessidade de cultivo de novas espécies de plantas.

Exemplificando uma tecnologia em que o dispositivo possa ser aplicado são as fazendas verticais, estruturas produtivas localizadas em centros urbanos com grande aplicação de tecnologia no aperfeiçoamento do desempenho genético da planta, “encurtamento do ciclo produtivo das culturas agrícolas, redução da emissão de CO através do achatamento das cadeias produtivas, entre outros benefícios ao bem-estar das gerações futuras” (LUCENA *et al.*, 2014).

1.5 Metodologia da pesquisa

Para desenvolver este projeto, foram analisadas as maneiras mais acessíveis de suprir cada necessidade fisiológica das plantas, por um sistema eletrônico baseado em microcontrolador. Ao dar início a pesquisas bibliográficas, buscaram-se bibliografias relacionadas a microcontroladores, sistemas de controle, componentes eletrônicos (sensores e atuadores), banco de dados, programação em linguagem C, espécies de plantas bem como seu cultivo.

Na etapa seguinte foi atribuída a dimensão do projeto considerando em qual escala seria desenvolvido, baseado em manual de métodos de cultivo.

Para supervisionar o sistema, foi definido qual microcontrolador seria mais adequado para o projeto, considerando a quantidade de portas disponíveis para comunicação e também a acessibilidade de bibliotecas e funções do dispositivo.

Ao analisar as necessidades das plantas, foram determinados quais sensores e atuadores que devem compor o sistema. E a partir destes dispositivos, foram elaborados circuitos eletrônicos, quando necessário, para estabelecer a comunicação adequada com o microcontrolador, e também possibilitar acionamentos de potência.

Para efetuar o sistema supervisor, foram desenvolvidas planilhas em Excel, onde as informações coletadas geram parâmetros de desenvolvimento da planta e de acionamento dos sensores e atuadores.

1.6 Estrutura do trabalho

No primeiro capítulo é apresentado o tema deste trabalho e alguns problemas previstos. Também explicitado seu objetivo, a descrição das etapas do desenvolvimento do projeto e o cronograma.

No segundo capítulo são mostrados os resultados das pesquisas bibliográficas referentes aos tipos e necessidades de plantas, técnicas de cultivo, sistema supervisorio, métodos de controle, circuitos eletrônicos, microcontrolador, sensores e atuadores.

O terceiro capítulo compõem a modelagem do sistema, isto inclui o detalhamento dos circuitos eletrônicos e do funcionamento de cada sensor e atuador empregados no mesmo.

O quarto capítulo compreende a funcionalidade do dispositivo em operação, de modo detalhado a análise de trabalho de cada atuador correspondente à medida detectada pelo sensor, ao fim deste, uma análise sobre o resultado de todo o processo no desenvolvimento da planta.

Por fim, o quinto capítulo abrange as demais características do dispositivo operando, tais como atuação e problemas e dificuldades no decorrer do processo de cultivo.

1.7 Cronograma

Figura 2 – Cronograma

Ano	2021												2022											
	Mês	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
Proposta TCC	X	X	X	X																				
Pesquisas Bibliográficas			X	X	X	X	X								X	X	X	X	X					
Definir Dimensões do Projeto				X	X	X							X	X	X	X	X							
Definir e Programar Microcontrolador						X	X	X																
Determinar Sensores e Atuadores						X	X	X																
Desenvolvimento do Circuito Eletrônico							X	X	X															
Pesquisas de Cabos para Instalações							X	X	X															
Elaboração do código em C									X	X	X						X	X	X	X	X	X	X	X
Implementação do Sistema de Controle									X	X	X													
Desenvolvimento da Parte Elétrica										X	X	X	X											
Medições e Reajustes										X	X	X	X	X										
Pesquisa de viabilidade Econômica											X			X	X	X	X						X	
Desenvolvimento do Material Escrito	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Correções Supervisionadas pelo Orientador	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X					
Análise da Banca											X							X						X

Fonte: Autoria própria

2 REFERENCIAL TEÓRICO

Abordando a fundamentação teórica do projeto, este capítulo servirá de base para o desenvolvimento do projeto proposto, dando sustentação argumentativa sobre seu desenvolvimento.

2.1 Grupos de plantas

A Botânica é a área da biologia que estuda o Reino Plantae, onde estão incluídas todas as plantas. O Reino Plantae é extremamente variado, com espécies simples que não apresentam folhas, caule e raízes verdadeiras até espécies com frutos carnosos e flores deslumbrantes. Tal Reino é dividido em quatro grupos: briófitas, pteridófitas, gimnospermas e angiospermas (FILGUEIRA, F.A.R., 2008).

Para armazenamento no banco de dados desta pesquisa, será utilizado somente plantas de pequeno porte para cultivo e desenvolvimento de mudas, com isso o projeto irá abranger apenas: angiospermas e gimnospermas.

O grupo das angiospermas é formado por plantas vasculares com a produção de flores e sementes onde gera um fruto, como representado na Figura 2. Os animais são os principais agentes polinizadores das angiospermas, algo que explica a grande diversidade de cores (FILGUEIRA, F.A.R., 2008).

O grupo das angiospermas contempla em torno de 90% de todas as espécies existentes. Todas as plantas que possuem flores e frutos são angiospermas. Dentre os exemplos de plantas desse grupo estão: a cebolinha, a cebola, o morango, o centeio, o feijoeiro, a grama, os lírios, as orquídeas.

Figura 3 – Exemplos de plantas do grupo Angiospermas



Monocotiledôneas e Dicotiledôneas (2013)

O segundo é conhecido como Gimnosperma, mostrado na Figura 3, é um grupo de

plantas terrestres que possuem semente, porém não geram o fruto onde as sementes se encontram totalmente “nuas”. Esse grupo é encontrado na maioria das vezes em florestas de clima temperado. As Gimnospermas não possuem flores, porém, em alguns casos tem estróbilos, como a pinha.

Figura 4 – Exemplos de plantas do grupo Gimnospermas



<https://cursosnocd.com.br/biologia/reino-metafita-classificacao-geral-criptogamas-briofitas-pteridofitas-fanerogamas-gimnospermas-e-angiospermas-biologia.htm> (2013)

2.2 Escolha das plantas

Para fins didáticos, foi tomada a decisão de utilizar neste experimento plantas com pouco tempo de desenvolvimento e com a possibilidade de cultivo em recipientes ou vasos, para manter mais precisamente o controle do desenvolvimento das amostras de plantas utilizadas. A planta escolhida para a amostragem foi a cebolinha, pois será mais visível seu desenvolvimento, para fins de comparação pode-se analisar a altura das folhas ou até o peso da mesma após a colheita.

A cebolinha-comum (*Allium schoenoprasum*), originária da Europa continental, é considerada perene, apresenta folhas cilíndricas e fistulosas, com 0,30 a 0,50 m de altura, produz pequeno bulbo cônico, envolvido por uma película rósea, com perfilhamento e formação de touceira, ela faz parte do grupo de angiospermas e sua classificação científica está listada na Tabela 2.

A cebolinha é um dos temperos mais utilizados na cozinha brasileira. O consumo da cebolinha proporciona vários benefícios para a saúde, “possuindo vitamina A e C, e

Tabela 1 – Classificação científica da cebolinha

Espécie	Allium schoenoprasum
Família	Amaryllidaceae
Ordem	Asparagales
Sub-classe	Liliatae (Monocotyledoneae)
Classe	Liliidae
Subdivisão	Magnoliophytina (Angiospermae)
Divisão	Spermatophyta
Tipo Fisionômico	Geófito
Nome Comum	Ceboletas-de-França; Cebolinha; Cebolinha-galega; Cebolinha-miúda

Adaptado de UTAD

minerais como cálcio, fósforo e niacina” (RAMOS NETO, 2021).

As plantas serão adicionadas e suas variáveis de desenvolvimento configuradas na base de dados, o sistema será configurado para cuidar de uma espécie de planta por vez, possibilitando substituir quando necessário, as configurações do sistema para cada tipo de planta.

Figura 5 – Exemplo de plantio de cebolinha

CPT(2020)

Após 60 dias do plantio feito diretamente no canteiro, como mostrado na Figura 4, inicia-se a colheita com as plantas que estiverem bem desenvolvidas e com uma altura aproximadamente de 30 cm. Neste caso, é mais interessante a parte folhosa do que o bulbo da cebolinha. A colheita pode ser feita arrancando-se todas as plantas, ou cortando cada planta na sua parte branca (colmo) um pouco acima do solo. Neste último método haverá novo rebrotamento da planta (soca) sem que seja necessário novo plantio (SOBREIRA FILHO, 2012).

2.3 Métodos de cultivo

Para o cultivo da cebolinha, precisam-se tomar certos cuidados durante o plantio e durante o desenvolvimento desta hortaliça. Os pés de cebolinha podem ser plantados em bandejas de isopor, conforme a Figura 5, até atingirem tamanho de mudas, e em seguida serem plantadas em canteiros, como mostrado na Figura 4, ou também podem ser plantados diretamente no solo, conforme a Figura 6.

O solo precisa ser fértil, com pH regulado para receber a cebolinha, é preciso que o local receba luz solar, 18 horas por dia na fase de germinação e 12 horas por dia na fase de floração (“FOTOPERIODISMO” . . . , 2008-2022). Ao distribuir as mudas no solo, dedicar um espaço para cada muda, deixando 15 cm de profundidade e 12 cm de diâmetro entre as mudas.

Figura 6 – Mudanças de cebolinha em bandejas de isopor



HORTA E FLORES (2017)

Figura 7 – Cebolinha germinando



PLANTEI (2021)

2.3.1 Adubação

Para ter um melhor desenvolvimento, as plantas necessitam de determinados tipos de nutrientes, quando o solo não possui estes nutrientes, ou há uma deficiência dos mesmos, um adubo é acrescentado ao solo para tal correção. Adubo é um composto que visa corrigir alguma deficiência ou melhoria de nutriente que a planta necessita, é normalmente aplicado ao solo onde são absorvidos em um processo natural (AMARO *et al.*, 2007).

As plantas podem também absorver nutrientes através das folhas, porém, para o projeto em questão a absorção deles pela raiz é o mais importante, para plantas de pequeno porte, o adubo do tipo NPK é o mais utilizado.

NPK é a sigla dos nutrientes mais relevantes para o desenvolvimento das plantas; Nitrogênio (N), Fósforo (P) e Potássio (K), a junção de um adubo rico nesses três nutrientes dá nome ao adubo.

Há diversos tipos de NPK disponíveis, dependendo da dosagem necessária para a cultura e o solo específicos para cada vegetal, normalmente um agrônomo é o profissional que define a dosagem de cada um destes elementos no adubo, pois há fatores externos que variam a dosagem de nitrogênio, fósforo e potássio no adubo.

Entre as variáveis que o profissional observará é a época do ano em que as plantas mais necessitam dos nutrientes, pois durante os meses mais frios, o metabolismo da planta diminui e requer menos nutrientes; enquanto durante os meses mais quentes, primavera e

verão, os fertilizantes NPK são aplicados com mais frequência.

Em relação à composição do NPK, os elementos principais deste adubo variam em proporções de seus elementos, como exemplo do adubo utilizado no projeto, o NPK 4-14-8 (4 partes de nitrogênio, 14 partes de fósforo e 8 partes de potássio), é indicado para espécies que produzem flores e frutos, composto esse que deve ser adicionado ao solo antes do plantio (AMARO *et al.*, 2007).

2.3.2 Área de Cultivo

O local ideal para plantar uma horta deve ser facilmente acessível, ensolarado e próximo a uma boa fonte de água. O solo deve ser raso ou suave, profundo, de textura média, precisa ser areno-argiloso ou argilo-arenoso, além de arejado, bem drenado, mas com capacidade suficiente de retenção de água e rico em matéria orgânica. Alguns vegetais também podem ser plantados em canteiros murados, caixas, potes, arcos, vasos ou outros recipientes com pelo menos 20 cm de profundidade, com furos para drenagem de água na sua base. Deve haver uma fonte ou acesso à material orgânico para compostagem no local a fim de reduzir à dependência de fertilizantes comerciais. No projeto em questão, à terra para o plantio será adubada como citado no capítulo anterior, logo, à necessidade de compostos orgânicos para à adubação não é necessária.

O dimensionamento dessa área de plantio é muito relevante em todo o processo, cada planta necessita de um espaço diferente, como espaçamento entre plantas, área total onde estas plantas podem se desenvolver e profundidade de terra fértil para o melhor desenvolvimento de suas raízes. Esta dimensão influencia também na deposição dos componentes do adubo no solo, onde os mesmos são alocados de forma incorreta, como excesso em locais diferentes no solo, faltando nutrientes em locais isolados neste espaço.

Na tabela Figura 8 podemos ver um fragmento da tabela de cultivo de hortaliças cedido pela EMBRAPA, se identifica, entre outras informações, a área que cada planta precisa para se desenvolver, e a quantidade da safra da mesma em relação a esta área.

Figura 8 – Informações gerais sobre o cultivo de hortaliças

Espécie	Época Favorável de Plantio *	Tipo de Plantio	Espaçamento (m x m)	Início da Colheita (dias)	Produtividade Normal em 10 m ²
Abóbora	Ago.-Nov.	Direto/Covas	2,50 x 2,50	90-120	10 – 15 kg
Abobrinha	Ago.-Fev.	Direto/Covas	1,50 x 1,00	60-90	10 – 15 kg
Acelga	Abr.-Jun.	Muda/Canteiro	0,40 x 0,30	60-70	15 – 20 kg
Alface	Abr.-Jun.	Muda/Canteiro	0,25 x 0,25	60-90	160 pés
Alface verão	Ago.-Fev.	Muda/Canteiro	0,25 x 0,25	60-80	160 pés
Alho	Mar.-Abr.	Direto/Canteiro	0,25 x 0,10	150-180	4 – 6 kg
Almeirão	Abr.-Jun.	Muda/Canteiro	0,25 x 0,25	60-90	160 pés
Batata	Abr.-Jun.	Direto/Sulco	0,90 x ,030	110-120	20 – 30 kg
Batata-baroa	Abr.-Jun.	Direto/Leira	0,80 x 0,30	240-360	10 – 20 kg
Batata-doce	Ago.-Fev.	Direto/Leira	0,90 x ,030	120-150	10 – 15 kg
Berinjela	Ago.-Fev.	Direto/Muda	1,20 x 1,00	90-100	80 kg
Beterraba	Abr.-Jun.	Direto/Canteiro	0,20 x 0,10	60-80	30 – 40 kg
Brócolos	Abr.-Jun.	Muda/Covas	0,90 x 0,50	90-100	10 – 30 kg
Cará	Jul-Ago.	Direto/Leira	0,80 x 0,30	150-180	20 – 30 kg
Cebola	Abr.-Jun.	Muda/Canteiro	0,40 x 0,10	100-120	10 – 20 kg
Cebolinha	Abr.-Jun.	Muda/Canteiro	0,25 x 0,15	70-90	6 kg
Cenoura	Abr.-Jun.	Direto/Canteiro	0,20 x 0,05	90-110	20 – 30 kg
Chicória	Abr.-Jun.	Muda/Canteiro	0,25 x 0,25	80-90	160 pés
Chuchu	Ago.-Fev.	Direto/Covas	6,00 x 5,00	90-120	15 – 20 kg
Couve	Abr.-Jun.	Muda/Covas	0,90 x 0,50	70-90	16 molhos

Embrapa Hortaliças-Circular Técnica (INFOTECA-E), 2007.

2.3.3 Ambiente controlado

Em condições normais de cultivo, grande parte das plantas não consegue se desenvolver em determinadas épocas do ano, na maioria das vezes no inverno, por tanto, em um ambiente controlado, pode-se ajustar as variáveis que mantêm o ambiente em condições preferíveis para a planta, não importando as condições externas, possibilitando a colheita da mesma durante todo o ano.

Como citado no capítulo 2.3, o NPK pode ter sua composição alterada, em diferentes proporções de cada um dos seus elementos, essa proporção depende de muitos fatores externos, assim sendo, um sistema fechado e isolado de condições externas, que também propicia para a planta o seu ambiente ideal, elimina a necessidade de um cálculo para determinar a proporcionalidade dos elementos do adubo NPK, já que o ambiente não varia, o adubo se manterá o.

Um ambiente isolado ajuda a evitar que a planta seja suscetível a pragas como insetos, animais que se alimentam das mesmas ou seus frutos e a proliferação de fungos e outras pragas que possam deformar alguma característica da planta.

2.3.4 Influência da luz no desenvolvimento das plantas

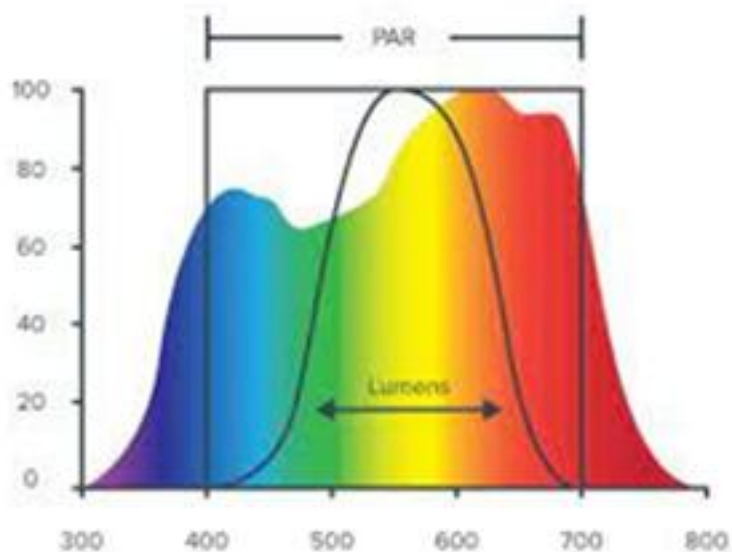
A luz do sol causa grande influência no desenvolvimento das plantas, a luz se divide em diferentes frequências eletromagnéticas e algumas delas têm uma influência no desenvolvimento das plantas diretamente, o conjunto de frequências de luz propícias para o desenvolvimento da planta são chamadas de luz par.

Fótons azuis (frequências entre 400 e 499 nm) promovem o crescimento das plantas, promove o desenvolvimento radicular e a produção de nutrientes e estimula a produção de clorofila e a troca de gases, os fótons amarelos e verdes (frequências entre 500 e 599 nm), permite a avaliação visual da saúde das plantas.

Fótons laranja e vermelho (frequências entre 600 e 700 nm) são eficazes para estimular o crescimento das plantas e melhorar a absorção da clorofila; age no controle da floração e no crescimento durante o dia.

E fótons vermelhos em diante (frequências entre 701 e 800 nm) promovem o pleno funcionamento da planta e aumenta a eficiência fotossintética da mesma.

Figura 9 – Amostragem de Luz Par



ITOGRASSEM CAMPO(2021)

As plantas não necessitam de luz do sol durante todo o dia, tanto o excesso quanto a falta de sol são prejudiciais à mesma, assim, como visto na (referência tal), cada planta necessita de horas de sol dependendo do seu estágio de desenvolvimento, no caso da cebolinha, este período é de 18 horas de sol na fase de eclosão das sementes e brotamento, e no momento da sua floração, esse período de luz do sol é de 12 horas diárias (“FOTOPERIODISMO”... , 2008-2022). No banco de dados de cada planta, há a variável que mostra o tempo de sol que a planta precisa, no caso, o tempo que a lâmpada de LED irá

ficar ligada dentro da estufa, fazendo assim com que a planta obtenha a luz necessária para seu desenvolvimento de maneira correta, sem alterações externas, como clima nublado ou sol em excesso.

Figura 10 – Fotoperíodo das plantas



"Fotoperiodismo" em Só Biologia. Virtuoso Tecnologia da Informação, (2008-2022).

2.3.5 Dosagem dos Insumos

A quantidade ideal de insumos deve ser controlada sempre que necessário, pois o excesso ou a falta de insumos pode acabar impactando negativamente o desenvolvimento das plantas.

O excesso de água, com outros nutrientes, contribui para proliferação de fungos e outras doenças, a falta de água contribui na falha de várias atividades fisiológicas da planta, como a fotossíntese e captação de nutrientes no solo.

Com pouco calor, a planta não consegue manter a sua fotossíntese de forma satisfatória, assim como a evaporação da água em suas folhas não ocorrer, atrapalhando o sistema fisiológico da planta, no seu ciclo de alimentação, em abundância de calor, a planta tende a evaporar mais depressa a água em seu anterior, antes de terminar seu ciclo de fotossíntese, podendo levar a secagem e morte do vegetal (KANEKO; CARRIJO; JUNQUEIRA, 2006).

O nitrogênio atua no crescimento das raízes, brotos, caules e folhas; a deficiência do mesmo pode amarelar as folhas da planta. Como a planta absorve e armazena este nutriente no início de seu ciclo de vida, a fertilização deve ser planejada do plantio.

O fósforo estimula a frutificação e a floração das plantas. A falta deste nutriente nas plantas causa subdesenvolvimento e coloração anormal. Este elemento desempenha um papel importante na formação da clorofila, responsável pela cor verde, e também contribui para a capacidade de absorver outros nutrientes e o desenvolvimento de raízes, frutos e sementes.

O potássio torna as plantas mais resistentes a pragas e outras influências nocivas. As plantas com baixo teor de potássio são mais fracas e produzem rendimentos mais baixos. É particularmente importante para a formação de tubérculos e para as plantas rizomatosas, ou seja, aquelas que crescem de forma rasteira.

Figura 11 – Lesões causadas na cebola pelo fungo



Frederick M. Aguiar (CT-133)

Figura 12 – Sintomas de clorose nas plantas



EDUARDO. IRRIGAT(2019)

2.4 Sistema supervisório

Com o avanço nos processos de produção e o aumento significativo no número de variáveis nestes processos, a observação humana em relação a estes dados deixou de ser satisfatória, devido à quantidade massiva de informações. A precisão nestes processos ficou muito mais fina, pois a não conformidade de certas características do processo pode deixar o material produzido de forma incorreta ou até causar acidentes, para isso, métodos de controle supervisório foram desenvolvidos para que um operador possa acompanhar o processo em tempo real, e receber estes dados com maior precisão, possibilitando uma ação correta do operador caso ele precise interferir no processo. (CURY, 2001)

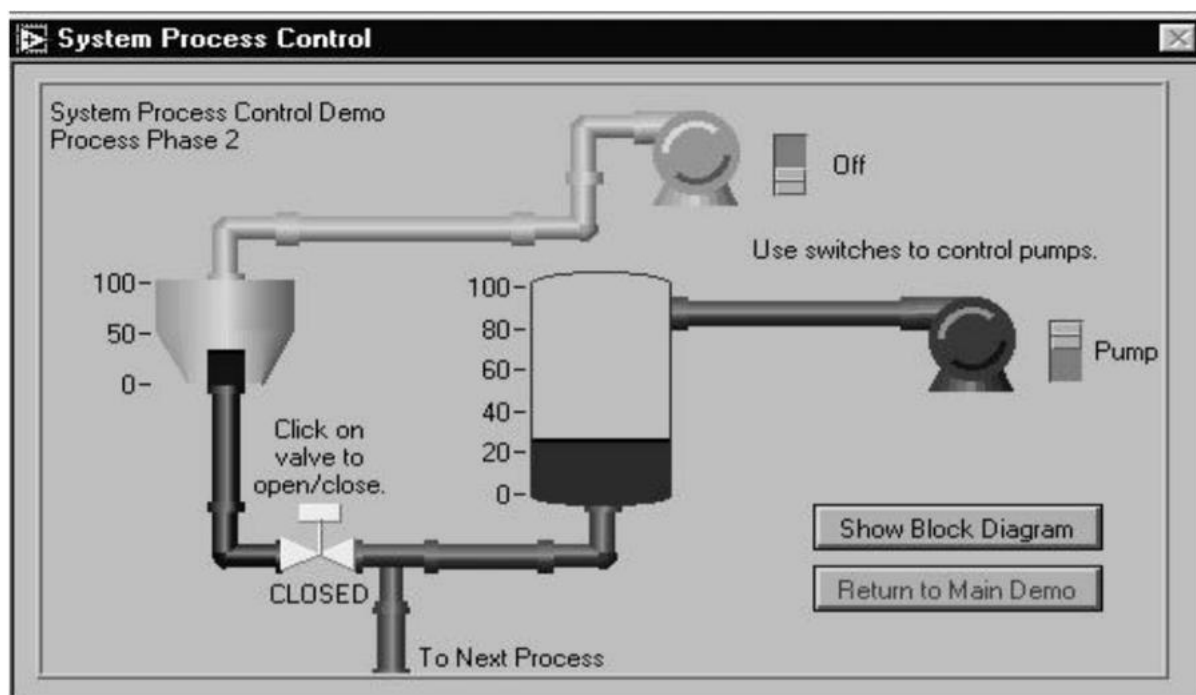
Os softwares de sistemas supervisórios tem como finalidade monitorar todos os dados de um sistema que está sendo controlado, tudo de forma automática. Estes sistemas colhem, medem e organizam todas as variáveis relevantes para o processo em questão, facilitando muito a interpretação de como o sistema se comporta, assim, o operador pode identificar qualquer alteração no processo, como falhas, ou também colher dados para eventuais melhorias, aplicando método para tais finalidades.

Grande parte dos sistemas supervisórios possuem um sistema que o usuário pode observar as mudanças das variáveis em questão, muitos deles são essenciais para certos dispositivos, pois a avaliação do operador é de suma importância.

As IHM's (Interface Homem Máquina) são utilizadas por dispositivos para que o operador possa ler informações do processo, e que o mesmo possa ou precise tomar alguma decisão. As telas em máquinas podem ser de inúmeras maneiras e com layout distintos para cada operação, mas feitas de maneira que o operador possa ler o funcionamento do processo de forma clara e com precisão, estas telas são desenvolvidas independente do tamanho da planta a ser analisada pelo sistema.

A Figura 12 mostra um exemplo de IHM aplicada em um sistema supervisório de processo de controle.

Figura 13 – Exemplo de tela de sistema supervisório



JURIZATO (2003)

2.5 Banco de dados

Um banco de dados é uma aplicação para armazenamento e recuperação de dados de forma eficiente. O que o torna um banco de dados relacional é como os dados são armazenados e organizados. Quando falamos de um banco de dados, nos referimos ao relacional.

Em um banco de dados relacional, os dados ficam armazenados em tabelas. Estas têm uma estrutura que se repete com cada linha, como em uma planilha de cálculo, são as relações entre as tabelas que definem este método. O modelo relacional forneceu uma forma padrão de representar e consultar dados que poderiam ser utilizados por qualquer aplicação. Desde o início, os desenvolvedores reconheceram que a principal força do modelo de banco de dados relacional era o uso de tabelas, que forneciam uma forma intuitiva, eficiente e flexível de armazenar e acessar informações estruturadas.

Figura 14 – Exemplo de banco de dados relacional



RIBEIRO M. MONOGRAFIAS (2022)

2.6 Controle

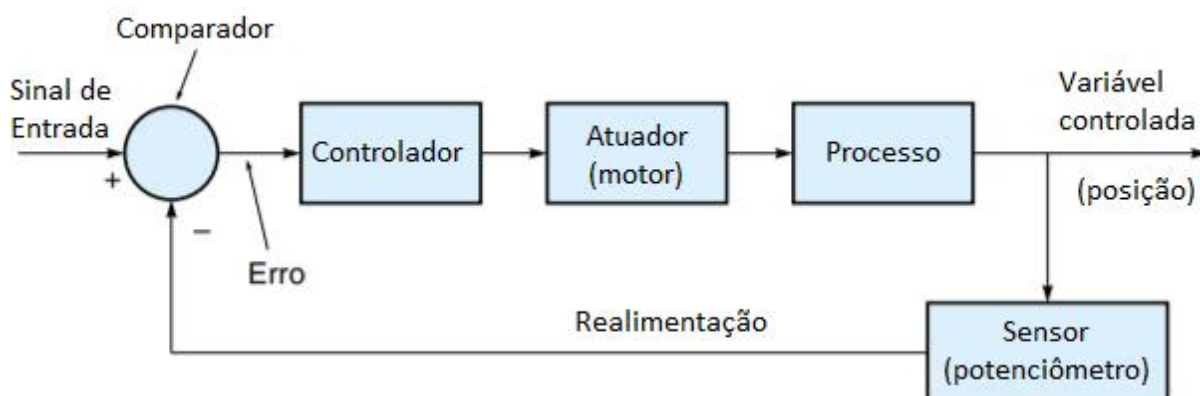
No escopo de engenharia, controle abrange técnicas de manipular saídas de um sistema através de processos, a partir de uma entrada (NISE, 2012).

Para reduzir os efeitos de distúrbios implementa-se a realimentação no sistema (formando um sistema em malha fechada, mostrada na Figura15), que se resume em sensorar a saída, e subtrair este sinal à entrada, denominamos esta diferença de erro. Um erro informa ao sistema se é necessário aumentar ou diminuir a saída, qualitativa e quantitativamente. (OGATA, 1982)

Sistemas de controle fazem parte do cotidiano de todos, ao abrir e fechar a válvula de controle de vazão do chuveiro a fim de controlar a temperatura da água, e até mesmo no nosso organismo com o controle de temperatura corporal.

Nise (2012), exemplifica que controle além de atingir o valor de sua variável, ainda cuida de algumas especificações como no caso de um elevador, onde não se pode acelerar demais ao ponto de se tornar desconfortável para quem está dentro, mas também não pode acelerar de menos, pois tomaria tempo e paciência dos usuários.

Figura 15 – Exemplo de sistema de controle de malha fechada



Adaptado de Kilian (2006)

2.7 Métodos de controle

Métodos de controle são diferentes técnicas de se manipular variáveis. Neste trabalho são estudados o controle de umidade do solo e de temperatura do ambiente das plantas.

Ressalta-se também que serão tratados apenas métodos de controle discreto, pois apesar dos microcontroladores funcionarem com clocks na faixa de MHz, fazendo múltiplas leituras por segundo, ainda assim este método não é contínuo no tempo, e sim discreto.

2.7.1 Bang-Bang

Este método de controle clássico com realimentação consiste em estabelecer limites de mínimo e de máximo para a entrada, a partir de uma lógica de comparação direta entre a entrada e os limites especificados, o sistema alterna abruptamente a saída entre duas opções, mantendo a variável controlada dentro do intervalo desejado. (WANG; WANG, 2007)

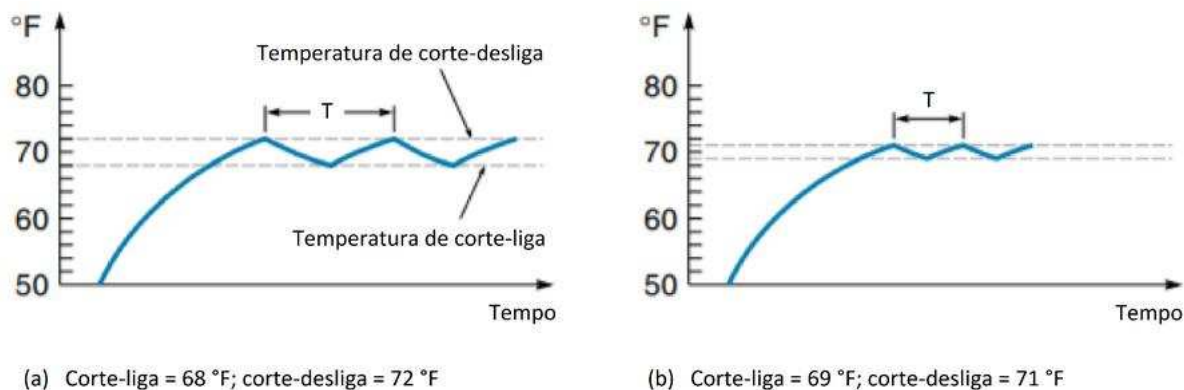
Este intervalo também é chamado de intervalo diferencial, histerese, ou zona morta (VISIOLI, 2006). No caso de uma zona morta, significa que não é estabelecido qualquer controle para a variável se ela estiver no intervalo desejado, isto seja uma economia de energia. No caso da histerese, o valor da variável deve atravessar completamente o intervalo para que a outra opção de saída seja acionada.

Numa aplicação ainda mais básica pode-se utilizar um intervalo nulo, determinando apenas um único valor para onde o sistema deve estabilizar a variável controlada. Neste caso a saída oscila em torno deste valor definido.

Uma particularidade muito empregada deste método, onde a saída tem as opções “ligado” ou “desligado”, é chamada de controle liga-desliga (ou on-off), o qual foi muito empregado em tecnologias de geladeiras antigas, por exemplo.

A Figura 16 mostra um exemplo, onde o atuador quando ligado aumenta a temperatura do sistema, e quando desligado, ele perde temperatura:

Figura 16 – Temperatura controlada por liga-desliga com histerese



Adaptado de Kilian (2006).

A umidade do solo diminui muito lentamente se comparada com o tempo de rega, onde ela aumenta em poucos segundos. Como o período de ciclo (T_{cyc} na Figura 15) é naturalmente longo, basta reduzir o intervalo de histerese para operar numa margem de umidade ideal (e segura) para a planta.

2.7.2 PID

O sistema de controle mais utilizado no mundo é o PID, uma combinação de ganhos proporcional, integrativo e derivativo do erro de malha fechada. Cada uma dessas três parcelas por si só caracteriza um sistema de controle.

No controle proporcional, o erro é multiplicado por um ganho K_p constante, sendo suficiente para algumas aplicações mais simples. No entanto, um controlador apenas proporcional apresenta problemas, tais como: aumentando o ganho K_p , o sistema apresenta maior máximo sobressinal (M_p), que seria o quanto a saída (PV - present value) passa do objetivo (SP - setpoint), ou até leva a instabilidade do sistema, onde o erro cresce indefinidamente e o PV se distancia indefinidamente do SP; diminuindo o K_p , o sistema vai demorar mais para atingir a tolerância aceitável (t_s grande), ou nem mesmo atingir, pois este controlador não é suficiente para zerar o erro de regime permanente (erro = SP - PV para realimentação unitária).

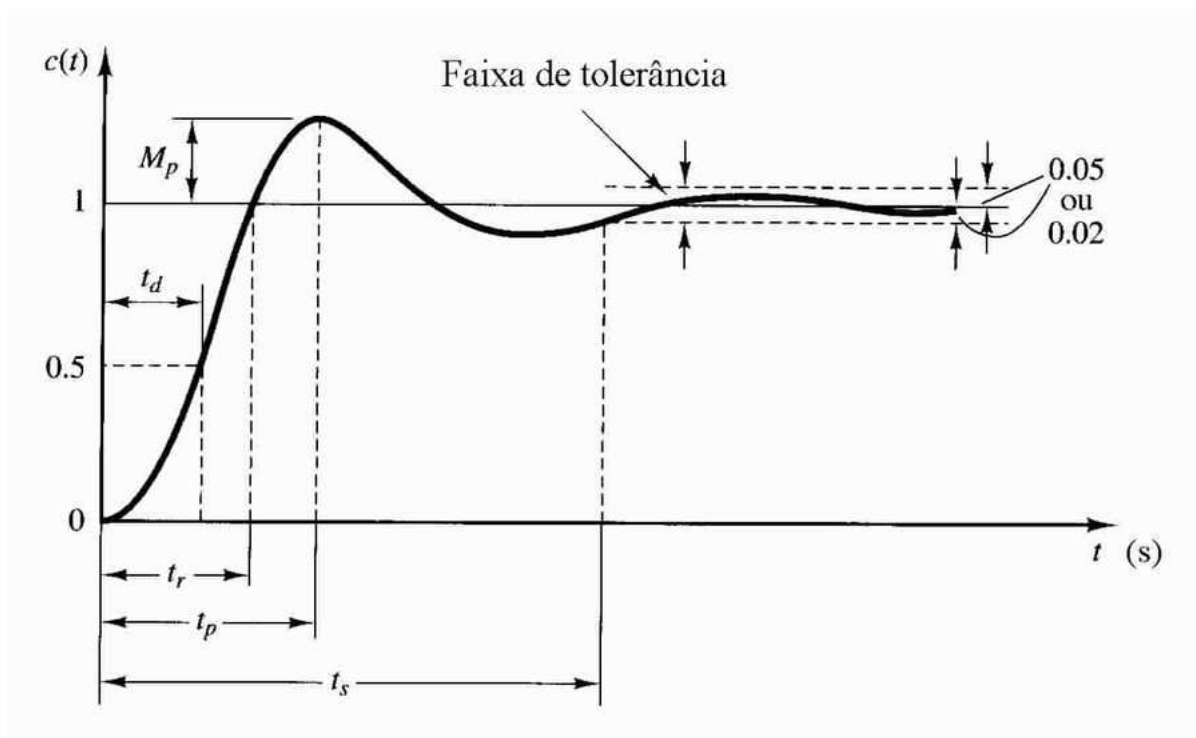
$$G_D(z) = K_P \quad (2.1)$$

Na Fórmula 1, GD representa a função de transferência, a qual recebe uma entrada, e como a função de transferência, neste caso, é uma constante, a saída desta função será a multiplicação do sinal de entrada por esta constante K_p .

No que lhe concerne, K_p representa a constante proporcional, denominada assim por se tratar de uma constante, determinada a fim de multiplicar o sinal de entrada da função de transferência GD.

As variáveis citadas acima podem ser visualizadas na Figura 17.

Figura 17 – Especificações da resposta ao degrau



Adaptado de OGATA (1990)

M_p : ao iniciar a resposta ao degrau, o sistema de controle acelera a saída em direção do SP, podendo até ultrapassá-lo devido à inércia. Caso ultrapasse o SP, o quanto se ultrapassa é medido em percentual, relativo ao tamanho do degrau em questão, e a este valor é dado o nome máximo sobressinal M_p .

Faixa de tolerância: é um intervalo centrado em 1, definido por um valor relativo previamente estabelecido que determina o valor aceitável, para mais e para menos, em que o PV pode oscilar.

t_d : tempo de atraso, representa o tempo que o sinal levou para atingir 50% do SP.

t_r : tempo de subida, representa o tempo que o sinal levou para atingir 100% do SP.

t_p : tempo de pico, representa o tempo até o sinal atingir o valor de M_p .

t_s : tempo de acomodação, é o tempo decorrido até o último momento em que o PV entra na faixa de tolerância.

Na Figura 10, observa-se uma variável onde se deseja alterar a amplitude $c(t)$ de 0 para 1 a partir do instante de tempo $t = 0$ [s], esta prática é conhecida como resposta ao degrau unitário, e é amplamente utilizada para avaliar o comportamento de um sistema.

O sinal parte de seu referencial com um atraso inercial que neste caso é avaliado medindo-se o tempo de atraso t_d que leva até atingir 50% da amplitude.

O primeiro instante em que o sinal atinge o valor 1 é chamado de tempo de subida t_r .

Após o tempo de subida, o sinal atingirá seu primeiro e maior pico de amplitude, este valor que ultrapassa o objetivo estabelecido é chamado de máximo sobressinal M_p , e o tempo até atingi-lo é o tempo de pico t_p .

O sinal continua a oscilar em torno do valor objetivado 1, enquanto o último valor a ser observado é o tempo de assentamento t_s , que representa o momento em que o sinal entra pela última vez na faixa de tolerância, que geralmente é 5% ou 2% para mais ou para menos.

Para contornar o problema de erro de regime permanente, acrescenta-se um ganho proporcional à integral do erro, chamada de taxa de reset ou tempo integral T_i . Além de solucionar este problema, eliminando o erro de regime permanente, agrega-se velocidade para se atingir o valor alvo do sistema de controle, aumentando tanto o máximo sobressinal (M_p) quanto o tempo de acomodação (t_s).

Com a combinação proporcional-integral, tem o controlador discreto PI:

$$G_D(z) = K_P + \frac{T_i}{1 - z^{-1}} \quad (2.2)$$

Mas o maior problema nem sempre será o erro de regime permanente, o último incremento que se aplica neste método de controle, o tempo derivativo T_d , tem a função de amenizar os problemas causados pelo incremento dos ganhos anteriores, o máximo sobressinal M_p aumentado por K_p e T_i , e o tempo de acomodação t_s aumentado por T_i .

O tempo derivativo (T_d) é uma margem de tempo à frente onde se estima uma previsão de onde estará o PV (Present Value) naquele momento através da tangente no PV atual, seu comportamento esperado é frear variações muito bruscas causadas pelos ganhos anteriores.

É válida a aplicação de um controlador discreto proporcional-derivativo PD:

$$G_D(z) = K_P + T_d \cdot (1 - z^{-1}) \quad (2.3)$$

Bem como a combinação destes três incrementos, proporcional-integral-derivativo, PID:

$$G_D(z) = K_P + \frac{T_i}{1 - z^{-1}} + T_d \cdot (1 - z^{-1}) \quad (2.4)$$

2.8 Eletrônica

A eletrônica é considerada a parte da física que trata dos circuitos eletrônicos, circuitos integrados, instrumentos, dispositivos semicondutores entre outros circuitos e tem como papel principal processar dados, controlar processos, representar, armazenar,

transmitir ou processar informações através de qualquer sinal sendo eles, por exemplo, vindo de sensores.

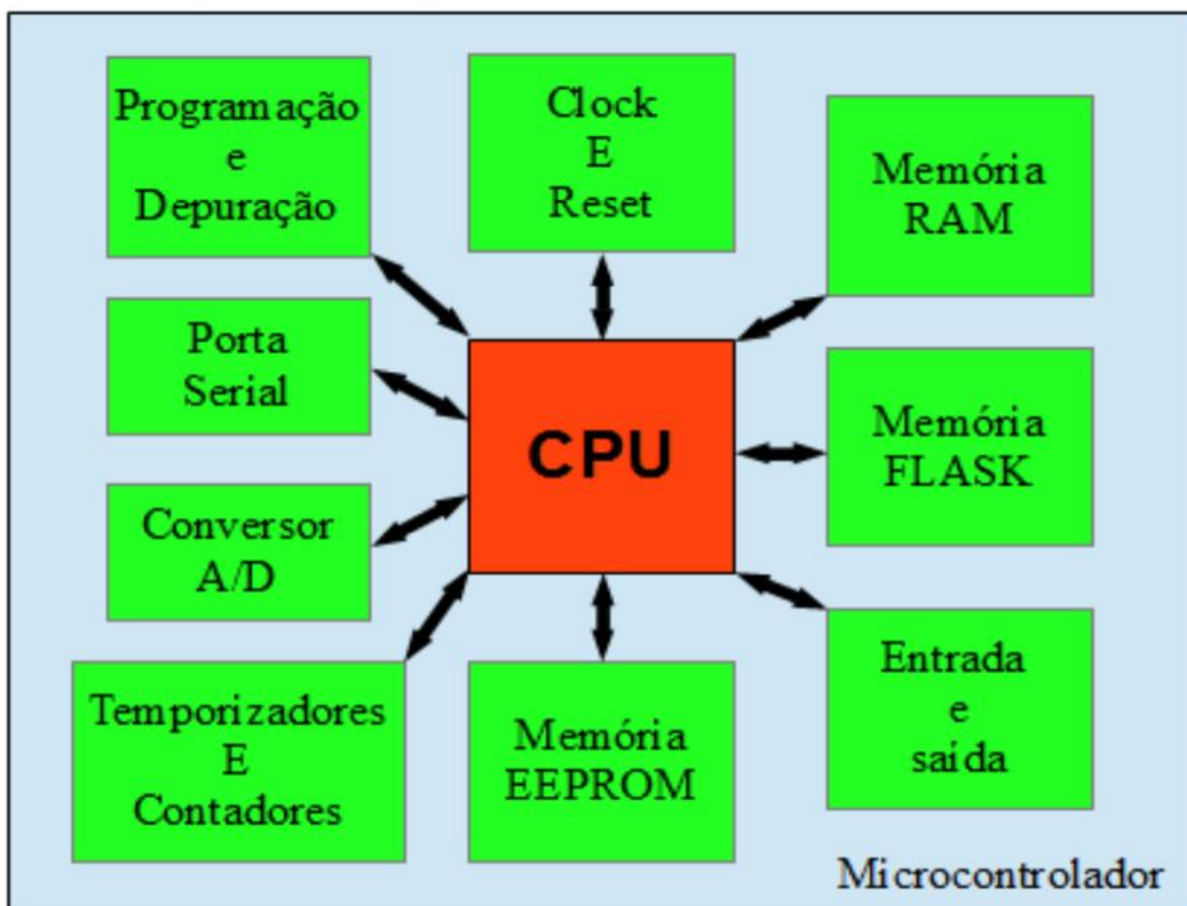
Um sistema eletrônico pode ser composto por componentes analógicos, digitais ou de uma combinação de ambos.

A eletrônica digital é composta por componentes discretos, necessária para o armazenamento de dados, também sendo úteis para cálculos. Os componentes digitais trabalham com informações booleanas, podendo assumir dois valores: nível lógico alto (1) ou nível lógico baixo (0) (HAUPT; DACHI, 2018)

2.9 Microcontrolador

Microcontrolador é um circuito integrado (CI) composto por um núcleo de processador, memórias voláteis (*RAM (Random Access Memory*, ou memória de acesso randômico) ou *SRAM (Static Random Access Memory*, ou memória estática de acesso aleatório), onde os dados são perdidos quando a energia é removida) e não voláteis (*flash* ou *EEPROM (Electrically-Erasable Programmable Read-Only Memory)*), memórias que mantêm dados armazenados mesmo sem o uso de energia), circuito de *clock*, e portas digitais e/ou analógicas de entrada e saída, onde pode-se conectar outros circuitos que fornecerão informações ao sistema e que receberão informações (como ordens, se tratando de atuadores) (GIMENEZ, 2010). A Figura 18 mostra um exemplo de arquitetura de microcontrolador.

Figura 18 – Exemplo de arquitetura de microcontrolador



KERSCHBAUMER (2018)

2.9.1 Arduino Mega

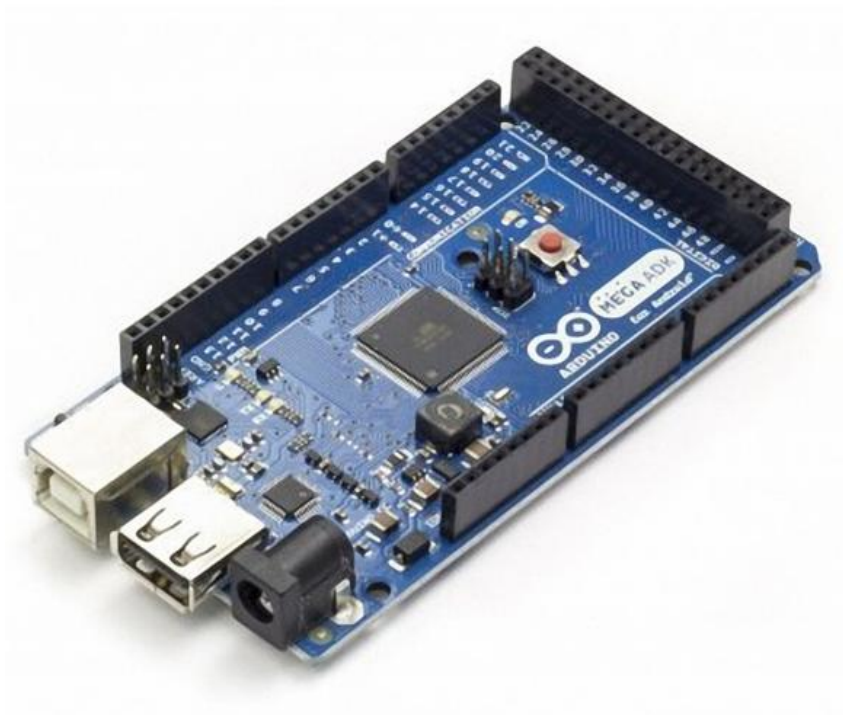
Arduino é uma plataforma de prototipagem eletrônica de *hardware* livre e de placa única, projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embutido, uma linguagem de programação padrão, a qual tem origem em Wiring, e é essencialmente C / C++. (ARDUINO, 2016)

É um dispositivo para prototipagem eletrônica em *hardware* livre, ou seja, o código de linguagem do *hardware* pode ser disponibilizado a qualquer pessoa, possibilitando alta colaboração entre os desenvolvedores. Dos vários modelos de placas de Arduino disponíveis no mercado, a placa Arduino MEGA se mostrou viável para o projeto.

O Arduino MEGA ADK Rev 3 é uma placa de microcontrolador baseada na ATmega2560, possui uma interface USB para conexão com telefones baseados em Android, com um CI MAX3421e. Possui 54 pinos de entrada/saída digitais (15 dos quais podem ser usados como saídas PWM), 16 entradas analógicas, 4 UARTs (portas seriais de hardware), um oscilador de cristal de 16 MHz, uma porta USB, uma tomada de alimentação, um cabeçalho ICSP e um botão de reset. O MEGA ADK é baseado no Arduino Mega 2560.

Similar ao Mega 2560 e ao Uno, porém com uma maior quantidade de pinos para ligação de dispositivos periféricos.

Figura 19 – Arduino MEGA



https://electropro.pe/index.php?route=product/product&product_id=363

Figura 20 – Especificações técnicas Arduino MEGA

Microcontrolador	ATmega2560
Tensão operacional	5V
Tensão de entrada (recomendado)	7-12V
Tensão de entrada (limites)	6-20V
Digital I/O Pins	54 (dos quais 15 fornecem saída PWM)
Pinos de entrada analógica	16
Corrente CC por pino de E/S	40 mA
Corrente CC para pino de 3,3 V	50 mA
Memória flash	256 KB dos quais 8 KB usados pelo bootloader
SRAM	8 KB
EEPROM	4 KB
Velocidade do relógio	16MHz
Chip de host USB	MAX3421E
Comprimento	101,52 milímetros
Largura	53,3 milímetros
Peso	36g

<https://docs.arduino.cc/retired/boards/arduino-mega-adk-rev3?queryID=d7eab2a92199a73450b6e9c14d8842ed>

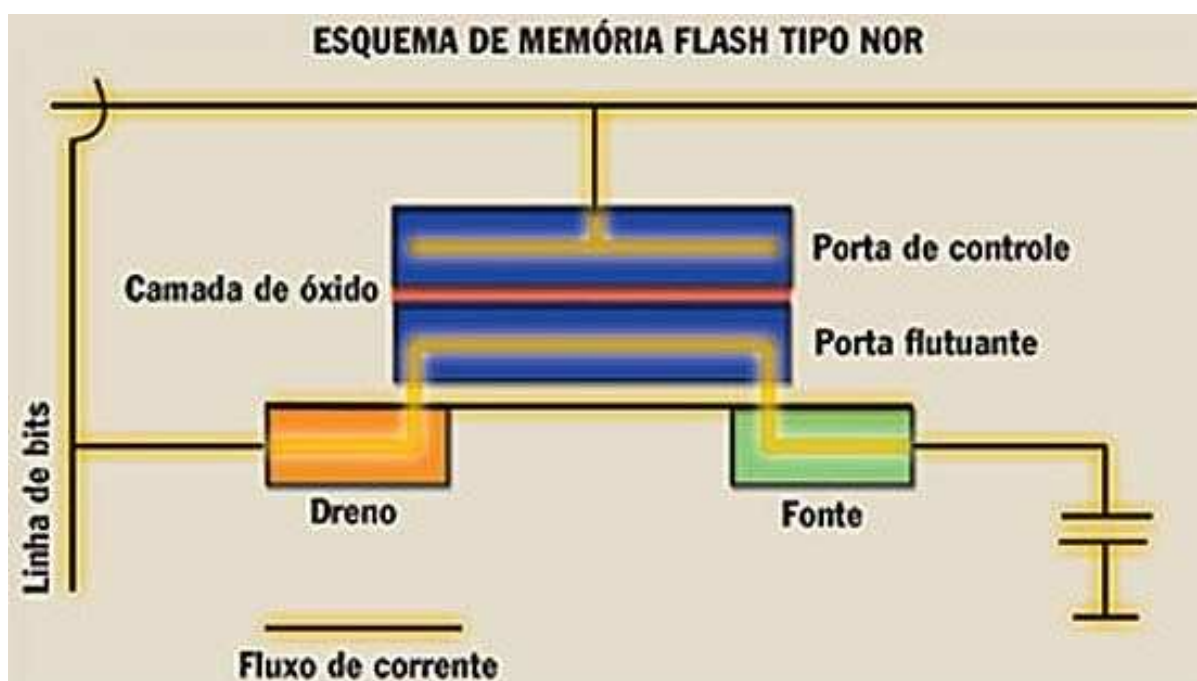
2.9.2 Memória Flash

A memória flash é um tipo de memória não volátil, significa que ela retém a informação armazenada mesmo quando a energia está desligada. A memória flash é uma variante da EEPROM (Electrically-Erasable Programmable Read-Only Memory). Existem basicamente dois tipos de memória flash, a NOR e a NAND, o primeiro permite que as células de memória sejam acessadas aleatoriamente, como na memória RAM, mas em alta velocidade. O tipo NAND, por outro lado, acessa as células de memória em alta velocidade,

mas sequencialmente.(HAMMERSCHMIDT, 2012)

Embora existam dois tipos de memória flash (NOR e NAND), suas arquiteturas internas são muito similares. Com o armazenamento flash, as informações digitais são armazenadas em chips flash de estado sólido, como pen drives, em vez de em discos rígidos rotativos. Cada célula de memória tem dois transistores, a porta de controle, usada para ativar a célula e ler dados, e a porta flutuante, onde a informação é armazenada. As memórias flash NAND operam em alta velocidade e acessam as células de memória sequencialmente, tratando-as todo em vez de individualmente.

Figura 21 – Memória Flash



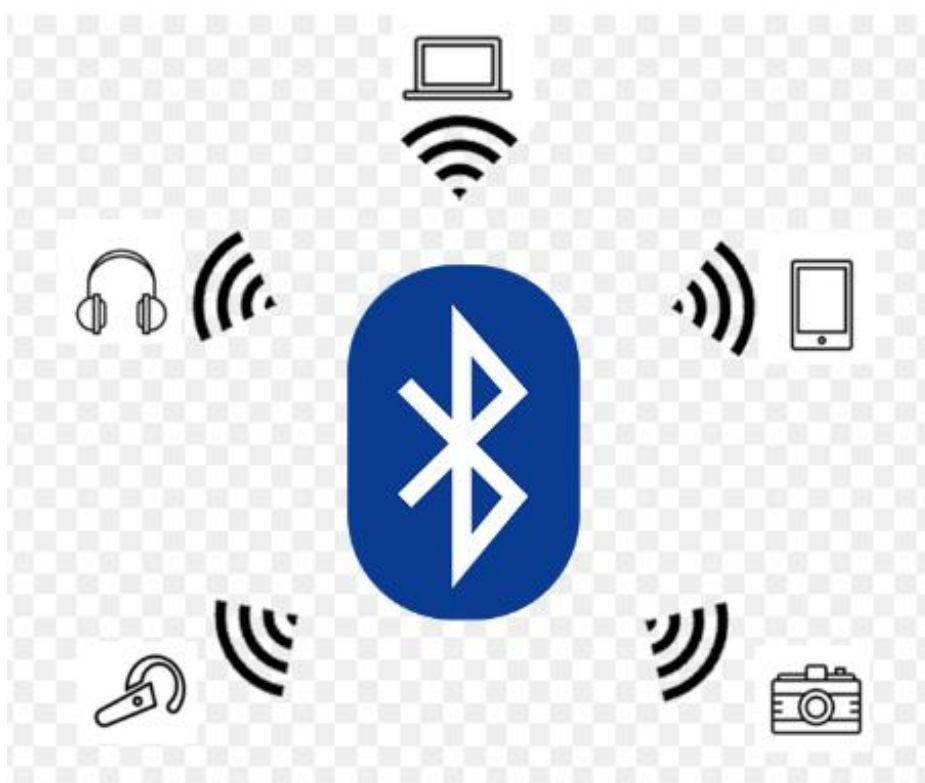
<https://www.tecmundo.com.br/cartoes-de-memoria/31521-tudo-o-que-voce-precisa-saber-sobre-cartoes-de-memoria.htm>

2.9.3 Bluetooth

Bluetooth é uma tecnologia projetada para comunicação sem fio para troca de dados e arquivos entre diferentes dispositivos móveis, de forma rápida e segura, sem a necessidade de cabos. O sistema de comunicação é pessoal, ou seja, ocorre apenas localmente entre dispositivos. Ele usa uma radiofrequência de ondas curtas (2,4 GHz) para estabelecer uma conexão entre dispositivos que compartilham a mesma frequência. Com baixo consumo de energia, o alcance é curto, portanto a comunicação só é possível entre dispositivos vizinhos, e ocorre através de uma rede chamada piconet. Este tipo de rede permite que até oito dispositivos sejam conectados, mas é possível aumentar o número através da sobreposição de vários piconets. Embora existam alguns tipos de conexões Bluetooth que atingem um alcance de 100 metros, a maioria dos dispositivos tem um pequeno alcance

entre 1 m á 10 m. Embora isto possa parecer uma desvantagem, as conexões sem fio de curto alcance garantem a segurança do usuário. Com Bluetooth, a segurança é ainda maior, pois cada conexão requer uma chave fornecida pelo usuário, de modo que nenhum arquivo de qualquer tipo é transferido de um dispositivo sem autorização. Atualmente, o recurso está disponível na maioria dos dispositivos, como smartphones, computadores, scanners, fones de ouvido e consoles e seus respectivos joysticks.

Figura 22 – Bluetooth



<https://pc-solucion.es/terminos/bluetooth/>

2.9.4 Nobreak

Com a finalidade de manter o sistema em funcionamento em casos de queda de energia, um *nobreak* será adicionado ao projeto. O *nobreak* recebe energia da rede e, por um chaveamento interno, fornece energia ao sistema supervisorio.

Este equipamento também regula a tensão e protege o sistema de oscilações na rede elétrica, trazendo segurança e prolonga a vida útil do sistema.

Figura 23 – Nobreak



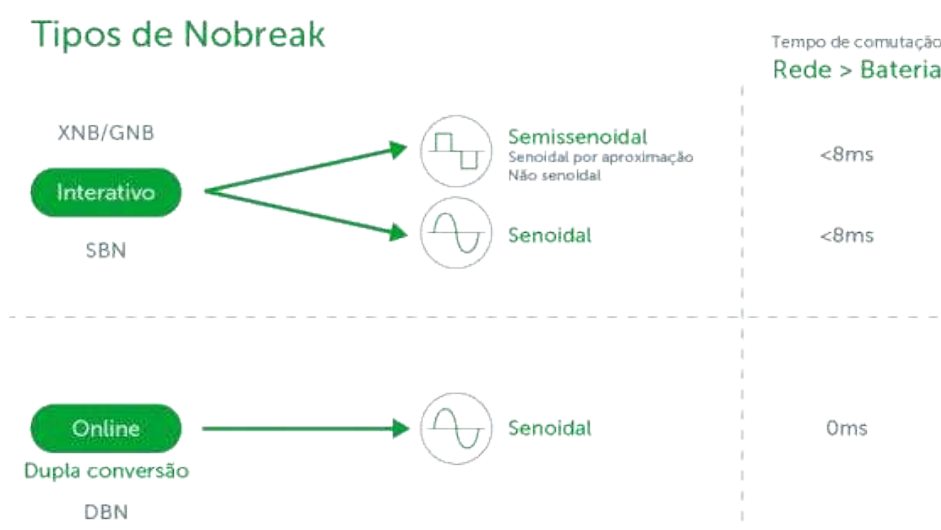
INTELBRAS(2020)

Na Figura 24 são mostrados dois tipos principais de nobreak.

No modelo interativo, um nobreak de menor custo, e maior economia de energia, ocorre proteção de oscilações da rede, curto circuitos e sobrecarga, e no caso da falta de energia ocorre um chaveamento com delay de cerca de 8 ms, a partir deste chaveamento, o *nobreak* comuta um chaveamento interno e fornece energia da sua bateria ao sistema por PWM, ou, dependendo do modelo do *nobreak*, por PWM senoidal.

No modelo *online*, sendo mais caro, o *nobreak* está o tempo todo recebendo energia da rede, retificando-a e convertendo-a num PWM senoidal. Deste modo o equipamento energizado por este *nobreak* tem a garantia de que receberá uma onda senoidal de qualidade, e ainda sem *delay* de chaveamento. Este processo torna a eficiência energética deste modelo menor que a de um *nobreak* interativo.

Figura 24 – Tipos de Nobreak



INTELBRAS(2020)

2.10 Sensores

Sensores são também designados por transdutores sendo utilizados para medir parâmetros pré-determinados. São componentes de extrema importância em processos onde variáveis são coletadas a fim de que o sistema compute dados e ações referentes às características coletadas.

Existem vários tipos de sensores, os mais utilizados na indústria, em processos de máquinas, são os sensores indutivos e capacitivos.

2.10.1 Sensor de umidade

O sensor de umidade consegue detectar variações de umidade do solo através da sua hastes capacitivas de acordo com condutividade, quanto maior a umidade do solo maior será sua condutividade e quando mais baixa a umidade do solo mais baixa será a condutividade.

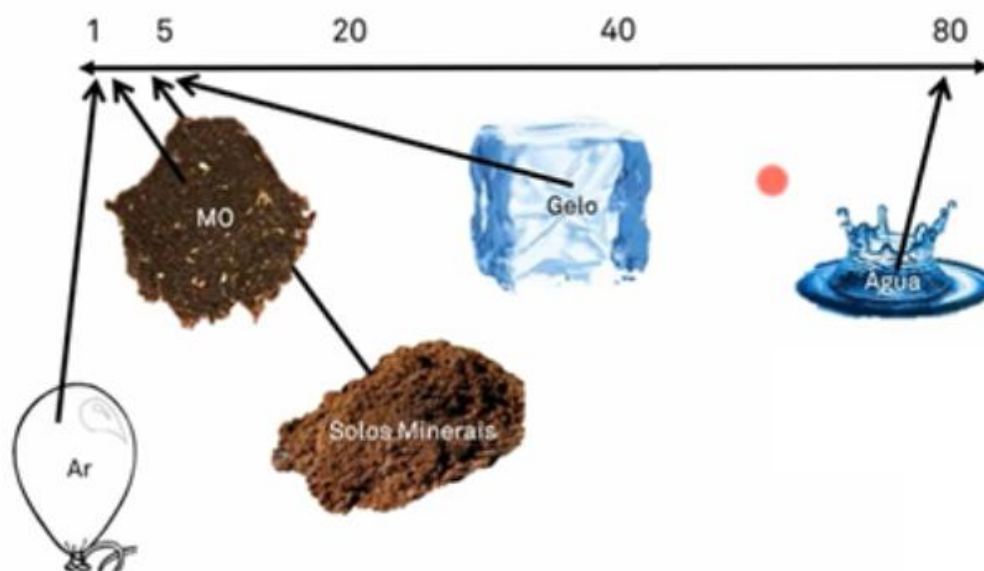
A capacitância de um capacitor pode ser descrita pela Equação 5, e depende da permissividade elétrica do meio ϵ , da área S das placas (do capacitor) e da distância d entre elas (ZAMBROZI JUNIOR *et al.*, 2005)).

$$C = \epsilon S/d$$

Para esta aplicação, permissividade elétrica do meio ϵ (verificar tipo de letra) é a grandeza variável, pois depende do tipo e da quantidade das componentes de matéria que compõe o meio. Podemos considerar S e d (verificar tipo de letra) constantes para esta aplicação com dispositivos semelhantes ao da Figura 26. (SILVA *et al.*, 2013).

Neste estudo, o meio é o solo e a água é o elemento que mais contribui para a permissividade elétrica, como indicado na Figura 25.

Figura 25 – Permissividade elétrica dos componentes do solo



LABSTORE (2020)

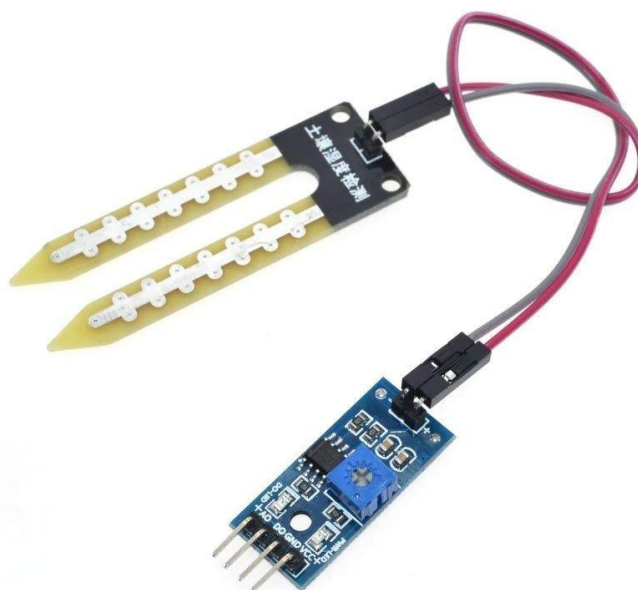
Tabela 2 – Permissividade elétrica relativa dos elementos do solo

Material	Permissividade elétrica do relativa
Ar	1
Solo Minerais	1 - 7
Matéria Orgânica	2 - 5
Gelo	5
Água	80

LABSTORE (2020)

A medida que o solo ao redor do sensor se torna mais úmido, o meio dielétrico poroso absorve umidade, este fenômeno se chama “difusão da água através de meios porosos”, aumentando a capacitância. O efeito oposto também ocorre, com o solo ao redor do sensor mais seco que no interior do mesmo, a umidade tende a se equilibrar, Diminuindo a umidade na região do sensor, e com isso, Diminuindo a capacitância.

Figura 26 – Baú da Eletrônica (2021)



Baú da Eletrônica (2021)

O sensor de umidade é composto pela sonda e pelo módulo. A sonda é o elemento mais acima da Figura 26, e é o componente que será introduzido no solo, enquanto o módulo é o componente inferior da Figura 26.

2.10.2 Sensor de temperatura

Os sensores de temperatura são utilizados em processos onde há a necessidade do controle da temperatura ambiente ou de um determinado elemento. Estes sensores possuem várias categorias definidas por sua forma de construção e de medição da temperatura. Nesta pesquisa, serão usados sensores de temperatura do tipo termistor, dispositivos que possuem elementos resistivos sensíveis à temperatura, como manganês, níquel, cobalto, ferro e titânio.

Existem dois tipos de termistores, os de coeficiente positivo de temperatura (PTC), e os sensores de coeficiente negativo de temperatura (NTC), ou seja, a resistência varia direta ou inversamente com a variação da temperatura.

O sensor DS18B20 é um termômetro de temperatura que opera entre 9 e 12 bit em suas medições de temperatura em graus celsius. A comunicação com o arduino é através de um barramento *1-Wire*, onde a transmissão de dados necessita de uma única linha de comunicação com o microprocessador, conforme a Figura 27. O DS18B20 não necessita de fonte de alimentação externa, e sua temperatura de operação é entre -55 °C a 125 °C com uma precisão de $\pm 0,5$ °C. FONTE??

A transmissão de dados para o arduino ocorre através de um sinal binário, e o sensor

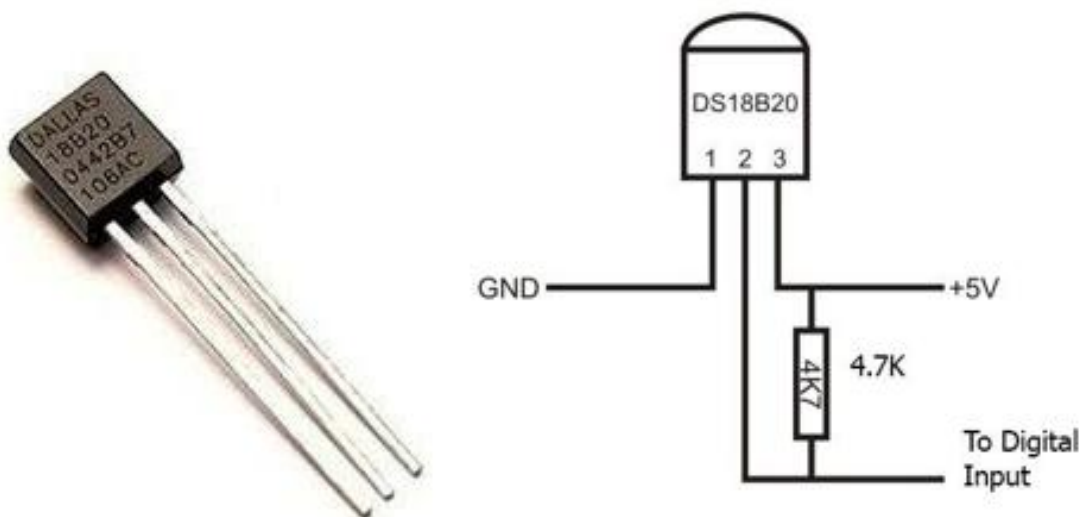
pode ser ligado a uma entrada digital do dispositivo. O DS18B20 gera um sinal binário para cada faixa de temperatura, e cabe a programação no arduino fazer uma análise desse sinal com a biblioteca *One Wire*.

Tabela 3 – Saída digital do sensor DS18B20

Temperatura °C	DigitalOutput (Binario)
+ 125	0000 0111 1101 0000
+ 85 -	0000 0101 0101 0000
+ 25.065	0000 0001 1001 0001
+ 10.125	0000 0000 1010 0010
+ 0.5	0000 0000 0000 1000
0	0000 0000 0000 0000
— 0.5	1111 1111 1111 1000
— 10.125	1111 1111 0101 1110
— 25.0625	1111 1110 0110 1111
— 55	1111 1100 1001 0000

MAXIMINTEGRATED (2021)

Figura 27 – Sensor DS18B20



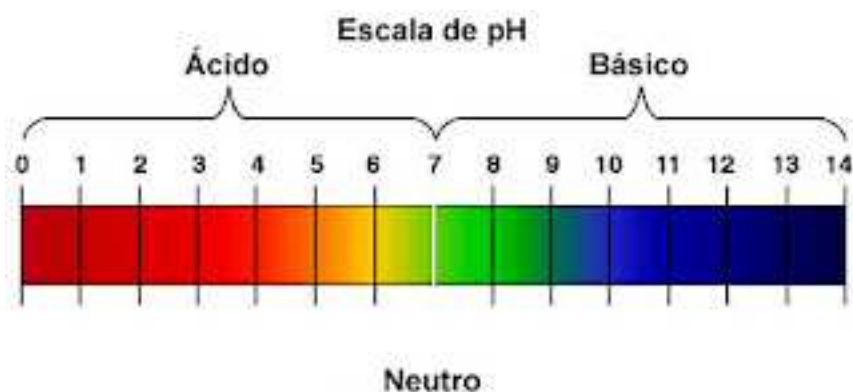
ARDUXOP (2021)

2.10.3 Sensor de pH

O medidor de pH tem como seu principal objetivo medir e analisar a escala de níveis de alcalinidade e acidez presentes em sólidos e líquidos, são muitos utilizados na produção agrícola pois o pH tanto da água e solo são fatores fundamentais para o desenvolvimento da planta. Sendo possível classificá-la como ácida ($\text{pH} < 7$), básica ($\text{pH} > 7$) ou neutra ($\text{pH} = 7$).

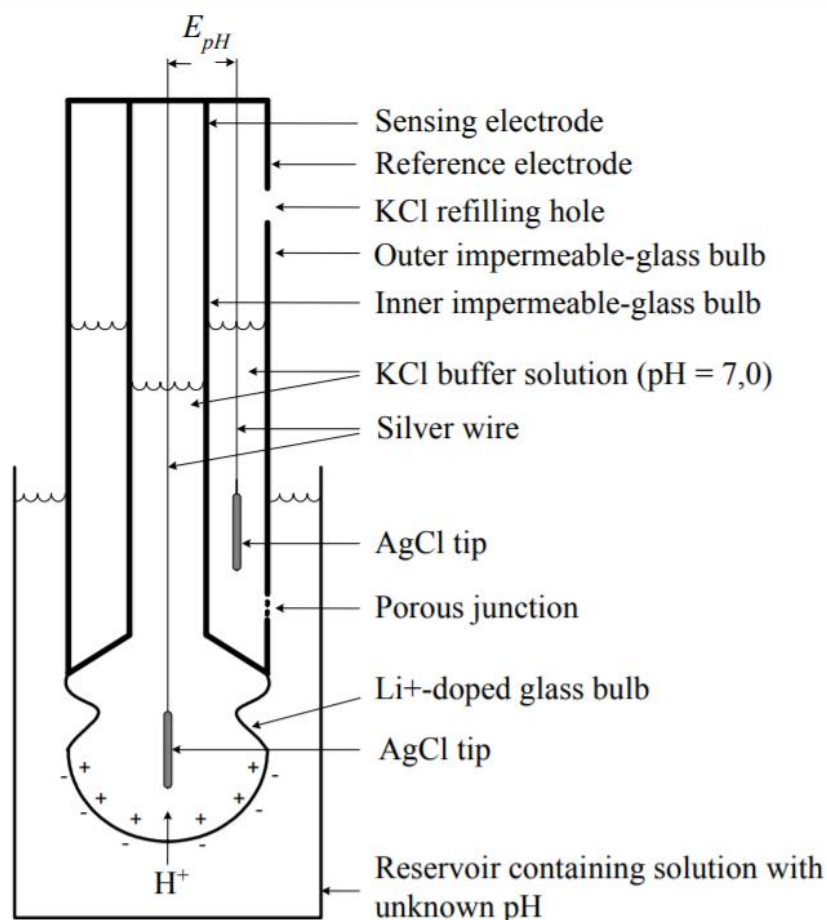
A esmagadora maioria dos sensores destinados a esta medição são compostos por um par de eletrodos de vidro, um de referência e outro de medição, este último, tem em sua extremidade um bulbo fino de vidro (para ser semi impermeável) dopado com Li^+ (íons de lítio), permitindo a passagem apenas de H^+ , como ilustrado na Figura 29).(LOTUFO; LUCENA, 2010)

Figura 28 – Escala de pH



Pibid-UCS (2017)

Figura 29 – Sensor de pH para soluções



DE LUCENA, S. (2006)

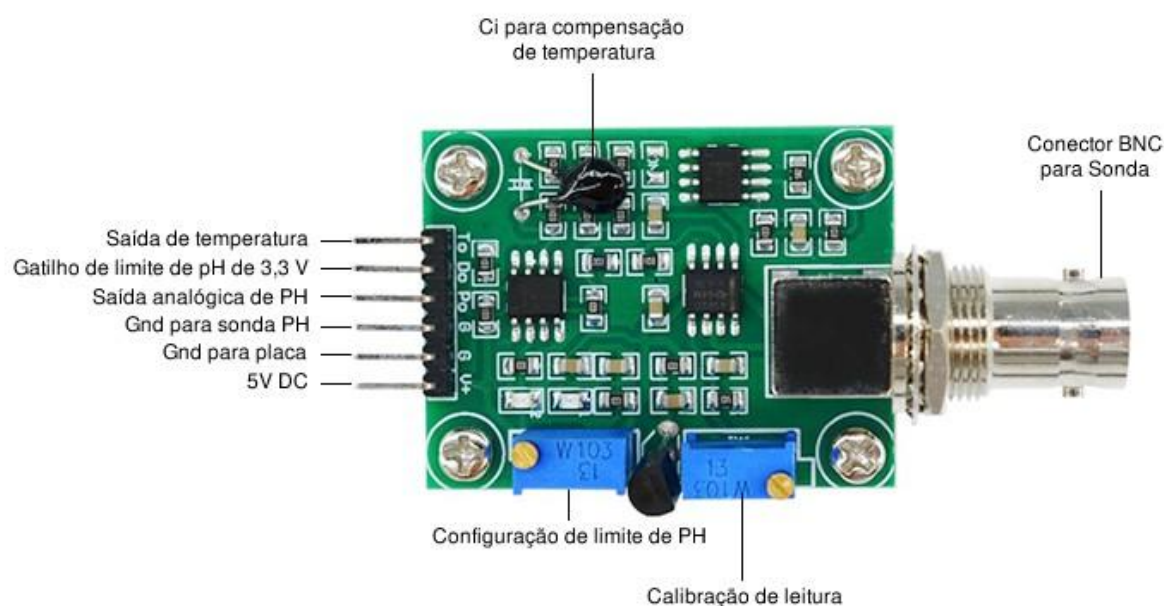
O PH4502C é um módulo comercial que faz a ligação com o arduino, como mostrado na Figura 28, retornando ao microcontrolador uma leitura analógica. Por padrão está saída do módulo varia de -5 V a +5 V, correspondendo ao intervalo de 0 a 14 do pH, mas o arduino não lê tensão negativa nas entradas analógicas, logo, se faz necessário um circuito de interface entre os dois dispositivos.

Para ajustar inicialmente o deslocamento de tensão, ajustando o pH 7 em 2,5 V, primeiro retira-se o sensor do módulo e aplica-se um curto-circuito nesta entrada do módulo (conector BNC para sonda, representado na Figura 30). Em seguida, com o auxílio de um multímetro na saída analógica do módulo (também é possível utilizar a própria entrada analógica do arduino para realizar esta leitura), deve-se ajustar o trimpot de ajuste de leitura, que está indicado mais à direita na Figura 30 (retângulo azul), até que o mesmo atinja 2,5 V. Assim, têm-se ajustado o centro do intervalo de medição com o centro do intervalo de pH.

Para ajustar a largura do intervalo de tensão de saída do módulo, que inicialmente é 10 V, para 5 V, é necessário o uso de pelo menos uma solução, ácida ou básica, que se tenha conhecimento do pH, para usá-la de referência para os ajustes. Existem alguns sachês, sendo comumente comercializados para esta finalidade, que vêm com um pó

a ser dissolvido em determinado volume de água destilada e temperatura ambiente. A embalagem menciona o ph que será atingido após a diluição. Com isto em mãos e com a sonda devidamente conectada no módulo, deve-se conectar o multímetro na saída analógica do módulo, submergir a sonda na solução de referência e ajustar o trimpot de configuração de limite de ph mostrado na Figura 30, até que a leitura analógica de ph corresponda ao valor característico da solução que se encontra explícito na própria embalagem.

Figura 30 – Pinagem do módulo do sensor de ph BNC PH4502C



USINAINFO (2021)

2.11 Atuadores

Os atuadores são dispositivos que fazem as ações pré-calculadas pelo sistema de controle. Convertendo o sinal elétrico enviado pelo sistema em diversos tipos de ações, os atuadores são responsáveis por realizar trabalhos referentes às suas atribuições, atuando na correção de seus respectivos parâmetros do sistema.

Nesta pesquisa, serão usados os seguintes dispositivos como atuadores:

2.11.1 Bomba hidráulico

No sistema de irrigação das plantas pode ser utilizado uma cisterna, onde ficará armazenada a água necessária para as plantas.

Para ocorrer o deslocamento da água até as plantas é necessário o uso de uma bomba hidráulica com a cisterna.

A bomba hidráulica tem como principal característica transformar energia mecânica através de um motor em energia hidráulica. Conseguem movimentar ou transportar fluidos de um ponto para outro. Ela também permite regular a pressão do sistema e gera uma vazão que impulsionada por uma força igual o peso do fluido que gera o deslocamento.(NORONHA *et al.*, 1995)

2.11.2 Sistema de controle de temperatura

Para compor o sistema de controle de temperatura, além de isolar o ambiente para reduzir a troca de ar, foram definidos atuadores para controlar essa troca de ar, permitindo ventilação em dias mais quentes, assim como um sistema de aquecimento que suprirá a temperatura nos dias mais frios.

2.11.2.1 Cooler de ventilação

Para possibilitar que a troca de ar seja controlada, será incluído um *cooler* para ventilar o ambiente (em horários de maior temperatura) a princípio isolado das plantas.

Cooler ou Ventoinha é um dispositivo mecânico utilizado para converter energia mecânica de rotação, aplicada em seus eixos, em aumento de pressão do ar, e sua principal função é fazer com que o ar circule e diminua a temperatura do ambiente.(TERMOQUÍMICA. . . , 05/07/2020)

2.11.2.2 Sistema de aquecimento

Para suprir o aquecimento do ambiente, quando a temperatura estiver abaixo do ideal, será necessário um dispositivo que dissipe potência elétrica na forma de calor, como uma resistência ou até mesmo lâmpadas incandescentes, ou infravermelho.

(PANDORFI *et al.*, 2005) fez uma análise técnico-econômica comparando estes métodos de aquecimento de ambiente, mostrando que resistência elétrica e lâmpada incandescente foram mais viáveis economicamente, dando destaque à lâmpada incandescente, com rendimento 3,6% superior à resistência.

3 DESENVOLVIMENTO DO PROJETO

O cultivo de plantas de pequeno porte demanda níveis de precisão no fornecimento dos nutrientes e no manejo dos métodos de plantio. A fim de obter-se alto desempenho no desenvolvimento dos cultivares, o presente projeto propõe a utilização de métodos de controle para gerenciar as necessidades das plantas.

O estudo intenso do cultivo de pequenas plantas, tais como a satisfação precisa de suas necessidades, é o nicho de atuação do projeto, que aplica métodos de controle para fornecer à planta nutrientes e atividades de cultivo que garantam seu máximo desempenho em desenvolvimento.

Cultivando plantas visando um aumento no desenvolvimento, tanto em tamanho, tempo de germinação e outros fatores característicos, um dispositivo eletrônico no qual em sua memória há uma biblioteca de dados de uma determinada planta, tudo para o cultivo no melhor desempenho, conectado a ela por sensores que controlam os atuadores para fornecer os nutrientes e outros insumos que a planta necessita com precisão, para que o desenvolvimento da mesma seja melhor que os métodos convencionais, utilizando recursos e controle para que a planta não sofra com excedentes, reduzindo seu desempenho.

Para melhor controle do processo, a planta ficou sujeita ao sistema desde a sua germinação, garantindo as melhores condições.

A planta foi cultivada em um vaso, pois com isso, o controle de umidade na terra é mais eficiente e individualizado.

O atuador responsável pela manutenção da umidade do sistema é uma pequena bomba d'água, que fica fixa sobre a borda de cada vaso, liberando água conforme a necessidade da planta e mantendo a umidade padrão para determinada planta.

Os atuadores responsáveis pelo controle da temperatura, especificamente para resfriamento, são um *cooler* de ar, que desloca ar frio do ambiente para dentro da estufa, e uma abertura ao lado oposto da estufa para que o ar quente possa sair, diminuindo a temperatura dentro da estufa. Em relação a baixa temperatura dentro da estufa, uma lâmpada de aquecimento é utilizada, já que a cebolinha, assim como demais hortaliças, deve ser mantida a uma temperatura entre 15,5 °C e 21,1 °C, que promove melhor formação de bulbos e maior produção ((AMARO *et al.*, 2007)).

Na programação do dispositivo, o banco de dados da planta auxilia a acompanhar os parâmetros de seu crescimento.

Todo o processo de controle é repetido até o momento de colheita da planta, tempo este indicado pelo dispositivo. O processo é acompanhado por um operador através de um sistema supervisor, que consiste em receber as informações coletadas em um cartão SD, adicionando estes dados em uma planilha Excel programada para indicar parâmetros pertinentes ao projeto, como o acionamento dos sensores e o momento de operação de cada atuador.

3.1 Modelagem

A modelagem consiste em uma explicação resumida do funcionamento do sistema por meio de um fluxograma das ações para as tomadas de decisão no decorrer do funcionamento do processo, desde o plantio até o momento da colheita da planta como representado na Figura 31, abrangendo também um resumo do esquema elétrico do dispositivo apresentando as conexões entre sensores e atuadores com o sistema eletrônico, representado em um diagrama na Figura 32 .

Figura 31 – Diagrama do processo de cultivo da planta

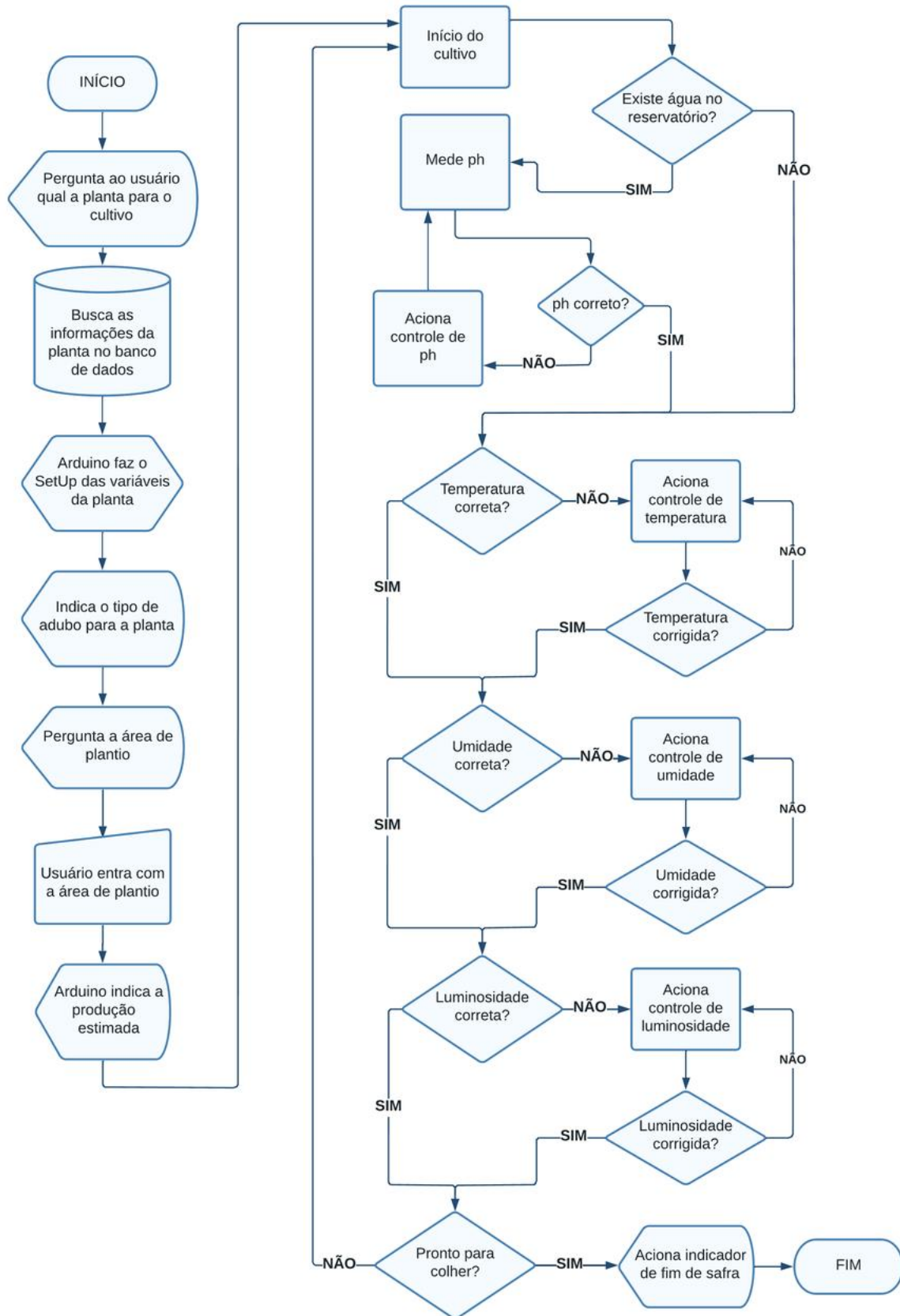
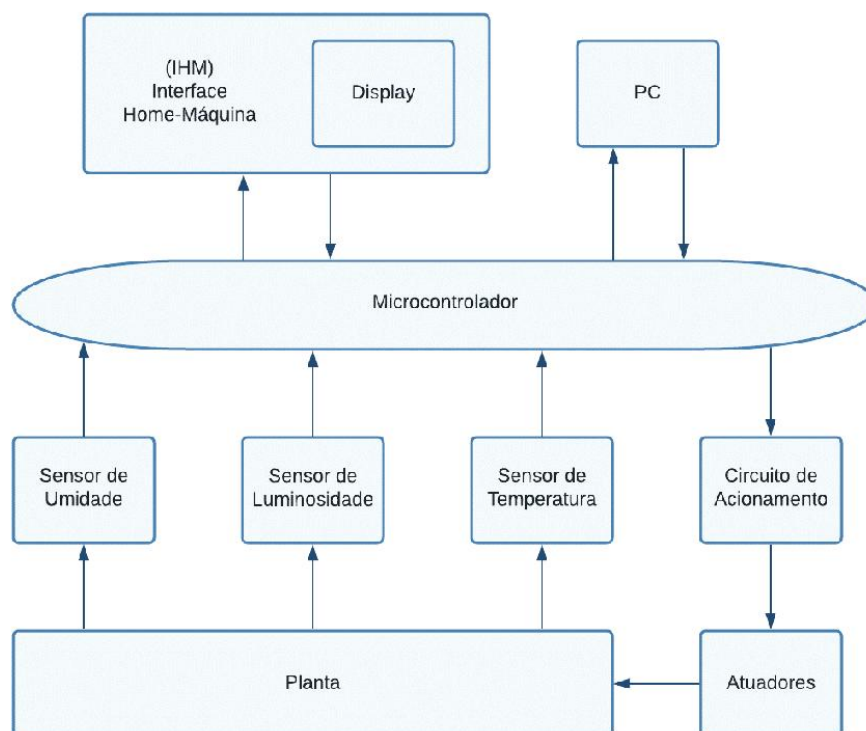


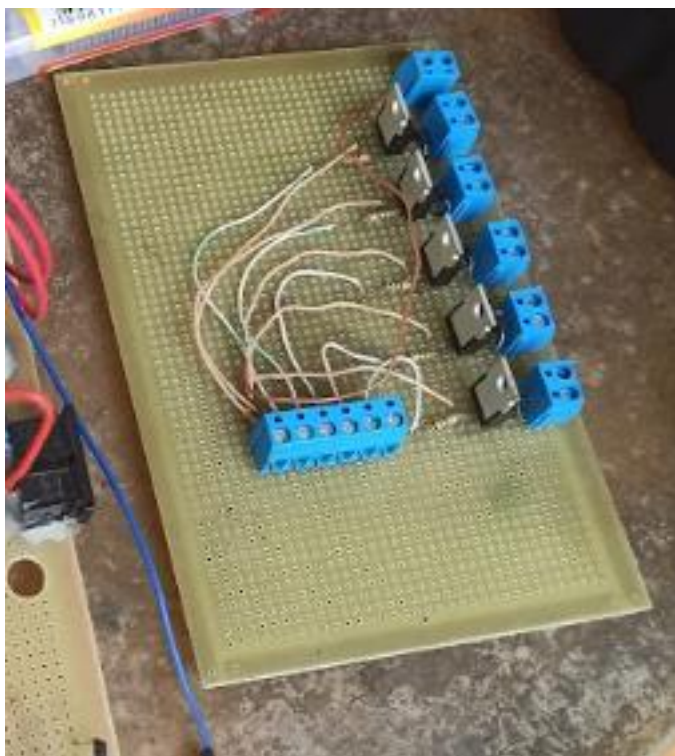
Figura 32 – Diagrama do sistema eletrônico



Autoria própria (2021)

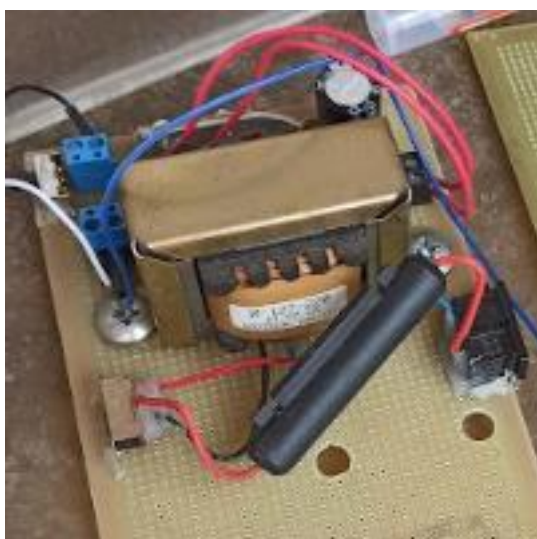
3.2 Ligação dos dispositivos periféricos

Os componentes periféricos, como sensores e atuadores, recebem e enviam sinais na tensão de 5V para o Arduino, porém, a alimentação para tais componentes funcionar exige uma tensão maior do que a cedida pelo Arduino. Para os acionamentos dos dispositivos 12V, dispositivo 1, 2 e etc, foi desenvolvida uma placa que recebe o sinal 5V do Arduino, acionando transistores que no que lhe concerne vão alimentar o dispositivo referente com 12V.

Figura 33 – Desenho do esquema elétrico de acionamento da placa de 12V

Autoria própria (2022)

O Arduino possui uma tensão de funcionamento de 12V, optou-se em manter o dispositivo ligado a uma fonte de alimentação oriunda da rede elétrica, um conversor de tensão de 127V para 12V foi projetado para alimentação da placa microcontrolada do dispositivo.

Figura 34 – Esquema elétrico fonte 127V/12V

Autoria própria (2022)

3.2.1 Esquema de ligação dos sensores

Cada sensor do dispositivo contém ligações distintas e formas próprias de processamento e transmissão de informação, pois cada um transmite sinais para um microcontrolador a sua maneira, para uma melhor interpretação de tais dados, o Arduino possui portas de entrada também distintas, além das digitais e analógicas para receber os mais diversos dados, permitindo que a programação interna interprete cada um de forma sucinta, compreendendo a escala de informação de cada sensor.

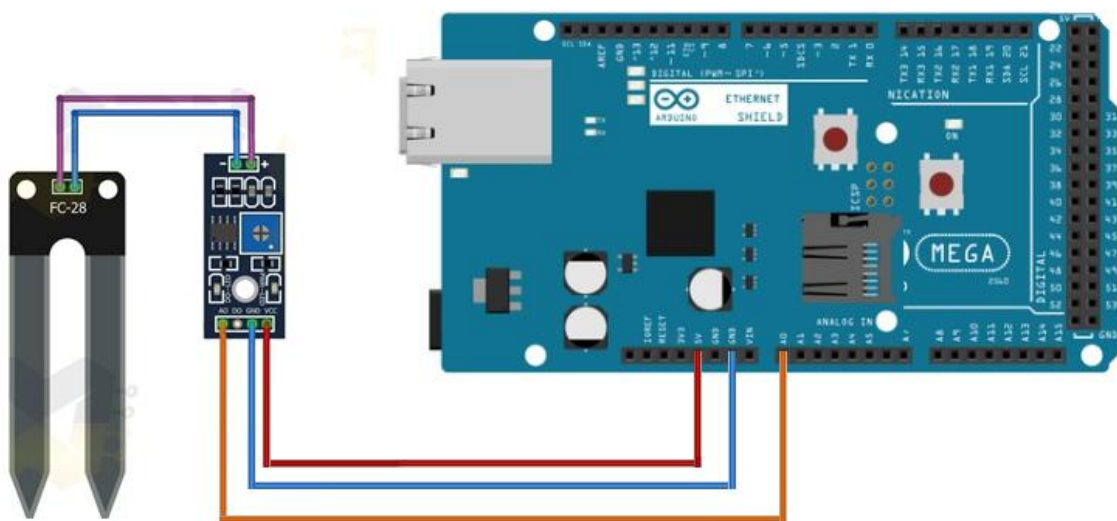
3.2.1.1 Sensor de umidade

O módulo sensor de umidade do solo pode medir a umidade no mesmo em um ponto específico. Ele consiste em duas partes, um sensor que usa duas sondas para medir a umidade comparando a corrente entre as sondas, e um circuito com um trimpot para ajustar a sensibilidade. O módulo sensor de umidade do solo é alimentado com Vcc de 5 V e GND, e possui duas saídas.

A saída digital pode ter seu valor de umidade calibrado pelo trimpot para que a saída passe de 0 para 1. Já a saída analógica, mostra um valor que varia entre 0 e 5 V.

As conexões do módulo da Figura 35 podem ser mais facilmente visualizadas na Figura 36, a saída analógica do sensor (A0) é ligada na primeira entrada analógica no Arduino. Para a alimentação do kit, o pino de Vcc do módulo é conectado na saída de 5V do Arduino e a conexão GND do módulo tem seu correspondente de mesmo nome no Arduino.

Figura 35 – Conexão do sensor de umidade do solo no arduino

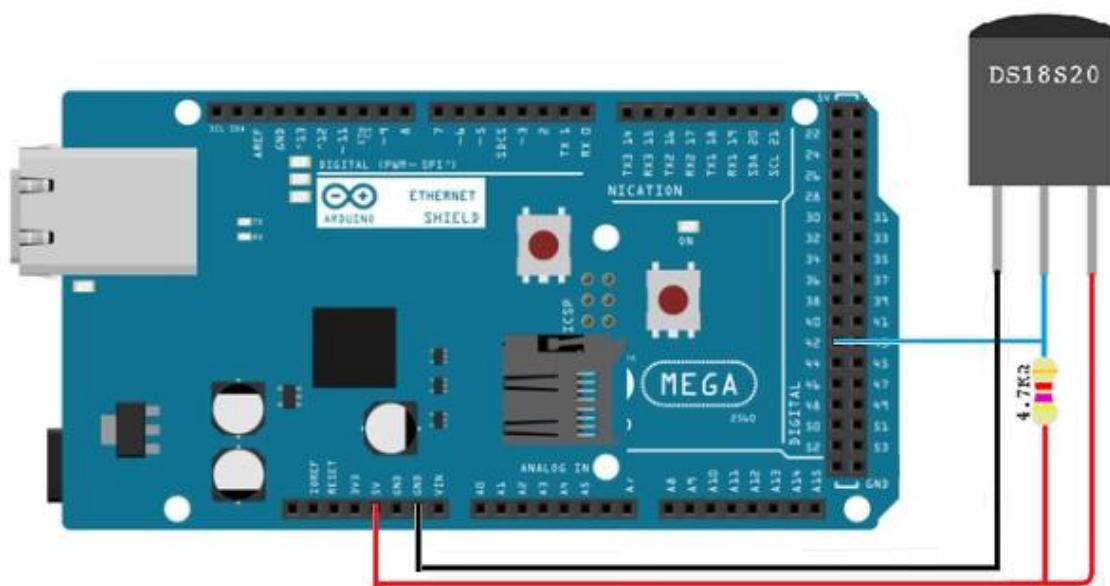


3.2.1.2 Sensor de temperatura

Normalmente, os sensores de temperatura são analógicos e fornecem um valor de tensão ou corrente, que é interpretado pelo Arduino ou outro microcontrolador. O sensor DS18B20 tem sua própria capacidade de processamento, pode ler a temperatura, interpretá-la e enviar a informação do valor de temperatura em graus Celsius para o microcontrolador, utilizando dois pinos como alimentação Vcc e GND e um pino único para transmissão de informação (protocolo de comunicação One wire ou 1-wire), e pode medir temperaturas entre $-55\text{ }^{\circ}\text{C}$ e $125\text{ }^{\circ}\text{C}$ com uma precisão de cerca de $0,5\text{ }^{\circ}\text{C}$ na faixa de $-10\text{ }^{\circ}\text{C}$ a $+85\text{ }^{\circ}\text{C}$. Cada sensor DS18B20 tem um único endereço serial de 64 bits, o que permite ao projetista conectar vários sensores ao mesmo barramento e usar uma única porta do microcontrolador para obter os valores de temperatura de cada sensor separadamente.

Para a conexão do sensor de temperatura DS18B20, como representado na Figura 36, seu pino à esquerda é conectado no pino de 5 V do Arduino, enquanto o da direita é conectado no pino de referência GND do mesmo. A conexão central do sensor que transmitirá dados no formato digital está ligada no pino 2 de entrada digital do arduino, o qual também está ligado em *pull-up*, conectado a um resistor de $4,7\text{ k}\Omega$, o qual é conectado no pino VCC de 5 V do Arduino.

Figura 36 – Conexão do sensor DS18B20 com o Arduino(2021)



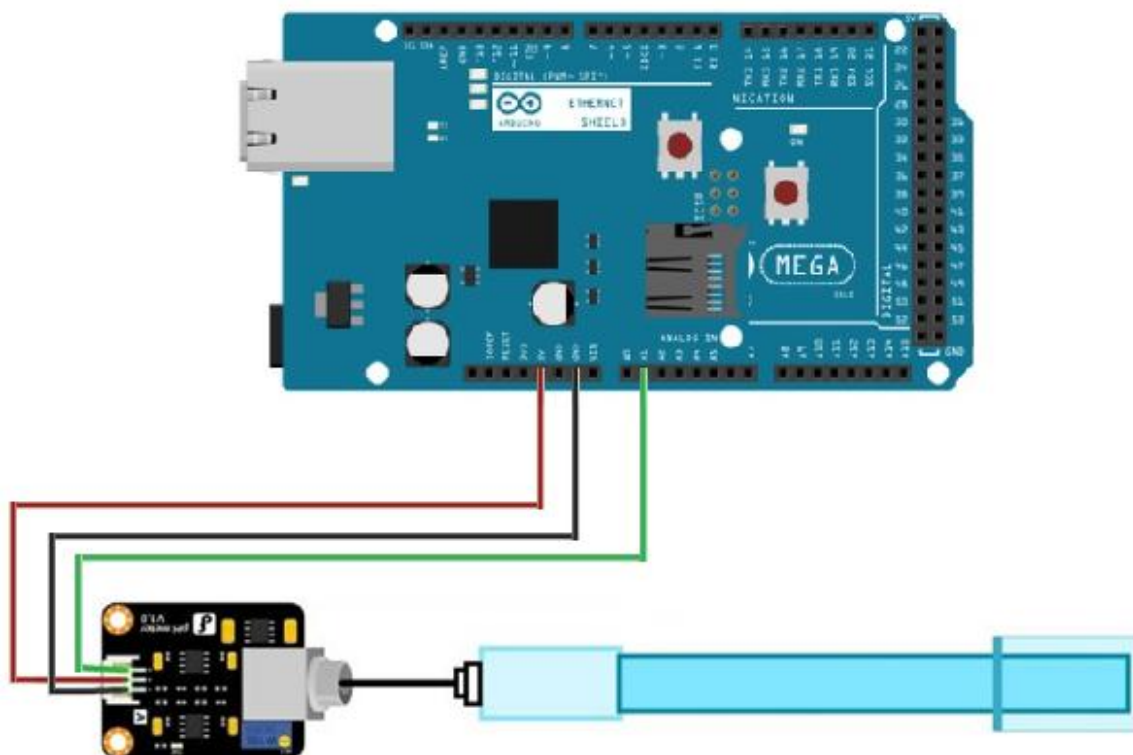
Autoria própria(2022)

3.2.1.3 Sensor de pH

O sensor de pH mede o potencial hidrogeniônico de uma solução para determinar seu nível de acidez ou alcalinidade. Sendo equipado com um regulador de tensão integrado, a placa suporta uma tensão de alimentação entre 3,3 e 5,5 VDC, compatível com Arduino e diversos outros microcontroladores. O sensor consiste em uma sonda, geralmente feita de vidro, com uma cápsula preenchida com uma solução de armazenamento chamada tampão de pH. A mesma proporciona um ambiente onde os íons H^+ podem ser armazenados e usados como base para leituras futuras. Quando o sensor é imerso na solução de teste, os íons hidrogênio na troca líquida com íons carregados positivamente na cápsula de vidro e criam uma espécie de potencial eletroquímico nela, convertido em sinais que o módulo pode ler.

Para a conexão do módulo do sensor de pH no Arduino, conectam-se os pinos (“5V DC” e “Gnd para placa” mostrados na Figura 30) de alimentação do módulo nos respectivos pinos (de 5 V e GND) do Arduino, ligações essas representadas em vermelho e preto na Figura 37. E na cor verde está representada a conexão da saída analógica do módulo com a entrada analógica A1 do Arduino.

Figura 37 – Esquema elétrico de conexão do kit de sensoriamento de pH



Adaptado de Cap Sistema (2021)

3.2.2 Atuadores do Dispositivo

O sistema possui atuadores com a finalidade de corrigir as características da planta que saem do padrão, suas ações são imediatas referente a estas diferenças e o desligamento de tais atuadores se concretiza com a leitura da informação dentro do intervalo estabelecido no *SetUp* do sistema.

3.2.2.1 Irrigação

Quando o sensor de umidade detecta um decréscimo do valor de umidade pré-estabelecido, neste caso é de 50%, o sistema de irrigação é ativado. Este sistema opera por gotejamento de água no vaso onde a planta se encontra, de modo a conduzir a água com uma bomba hidráulica, deslocando por tubos oriundos do sistema de armazenamento de água para irrigação. A bomba irá operar até que a próxima medição de umidade do solo meça um valor de 50% de umidade na terra, valor este estabelecido, o sistema interrompe o funcionamento da bomba hidráulica.

Figura 38 – Bomba Hidráulica na estufa



Autoria própria(2022)

3.2.2.2 Atuadores referentes ao sensor de pH

Diferente dos outros sensores do projeto, em que os mesmos podem ficar imersos no ambiente em que estão coletando os valores de medição, o sensor de pH não pode ficar imerso na água a todo o momento, este sensor precisa ser imerso água somente durante a medição e depois precisa ser armazenado em um local seco. Para suprir esta necessidade

do sensor, foi desenvolvido um dispositivo eletromecânico que desloca o sensor para dentro do recipiente de armazenamento de água e após a sua medição, automaticamente o atuador remove o sensor para fora da água, mantendo o elevador em um lugar seco.

Figura 39 – Servomotor que desloca o sensor de pH para o recipiente de água.



Autoria própria(2022)

Para regular o pH, foi utilizado uma bomba de água que vai deslocar uma solução ácida ao recipiente de água do sistema, no caso vinagre, que age como um regulador do valor de pH da água, ao gotejar este ácido no recipiente de água, um motor com uma haste de mistura em seu eixo, mistura o ácido adicionado à água, medindo novamente. A ordem dos atuadores fica nesta ordem, o sensor de pH é deslocado para a água, faz uma medição, se a água não estiver com o valor apropriado de pH, o sensor sobe novamente, a bomba com o ácido é ativada por 10 segundos, em seguida o sensor é deslocado para o tanque novamente fazendo outra medição, o processo se repete até que o valor medido de pH esteja correto para a planta.

Figura 40 – Misturador



Autoria própria(2022)

Figura 41 – Recipiente de vinagre



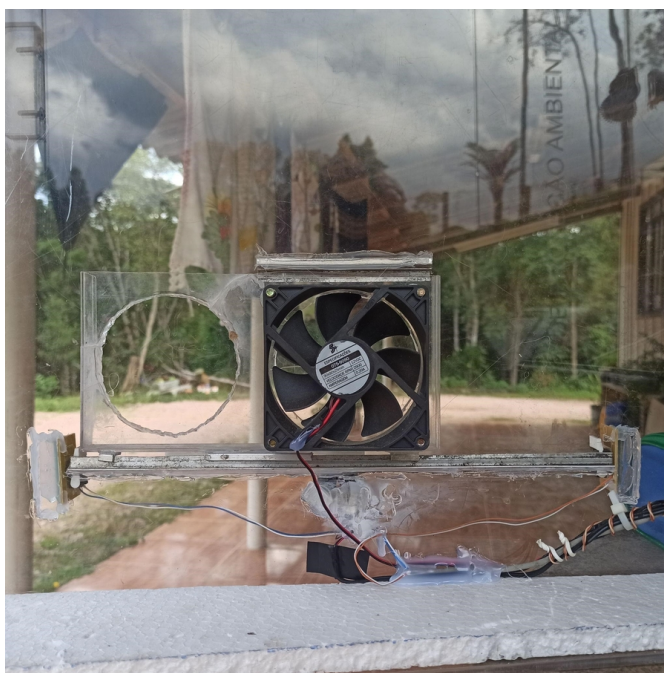
Autoria própria(2022)

3.2.2.3 Cooler

Quando dentro do sistema se encontrar uma temperatura mais elevada que a temperatura definida no *setup* do sistema, para diminuir a mesma, o *cooler* é o atuador

encarregado desta tarefa. Para que o sistema fique isolado, portas com abertura automática ficam fechadas frente ao *cooler*, quando o *cooler* precisa ser acionado, as portas abrem de forma automática, o *cooler* é acionado. Enquanto um *cooler* retira o ar quente de dentro da estufa, um segundo desloca o ar mais frio para dentro do sistema, diminuindo sua temperatura.

Figura 42 – Vista frontal do cooler e da porta



Autoria própria(2022)

Figura 43 – Vista traseira do cooler e da porta



Autoria própria(2022)

3.2.2.4 Lâmpada full espectro

A lâmpada *Full Spectrum* é uma lâmpada LED para aplicação em cultivo interno, isto significa que o dispositivo em questão fornece todos os comprimentos de onda de luz necessários para a planta. A luz da lâmpada tem todos os espectros de cor necessários para se o cultivo, desde o início até o final do período de floração da planta. O sol possui espectro de luz completo porque a luz que emite concentra todos os comprimentos de onda, que mudam com as estações do ano, na primavera e no verão os espectros são na maior parte azuis, no outono eles se tornam mais avermelhados. Razão está que a aplicação da lâmpada é mais viável neste projeto, garantindo todos os espectros de luz para a planta independente da estação do ano.

Figura 44 – Lâmpada *full* espectro

Autoria própria(2022)

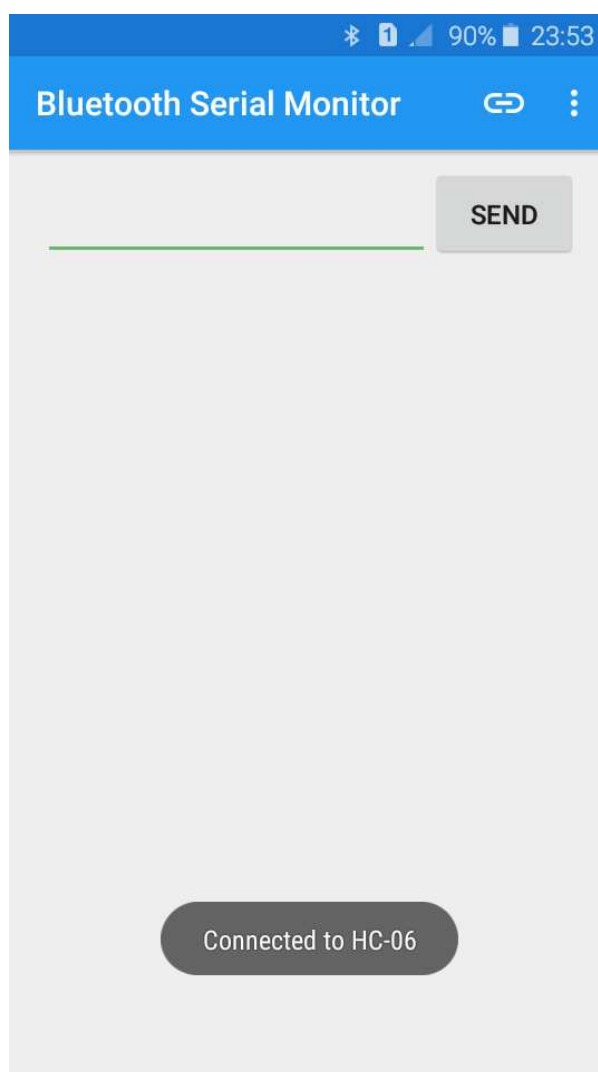
3.2.3 Interfaces Homem Máquina (IHM's)

Para estabelecer comunicação entre o usuário e nosso sistema, foi optado por utilizar duas interfaces, um aplicativo para entrada de dados via conexão bluetooth, e um *datalog* para saída de dados, que serão lidos pelo usuário no excel.

3.2.3.1 Aplicativo de comunicação serial bluetooth

Dentre as opções disponíveis na loja de aplicativo Google Play, foi escolhido o aplicativo "Bluetooth Serial Monitor" da desenvolvedora "CSA Apps" mostrado na Figura 45.

Figura 45 – Bluetooth Serial Monitor



<https://play.google.com/store/apps/details?id=com.csa.btserialmonitor>

Antes de usá-lo, também foi necessário parear o smartphone com o dispositivo bluetooth conectado ao arduino, o HC-06. Ao ligar a funcionalidade bluetooth do smartphone, o dispositivo aparece disponível para conexão como “HC-06” e possui senha padrão de fábrica “1234”.

3.2.3.1.1 Módulo Bluetooth HC-06

O módulo bluetooth HC-06 foi escolhido para fazer a comunicação entre o usuário e o arduino. Este usa dois pinos do arduino, uma para enviar e outra para receber sinal digital, isto permite que o módulo envie e receba texto, por exemplo, para o arduino.

Este módulo já é apropriado para se usar com o arduino, não necessita de bibliotecas. HC-06 é um módulo bluetooth escravo, o que exige que a comunicação seja iniciada por outro dispositivo bluetooth.

Ao ser ligado com o arduino, o módulo bluetooth fica disponível para que o usuário conecte com o smartphone, por um aplicativo apropriado para realizar esta comunicação. Como, por exemplo, o aplicativo para android “Serial Bluetooth Terminal” do desenvolvedor “Kai Morich” que está disponível na Google Play Store.

Figura 46 – Módulo Bluetooth HC-06



<https://curtocircuito.com.br/modulo-bluetooth-slave-hc06.html>

3.2.3.2 Datalogger

No código do arduino adicionamos a funcionalidade de, a cada hora, criar um novo arquivo de texto, contendo a hora atual no nome do arquivo, onde são escritas as leituras de cada sensor e atuador do sistema a cada minuto. As informações de cada minuto são escritas em uma linha, com as informações separadas por ponto e vírgula. Uma nova linha é criada para as informações do minuto seguinte.

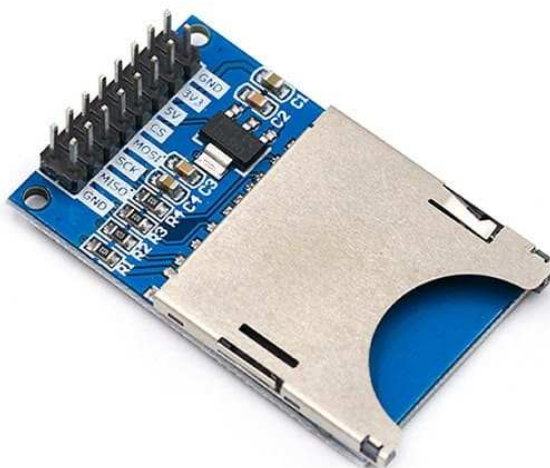
O que se espera ao retirar o cartão SD do sistema para coletar os dados, são arquivos para cada hora, cada um contendo 60 linhas de informações. Arquivos quais são importados diretamente pelo Excel para gerar planilhas e facilitar a visualização das informações.

3.2.3.2.1 Módulo Cartão SD

O módulo cartão SD adiciona ao projeto um meio para acessar um banco de dados que armazena as informações de cada planta, necessárias para ajustar os valores alvos dos controladores presentes no projeto. O uso deste módulo também permite salvar as configurações do usuário para acessá-las caso um desligamento acidental ocorra, assim como salvar as leituras dos sensores periodicamente.

Este módulo, visto na Figura 47, pode ser alimentado com 5V, mas trabalha com 3,3 V, fazendo necessária a conversão de nível lógico de 3 dos seus 4 pinos de comunicação com o arduino.

Figura 47 – Módulo Cartão SD



<https://www.filipeflop.com/produto/modulo-cartao-sd-card/>

3.2.4 Dados das plantas armazenadas no dispositivo

O banco de dados utilizado para o armazenamento das informações de cada planta é do tipo relacional, comportado em um único arquivo no formato “txt”. Onde as linhas ímpares carregam o nome de cada planta, e estas são sucedidas por um vetor que carrega todas as variáveis de sua planta. Neste exemplo um vetor de caracteres carrega o nome “Cebolinha” e a linha que o sucede contém outro vetor, que apesar de ter números, também é lida pelo arduino como um grande vetor de caracteres, este vetor é (no arduino) dividido em 9 outros vetores menores separados por espaço.

[...]

Cebolinha

18 3 35 55 18 4-14-8 60 006 kg

[...]

Os 5 primeiros (temperatura ideal; margem de temperatura (um intervalo de variação aceitável para mais ou para menos); umidade ideal; pH ideal; tempo de luz por dia) são convertidos em variáveis numéricas, a quinta variável especificamente é em seguida dividida por 10, pois este carrega o valor do pH, fazemos este processo para não precisar converter uma vírgula de caractere para ponto flutuante.

Após os 5 primeiros valores, temos espaço para um vetor de 8 caracteres (no formato XX-XX-XX, onde cada X pode ser um algarismo), este carrega a informação do tipo de adubo a ser mostrado ao usuário quando é informado o tipo da planta.

Em seguida são mais duas variáveis numéricas, e por fim mais um vetor de caracteres de tamanho indefinido que carrega um nome da unidade que informa a estimativa de produção da planta. Neste exemplo, estas três últimas informações significam que para a cebolinha, é esperada uma produção de 6 kg para cada 10 m² ao longo de 60 dias. A área de 10 m² é padrão para todas as plantas.

4 APLICAÇÃO DO PROJETO

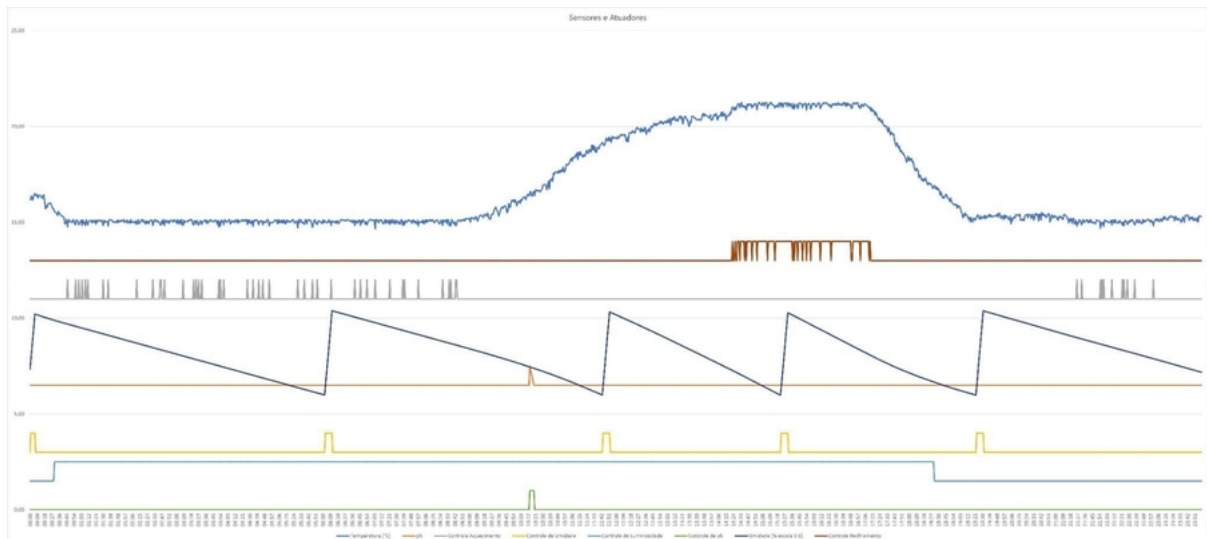
Após todas as ligações de alimentação do dispositivo, o mesmo funcionou como o esperado, na forma de uma estufa inteligente, onde os insumos necessários para a planta foram cedidos de maneira precisa, segundo o banco de dados disponível da planta no dispositivo.

Cedendo um ambiente propício para a planta, os sensores captam de forma efetiva os valores, a programação controla de forma efetiva as variáveis selecionadas no momento do *SetUp* do sistema. O dispositivo operou durante 20 dias entre os dias 17/10/2022 até 06/11/2022. Para efeitos de análise, na Figura 48, que se refere a planilha do sistema supervisor, indica a detecção dos sensores e os atuadores no decorrer de um dia.

A coleta de dados do sistema, bem como o tempo de resposta a anormalidades no mesmo, se dá a cada minuto, o que no decorrer de 24 horas em um dia, se obtém 1440 valores para cada variável. Com uma quantidade massiva de dados coletados, é possível aproximar significativamente da precisão destes valores que a planta precisa.

Para fins de exemplificação dos resultados, referente a demonstração do sistema de controle, é selecionado o dia 22/10/2022 como amostragem, indicando de forma gráfica o controle das variáveis no sistema.

Figura 48 – Gráfico com todos os acionamentos completos no dia específico



Autoria própria (2022)

A coleta de dados referente ao sistema se dá por conta de cartões SD, no sistema há dois cartões. Durante o dia, um cartão irá receber os dados do sistema a cada minuto, e outro poderá ser retirado do Arduino para a coleta de dados, pois este cartão não operante em um dia foi o que registrou os dados do sistema no dia anterior.

Os dados são registrados em forma de colunas na memória flash dos cartões, como

representado na Figura 49, tornando mais simples a transferência destes dados para a tabela em Excel responsável por gerar os gráficos.

Figura 49 – Dados gerados pelo Arduino

```
0;24;15.98;498;5;6.5;0;0;0;0
0;25;16.02;498;5;6.5;0;0;0;0
0;26;15.92;497;5;6.5;0;0;0;0
0;27;15.83;496;5;6.5;0;0;0;0
0;28;15.71;495;5;6.5;0;0;0;0
0;29;15.70;495;5;6.5;0;0;0;0
0;30;15.56;494;5;6.5;0;0;1;0
0;31;15.52;494;5;6.5;0;0;1;0
0;32;15.52;493;5;6.5;0;0;1;0
0;33;15.45;493;5;6.5;0;0;1;0
0;34;15.36;492;5;6.5;0;0;1;0
0;35;15.57;491;5;6.5;0;0;1;0
```

Autoria própria (2022)

4.1 Inicialização do procedimento de cultivo

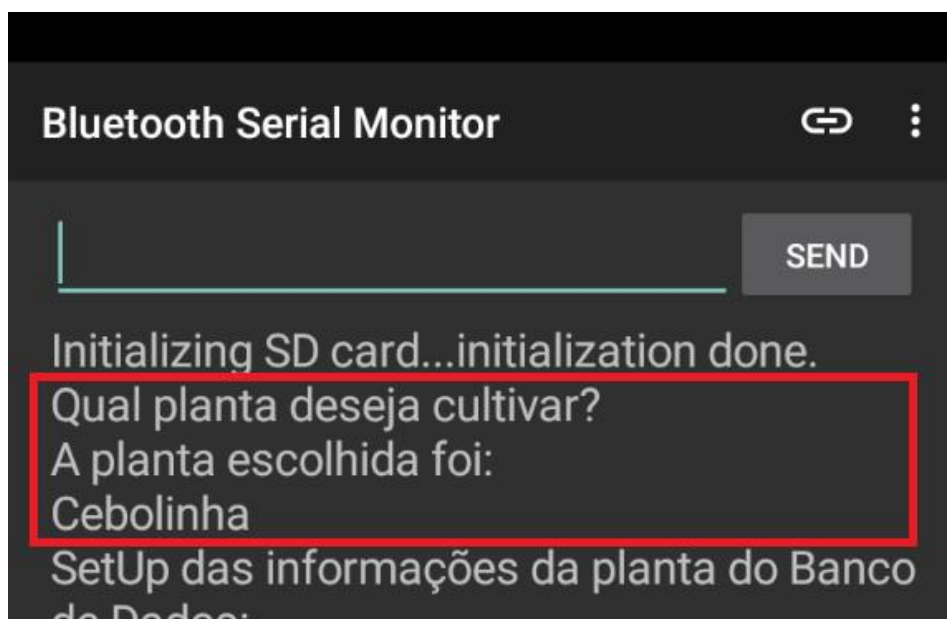
No início de sua execução é necessário que o usuário utilize seu smartphone com a função bluetooth ativada, e com uma aplicação de comunicação com bluetooth através de monitor serial. Vários aplicativos possuem esta função, o escolhido para operar o sistema foi o “Bluetooth Serial Monitor” da desenvolvedora “CSA Apps”.

A partir disto foi possível parear o smartphone ao módulo bluetooth do arduino, o qual, por padrão, é identificado como “HC-06” com senha “1234”.

Com o dispositivo pareado, foi iniciado o aplicativo de comunicação serial e pressionado o botão reset do arduino. Agora o arduino inicia o processo de setup, que começa com a comunicação com o usuário, e pergunta ao usuário qual planta será cultivada.

Ao responder “Cebolinha”, o arduino busca em seu arquivo de banco de dados disposto em seu cartão micro SD.

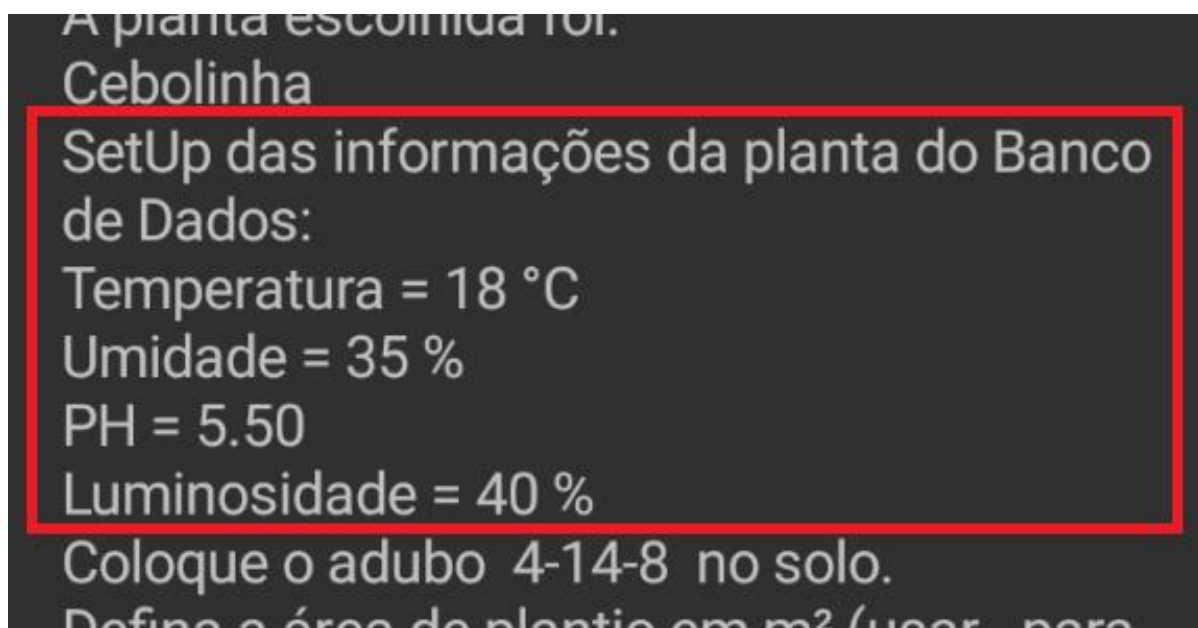
Figura 50 – Tela de seleção de planta)



Autoria própria(2022)

Ao encontrar a linha do arquivo referente ao nome de planta fornecido pelo usuário, o arduino salva em sua própria memória todas as variáveis referentes à planta selecionada (temperatura, umidade pH e tempo de luminosidade) em seguida retorna ao aplicativo que a planta foi encontrada no sistema, e mostra as variáveis referentes a planta.

Figura 51 – Tela de Amostra das variáveis da planta

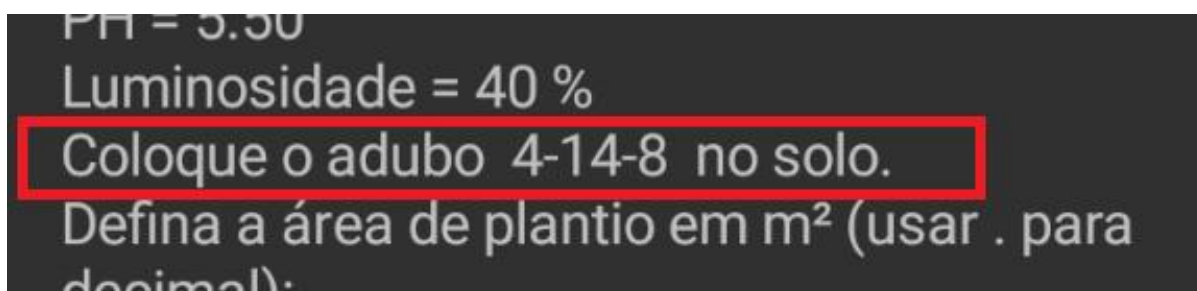


Autoria própria(2022)

Logo em seguida, na tela do aplicativo é indicado ao usuário o tipo de adubo que o usuário irá colocar no solo, solicitando uma confirmação do usuário que o mesmo colocou o

adubo no vaso da planta.

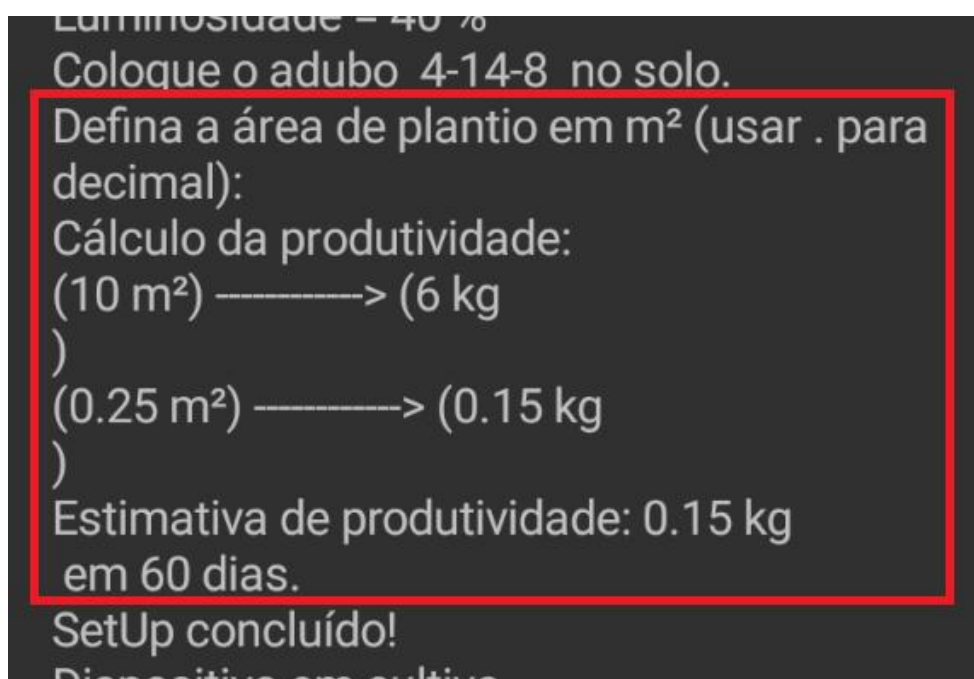
Figura 52 – Tela de solicitação de adubo



Autoria própria(2022)

Em seguida, ainda através da comunicação serial, o arduino solicita a área de plantio, solicitada em m², fazendo uso de ponto para valores decimais.

Figura 53 – Tela de definição da área de cultivo



Autoria própria(2022)

Com a informação da área de plantio e de produção estimada para 10 m² que foi extraída do banco de dados, o arduino calcula pela equação 41 a produção estimada e mostra-a ao usuário.

$$P_b = \frac{(P_a \cdot A_b)}{10} \quad (4.1)$$

A equação (número da fórmula) mostra como o arduino determina a produção estimada para a área fornecida pelo usuário. Onde P_b é a produção estimada, P_a é a produção estimada para uma área de 10 m², A_b é a área fornecida pelo usuário, e 10

representa a área, em m², onde se espera produzir uma quantidade Pa. Esta equação não considera a duração da produção, portanto esta informação é também mostrada ao usuário conforme a Figura (número da FIGURA).

Para finalizar esta etapa de comunicação com o usuário, o arduino verifica se o reservatório de água está cheio, caso não esteja, solicita que seja abastecido até que o nível alto do reservatório seja verificado pelo sensor respectivo. Se já estiver cheio apenas informa ao usuário que o *setup* está concluído e que pode se desconectar do bluetooth, e o arduino segue para o próximo passo, o qual já está incluso em seu loop principal.

4.2 Controle de pH

Como citado no capítulo 2.10.3, o pH é importante para a planta, logo, o controle dessa variável é necessário no projeto.

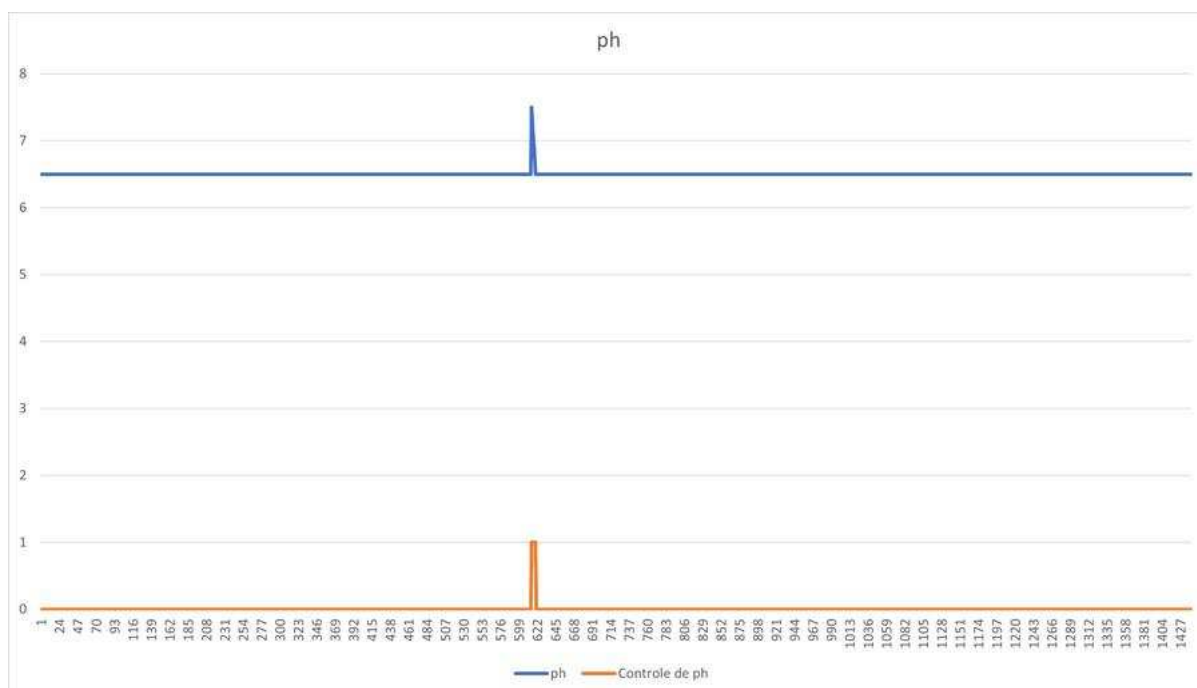
Com água presente no reservatório até o nível do sensor, o arduino dá início ao processo de controle do pH. O sistema possui um sensor de pH, para realizar a leitura, fixado em um servo motor que imerge e emerge o sensor da água. Possui também uma pequena bomba para adicionar vinagre ao reservatório de água e um motor com eixo ligado a uma pequena hélice no fundo do reservatório.

Neste processo o arduino inicia um *loop* onde o servo motor imerge o sensor na água do reservatório, que por sua vez realiza uma leitura do pH (*Present Value*) e compara com o pH ideal (*Setpoint*) da planta que está sendo cultivada, valor o qual foi anteriormente salvo em uma variável. Ao constatar que o *present value* está maior que o *setpoint*, é acionado a bomba de vinagre por um pequeno intervalo de tempo (constante, definido no código do arduino como meio segundo).

Ao desligar este acionamento da bomba, o arduino faz o acionamento do motor que mistura a solução do reservatório, desta vez por 20 segundos. Durante este tempo o arduino continua a executar suas outras funções, e no final deste tempo se reinicia o *loop* deste processo até que seja lido um *present value* menor ou igual ao *setpoint*.

Ao obter um *present value* menor ou igual ao *setpoint*, é finalizado o controle de pH, e o arduino acionar novamente o servo motor para emergir o sensor.

Figura 54 – Controle de ph



Autoria própria (2022)

Na Figura 54, se encontra o gráfico referente ao processo de controle de pH, essa medição ocorre somente no preenchimento do nível de água no reservatório, processo este que entra em operação em aproximadamente em 5 dias. Se avista no gráfico de controle do processo, que o ph se encontrou em um valor menor que o indicado em primeiro momento, ativando o processo de controle de ph, ficando operante até a valor de ph indicar o valor de SetUp preferido.

4.3 Controle de temperatura

Para realizar leituras de temperatura, este sistema utiliza o sensor DS18B20, descrito no capítulo 2.10.2. O sistema possui atuadores para o controle de temperatura nos dois extremos, um conjunto dos mesmos para aquecimento e outro para resfriamento da estufa.

Para o resfriamento da estufa temos como atuadores duas portas montadas em lados opostos na estufa, que são abertas e fechadas através de motores acionados por uma ponte H e pelo arduino, dois coolers, um frente a cada porta, acionados por um sistema de controle PID pelo arduino com uso de transistores.

Para o aquecimento da temperatura da estufa, uma lâmpada incandescente é utilizada, pois a mesma dissipa calor de forma propícia a um aumento da temperatura da estufa de forma eficaz

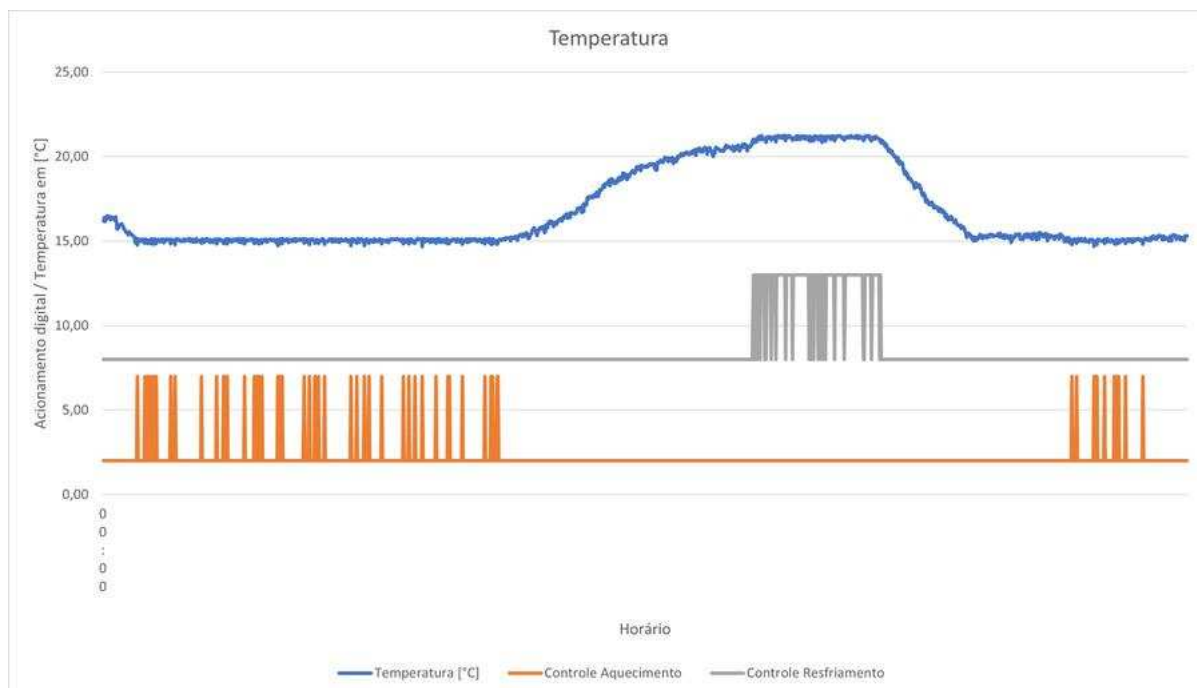
O processo de controle inicia com a medição da temperatura e a comparação deste com o valor ideal, com uma pequena margem, para evitar que as portas fiquem abrindo e fechando frequentemente.

Caso a temperatura lida (*present value*) seja maior que a soma da temperatura ideal (*setpoint*), o arduino liga o sistema de resfriamento abrindo as portas frente aos coolers acionando os mesmos em seguida. Um cooler se localiza em uma lateral superior da estufa e é responsável por retirar o ar quente de dentro dela, o outro em na lateral oposta inferior é responsável por jogar dentro da estufa o ar do ambiente, que é mais frio que o da estufa, estes acionamentos possuem um controle de velocidade por PWM do arduino, cujo *duty cycle* é a saída do sistema de controle PID da temperatura. Se o *present value* atingir valores abaixo da soma do *setpoint*, os coolers são desligados, mas as portas ainda continuam abertas.

Numa eventual redução da temperatura interna da estufa onde o *present value* atinja valores inferiores à diferença do *setpoint* com sua margem, as portas, se abertas, são então fechadas. Nesta situação, o arduino também faz o acionamento da lâmpada incandescente, dissipando o calor e aquecendo o ambiente interno da estufa até aproximar o *present value* do *setpoint*.

O gráfico mostrado na Figura 55 define a medição dos sensores durante o dia, tal como o acionamento dos atuadores para a correção das variáveis internas da estufa, referente aos valores no programa. Quando a temperatura segue entre os valores de máximo e mínimo, neste caso entre 15°C e 21,1°C, os atuadores se mantêm desligados, aproximadamente às 14:25 horas da tarde, a temperatura interna da estufa se eleva, passando do limite máximo tolerável, em seguida, podemos ver um gráfico de acionamento do sistema de resfriamento acionando a todo o momento que a temperatura ficar superior a 21,1°C. No acionamento oposto, quando a temperatura se encontra abaixo de 15°C, se observa no gráfico o período em que a lâmpada de aquecimento se mantêm ativada e a temperatura interna na estufa se elevando.

Figura 55 – Controle de temperatura



Autoria própria (2022)

4.4 Controle de luminosidade

Como citado no capítulo 2.3.4, é muito importante que a planta seja irradiada por todos os espectros de luz do sol, como já citado também, os comprimentos de onda para tais efeitos não são providos de forma eficaz no decorrer do ano, logo, a aplicação da lâmpada *Full Spectrum* foi feita ao sistema, tal qual o tempo que a planta irá ser irradiada pela luz da mesma, no *SetUp* de exemplo, a carga horária da planta no até o momento da germinação é de um período de 18 horas, recebendo todos os espectros de luz sobre o solo que ela vai germinar. Para atuar, a lâmpada *full* espectro fica simplesmente acionada durante o tempo pré-determinado, neste caso, da hora 12:30 até a hora 18:30, assim que o sistema conta o período de 18 irradiando luz, a lâmpada desliga. Não há nenhum sensor para detectar a luz desta lâmpada, já que por informações do fabricante, já é de conhecimento no projeto que a lâmpada vai suprir as necessidades de comprimentos de onda para a planta.

Na figura 56, podemos ver um gráfico em que a lâmpada *full* espectro fica ligada, seguindo do seu desligamento após atingir às 18 horas determinadas no *SetUp* do sistema.

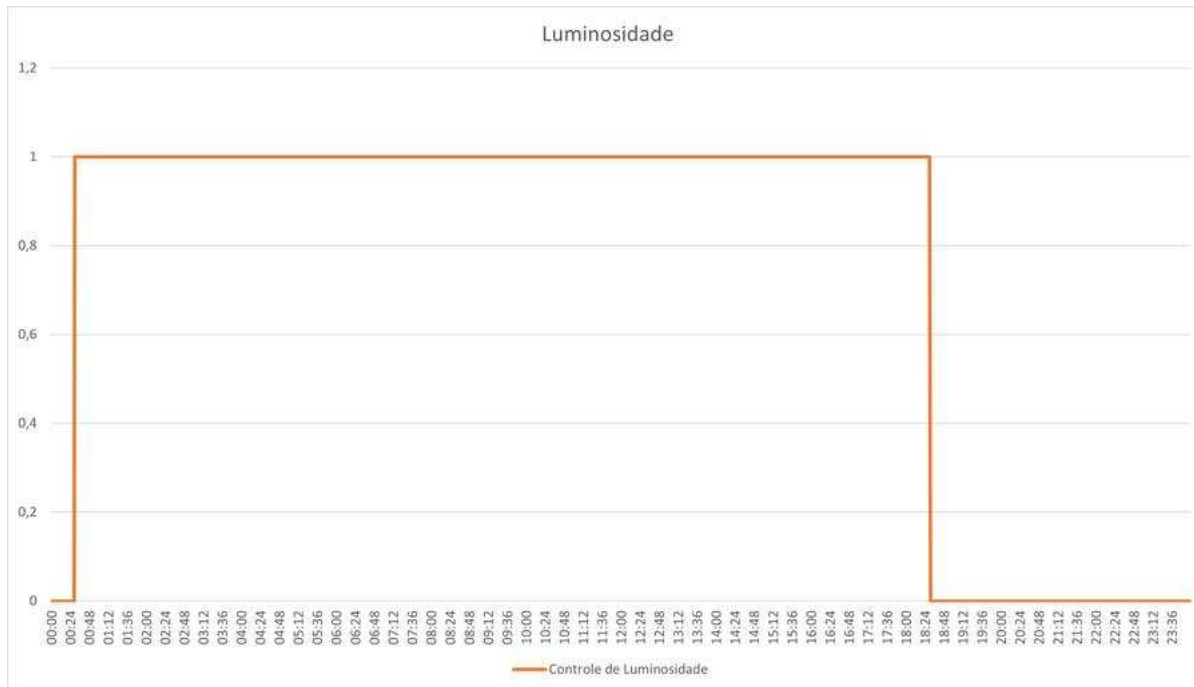
A estufa é isolada da luz do ambiente externo, para as plantas, ha faixas de luz que interferem em etapas distintas do seu desenvolvimento, com o clima externo variando muito durante o dia, a opção do isolamento da estufa corrige essa situação, com uma lâmpada LED para esse tipo de cultivo específico. A luz destas lâmpadas possuem todos os comprimentos de onda de luz que a planta necessita para desenvolver suas características.

As diferentes frequências eletromagnéticas da luz têm uma influência no desenvolvi-

mento das plantas diretamente. Fótons azuis (entre 400 e 499 nm) promovem o crescimento das plantas, promove o desenvolvimento radicular e a produção de nutrientes e estimula a produção de clorofila e a troca de gases, os fótons amarelos e verdes (entre 500 e 599 nm), permite a avaliação visual da saúde das plantas. Fótons laranja e vermelho (entre 600 e 700 nm) são eficazes para estimular o crescimento das plantas e melhora absorção da clorofila; age no controle da floração e no crescimento durante o dia e fótons vermelhos (entre 701 e 800 nm) promovem o pleno funcionamento da planta e aumenta a eficiência fotossintética da mesma.

As plantas não necessitam de luz do sol durante todo o dia, tanto o excesso quanto a falta de sol são prejudiciais à mesma, assim, como visto na (referência tal), cada planta necessita de horas de sol dependendo do seu estágio de desenvolvimento, no caso da cebolinha, este período é de 18 horas de sol na fase de eclosão das sementes e germinação, visto na Figura 56, e no momento da sua floração, esse período de luz do sol é de 12 horas diárias. No banco de dados da cada planta, há a variável que mostra a hora de sol que a planta precisa, no caso, o tempo que a lâmpada de LED irá ficar ligada dentro da estufa, fazendo assim com que a planta obtenha a luz necessária para seu desenvolvimento de maneira correta, sem alterações externas, como clima nublado ou sol em excesso.

Figura 56 – Controle de luminosidade



Autoria própria (2022)

4.5 Controle de umidade

O atuador cede água a planta definindo da umidade o faz por gotejamento na terra em que se encontra a planta. No banco de dados de cada planta, está armazenada a informação do valor de umidade necessária para seu melhor desenvolvimento, no caso da cebolinha, segundo a (Referência livro biologia aqui), este valor é de 50% de umidade presente na terra, assim.

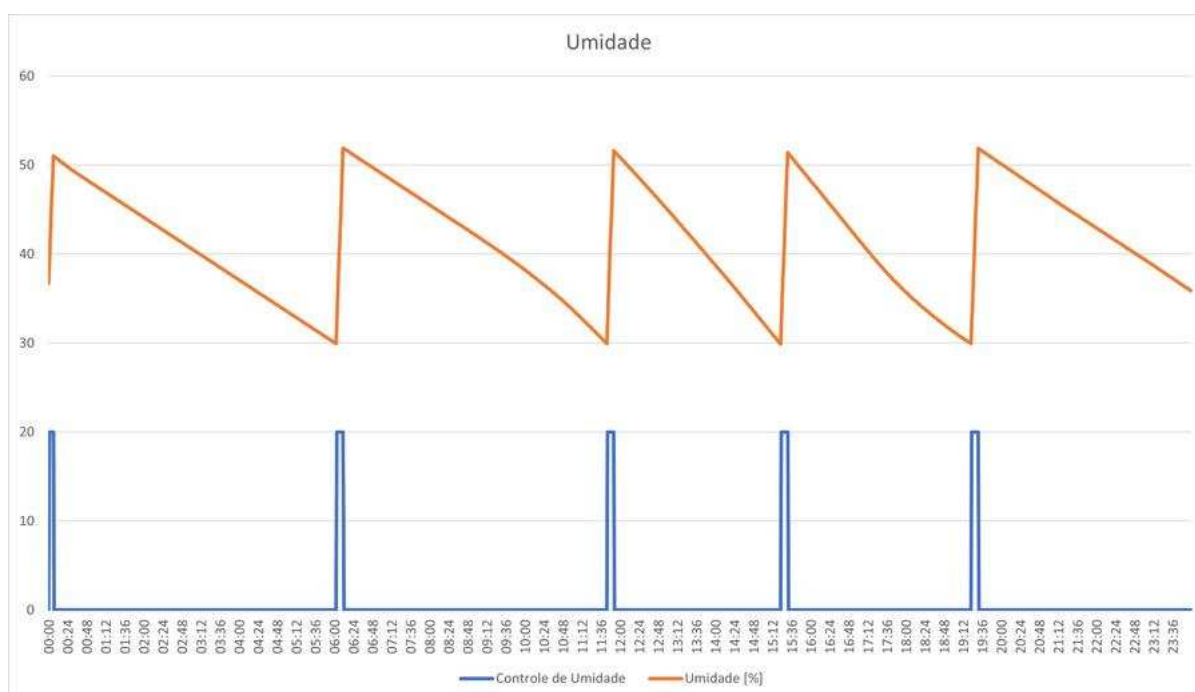
Medindo o valor de umidade com um sensor referente a ela introduzido na terra que se encontra a planta, o sistema identifica umidade a cada minuto, mantendo um valor de SP em 50% deste valor.

Se o valor de umidade no sistema (*present value*) for inferior ao valor de SP (*setpoint*), o arduino liga o sistema de irrigação, ligando a bomba de água levando a mesma até a terra por gotejamento, esse acionamento da bomba dura 30 segundos, após isso ela desliga, se na próxima medição de umidade estiver abaixo de 50% a bomba liga novamente por 30 segundos, este procedimento se repete até o sensor de umidade detectar um valor dessa variável igual ou superior a 50%.

A figura 57 mostra um gráfico da relação entre o sensor de umidade com o atuador para regular o valor da mesma, esta é a interação mais complexa que o sistema mostrou, pois são vários fatores que influenciam na diminuição de umidade na terra, diferente das outras variáveis em que somente um valor influencia no mesmo.

A princípio se nota que quanto maior a temperatura na estufa, visando a temperatura máxima em SP, 21,1 °C, mais rápido é o consumo de água, diminuindo a umidade mais rapidamente, precisando de uma reposição em maior número de vezes. No período subsequente em que a lâmpada de *Full Spectrum* é desligada, a umidade na terra diminui mais lentamente, isso se deve pelos fatores que a estufa está mais fria, mais próximo de 15 °C, e a luz, junto com a água são insumos necessários para a nutrição da planta, sem a luz, o consumo de água é diminuto pela planta, se elevando novamente no período em que a lâmpada *Full Spectrum* é acionada.

Figura 57 – Controle de umidade



Autoria própria (2022)

4.6 Resultado do desenvolvimento da planta

Como pesquisa e especificado no referencial teórico, a cebolinha é uma hortaliça que possui um período de germinação de 15 dias em condições normais de cultivo. Como o cultivo se sucedeu desde a semente no processo, a análise avaliou duas plantas, a Planta 1 estava suscetível ao dispositivo e a Planta 2 cultivada com o método convencional, em local aberto e sendo regada uma vez ao dia, sem uma quantidade de água exata no momento de sua irrigação.

No 13º dia desde o funcionamento do sistema, foi notado que a Planta 1 dentro da estufa germinou e que a Planta 2 não havia germinado ainda.

Com esta observação, optou-se por continuar os testes da mesma maneira, assim, no 15º dia a Planta 2 germinou, e a Planta 1 havia se desenvolvido um pouco a mais.

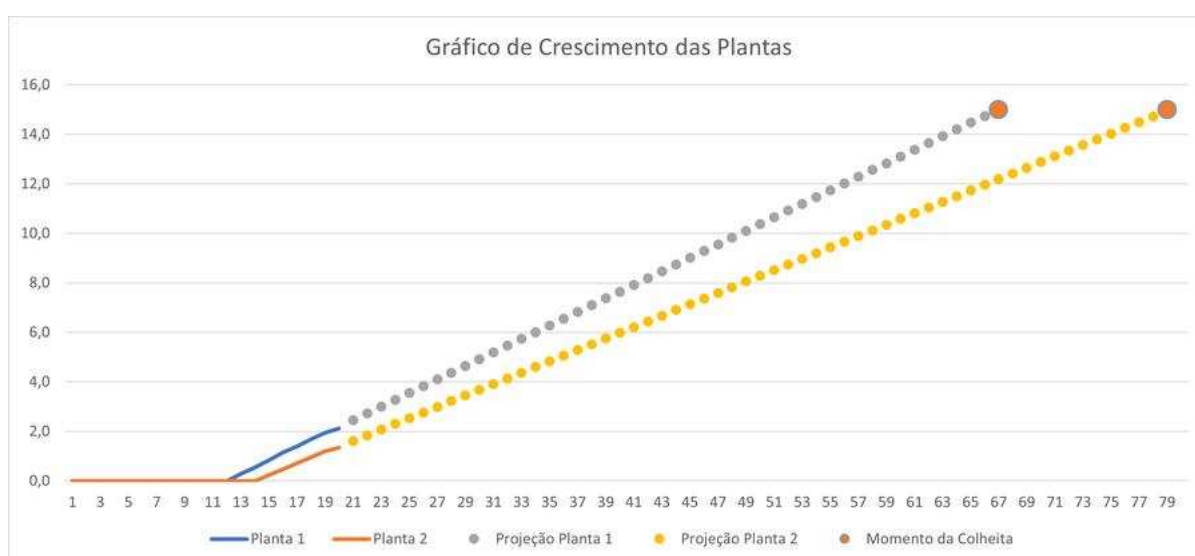
Apesar do dispositivo se aplicar a todo o desenvolvimento da planta até a colheita, já se pode obter um resultado de que a Planta 1 sujeita ao sistema, germinou primeiro por ter seus insumos cedidos de forma mais precisa e que eventualmente, a falta de precisão na quantidade dos insumos da Planta 2 acarretaram em um período maior de germinação.

Comparando o prazo de germinação das duas plantas, a Planta 1 em 13 dias e a Planta 2 em 15 dias, o tempo da Planta 1 sobre a outra reduziu em 15%. Apenas dois dias a mais a curto prazo são mínimos, porém, a soma destes dias a menos de colheita a longo

prazo, pode acarretar em um aumento no número de safras em um ano. O usuário também pode garantir a colheita, pois com este sistema, a planta não estará sujeita a condições climáticas adversas, tal qual como pragas ou doenças.

O dispositivo permite o usuário a planejar de forma mais efetiva a safra que irá colher, o quanto vai colher e quando vai colher, e a planta em questão pode ainda obter uma melhora em qualidade outros atributos, como visível nos testes o tempo de germinação e a altura maior da planta no mesmo prazo que a planta cultivada em método convencional, seguindo na Figura 58, foi elaborado uma projeção de safra para a Planta 1 em comparação com Planta 2.

Figura 58 – Projeção de crescimento das plantas de teste



Autoria própria (2022)

5 CONCLUSÕES E CONSIDERAÇÕES FINAIS

O dispositivo funcionou de forma muito próxima do planejado, coletando os dados da forma que se planejou no projeto. Os componentes eletrônicos funcionaram de forma correta até o fim do processo. Apesar do tempo de cultivo ser curto, se pode colher um resultado muito relevante, que foi o tempo de germinação da planta que diminuiu cerca de 15% comparando com os métodos convencionais, germinando em aproximadamente 13 dias, o que se mostrou satisfatório a longo prazo, reduzindo o tempo de safra para a planta.

Os atuadores foram acionados de maneira precisa a programação, com os empilhados de precisão, pois eles sendo acionamentos mecânicos, são plausíveis de ajustes de precisão e pequenos defeitos e desgastes físicos no decorrer de vários acionamentos sucessivos.

5.1 Microcontrolador

Como o projeto possui uma alta complexidade, foi preferível utilizar o controlador ARDUINO, para que o mesmo funcionasse como hardware principal do dispositivo, sendo o responsável por armazenar o programa principal e aplicar seu funcionamento, acionando as entradas para os acionamentos pertinentes.

O ARDUINO UNO foi a primeira escolha de dispositivo, pois como a equipe já dispunha de um deste, optou-se por utilizá-lo. Nas primeiras etapas do projeto, o dispositivo resolveu todas as necessidades do projeto, já que o mesmo possui uma memória muito extensa até para um projeto desse porte. Com o passar do tempo, com inúmeras modificações do projeto, o ARDUINO UNO ficou obsoleto em relação ao seu hardware, já que o mesmo não possuía o número de entradas digitais para os sensores e outros dispositivos adicionados à medida que sua necessidade aparecia.

Então o ARDUINO MEGA substituiu o antigo, pois o mesmo contém um número bem maior de entradas digitais, porém esta alteração acarretou uma significativa mudança na programação. Todas as pinagens para a conexão dos dispositivos precisaram ser refeitas no programa principal, tal como mudanças em determinadas linhas nos códigos e a alteração de pinos cuja aplicação difere entre os dois dispositivos. O ARDUINO MEGA também possui algumas funções melhoradas nas quais foram opções para o projeto, como entradas digitais específicas para leituras de determinados sensores também utilizados no projeto.

5.2 Estrutura mecânica

No planejamento foi definido como seria a estrutura da estufa onde seria cultivada a planta. Para um controle preciso, a mesma precisaria ser isolada o máximo possível do ambiente externo, para garantir o controle do ambiente interno, maneira essa muito eficiente para o controle do processo. Foi adquirida uma caixa de acrílico propícia a este isolamento,

porém, a mesma precisaria que fossem conectados componentes pertinentes ao projeto, como coolers, sensores, entre outros. A fixação destes componentes, precisou de um bom planejamento, pois em caso de falha, a estrutura poderia ser comprometida com furos e rasgos desnecessários, deixando a estufa não isolada das condições externas do ambiente.

O cooler para o controle de temperatura, é um dispositivo conectado à parte interna da estufa que obrigatoriamente precisa também ter acesso ao ambiente externo, para assim, um cooler coletar o ar mais frio do ambiente e outro liberar o ar quente no ambiente externo, assim, diminuindo a temperatura caso isso fosse preciso. Para quando os coolers não estivessem operando, e mesmo assim manter o ambiente interno da estufa isolado, optou-se por desenvolver uma porta de correr frente aos cooler, onde a mesma seria fechada frente ao cooler, mantendo esse isolamento.

A precisão de funcionamento dos componentes mecânicos, assim como a porta que fecha o acesso ao cooler para o ambiente externo, foram bem complexos, pois como se tratava de componentes relativamente pequenos e os mesmos precisam funcionar durante longos períodos de tempo, em relação ao tempo de colheita, a mecânica de tais componentes ficava muito prejudicada no decorrer do processo. Problemas de desgaste, deterioração das fixações de tais componentes e a precisão dos mesmos em suas atividades foram os grandes problemas da parte física do projeto, vários testes com vários materiais e componentes foram feitos atrasando a construção da estufa. Foi projetado também de maneira bem sucinta os locais onde os cabos de transmissão de energia e insumos iriam entrar para dentro da estufa, como são vários, o isolamento destes cabos, tubos e conexões foram aplicadas. Estes problemas foram enfrentados pois a construção foi feita totalmente de forma artesanal, com certa precisão compatível com o que se tinha no momento. O isolamento da estufa também não foi feito de maneira muito precisa, porém, o sistema de controle do dispositivo foi desenvolvido para suprir esta deficiência, controlando imediatamente as variáveis em caso de ocorrência de alguma alteração abrupta.

O sistema de controle de temperatura tem uma limitação no quesito de diminuir a temperatura interna, pois depende de ar frio que vem de fora. Assim, em dias quentes não será possível resfriar tanto quanto o necessário.

Sugerimos como trabalho futuro adaptar este projeto para adicionar placa Peltier ao sistema de refrigeração, substituindo ou não a ventilação.

A calibração do sensor de pH indicada pelo fornecedor está descrita no capítulo 2.10.3, onde é recomendado utilizar os dois trimpots disponíveis no módulo do sensor. Tentamos, mas sem êxito em calibrar, concluímos que havia defeito. Nesta calibração, é como se um trimpot fosse uma constante que multiplica o valor lido, e o outro trimpot, uma constante que se soma ao produto obtido. Implementamos o mesmo processo no código do arduino, e variamos, com cuidado e paciência, cada uma das constantes, e com um medidor de pH manual (devidamente calibrado por um técnico agrônomo) para nos auxiliar nesta calibração.

5.3 Sugestões para trabalhos futuros

O projeto se fez muito dinâmico, já que são várias aplicações na agricultura, algumas ideias de aplicações em plantas foram sugeridas e exploradas para o projeto, porém, em relação ao prazo de entrega, os recursos disponíveis e complexidade das aplicações, foram somente escolhidas as atividades mais importantes para a melhoria do desempenho da planta, deixando de fora várias aplicações que seriam interessantes ao projeto. Ideias estas que são totalmente cabíveis para uma possível melhoria do projeto para uma efetiva aplicação e prosseguimento do projeto além do trabalho de conclusão de curso. Citando algumas temos:

O controle de pragas através do processamento de imagens, onde o dispositivo poderia gravar a planta em período integral, identificando por imagens o aparecimento de pragas como insetos, fungos e outras anomalias perceptíveis por uma identificação visual.

Controle dos macronutrientes do solo através de sensores, onde o dispositivo mede os valores de nutrientes necessários para a planta e altera os valores dos mesmos automaticamente, misturando os insumos necessários na terra através de atuadores.

Cobertura automática da estufa com uma cobertura para inibir a luz do sol projetada na planta de maneira intensa, fazendo com que atuadores fossem acionados puxando a cobertura por cima da estufa.

Sensores de medição do tamanho das plantas, tal como a dimensão de suas raízes, aplicação simples, porém a estrutura física da estufa precisava ser reavaliada por conta do funcionamento de sensores como o ultrassônico.

Em relação à programação também havia a medição da planta em tempo real, onde poderia se atualizar em tempo real a data de colheita da planta, ajudando o operador a prever a safra de maneira mais eficiente.

Aplicação de sensor de fluxo de seiva, para medir mais precisamente as condições morfológicas da planta durante a safra, onde um dispositivo para tal medição para plantas de pequeno porte teria que ser projetado para tal aplicação.

Sugerimos também realizar a adição de uma placa peltier ao sistema de controle de temperatura, substituindo ou não o sistema de ventilação. Pois até então, a ventilação depende da temperatura externa para diminuir a interna, sendo assim um fator muito limitante ao controle de temperatura.

REFERÊNCIAS

- AMARO, G. B. *et al.* Recomendações técnicas para o cultivo de hortaliças em agricultura familiar. **Embrapa Hortaliças-Circular Técnica (INFOTECA-E)**, 2007.
- ARDUINO. **Visão Geral Arduino Mega 2560**. 2016. Disponível em: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560#>. Acesso em: 09/11/2016.
- COELHO, E. F.; COELHO FILHO, M. A.; OLIVEIRA, S. L. de. **Agricultura irrigada: eficiência de irrigação e de uso de água**. [S.l.]: Bahia Agrícola, v. 7, n. 1, p. 57 - 60, 2005. Acesso em: 2021.
- CURY, J. E. R. Teoria de Controle Supervisório de Sistemas a Eventos Discretos. In: V SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 2001, Canela, RS. Canela, RS, 2001.
- FILGUEIRA, F.A.R. Novo manual de olericultura: agrotecnologia moderna na produção e comercialização de hortaliças. Viçosa., n. 3, p. 421 –, 2008.
- “FOTOPERIODISMO” em Só Biologia. Virtuoso Tecnologia da Informação, 2008–2022. Disponível em: https://www.sobiologia.com.br/conteudos/Morfofisiologia_vegetal/morfovegetal34.php. Acesso em: 13/11/2022 às 21:41.
- GIMENEZ, S. P. **Microcontroladores 8051 Teoria e Prática**. [S.l.]: Editora Erica, 2010.
- HAMMERSCHMIDT, R. **O que é memória FLASH**. 2012. Acesso em 23 de Abr. 2016. Disponível em: <http://www.tecmundo.com.br/hardware/198-o-que-e-memoria-flash-.htm>.
- HAUPT, A. G.; DACHI, E. P. **Eletrônica Digital**. [S.l.]: Edgard Blücher Ltda., 2018.
- KANEKO, M. G.; CARRIJO, O. A.; JUNQUEIRA, A. M. R. **Produção de coentro e cebolinha em substratos regionais da Amazônia à base de madeira em decomposição (PAÚ)**. 2006. Dissertação (Mestrado). Disponível em: <http://repositorio.unb.br/handle/10482/5071>.
- LOTUFO, F. A. U.; LUCENA, S. E. de U. **Desenvolvimento de um sensor virtual para processos não-lineares e variantes no tempo, com aplicação em planta de neutralização de PH**. 2010. Tese (Doutorado) — Universidade Estadual Paulista (UNESP). Disponível em: <http://hdl.handle.net/11449/105335>.
- LUCENA, L. P. de *et al.* Avaliação multicriterial das fazendas verticais canadenses como modelos sustentáveis de agricultura urbana. RAI Revista de Administração e Inovação, v. 11, n. 1, p. 181 – 202, 2014. Acesso em: 2021.
- NATÁRIO, M. C. **Confúcio: da literatura à sabedoria**. [S.l.]: Portugal-Goa: os Orientes e os Ocidentes: Seminário Internacional de Filosofia e Literatura= Portugal-Goa: The East (s) and the West (s): International Seminar on Philosophy and Literature, 2019.
- NISE, N. S. **Engenharia de Sistemas de Controle**. 6. ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos Editora Ltda, 2012.
- NORONHA, F. de A. F. *et al.* **Bombeamento hidráulico a jato com sucção biofásica**. 1995. Dissertação (Mestrado) — Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica e Instituto de Geociências. Disponível em: <http://libdigi.unicamp.br/document/?code=vtls000114871>.

- OGATA, K. **Engenharia de Controle Moderno**. Rio de Janeiro: Pretice/ Hall do Brasil, 1982.
- OGATA, K. **Engenharia de Controle Moderno**. 5. ed. São Paulo: Pearson Education do Brasil, 2010.
- PANDORFI, H. *et al.* Microclima de abrigos escamoteadores para leitões submetidos a diferentes sistemas de aquecimento no período de inverno. **Revista Brasileira de Engenharia Agrícola e Ambiental**, scielo, v. 9, p. 99 – 106, 03 2005. ISSN 1415-4366. Disponível em: <http://www.scielo.br/scieloOrg/php/articleXML.php?lang=pt&pid=S1415-43662005000100015>.
- RAMOS NETO, J. O. Guia para produzir uma horta orgânica suspensa com materiais recicláveis. **Revista Brasileira da Educação Profissional e Tecnológica**, v. 2, n. 21, p. e11379 –, 2021. Disponível em: <https://www2.ifrn.edu.br/ojs/index.php/RBEPT/article/view/11379>. Acesso em: 2021.
- SANTOS, R. F. dos. **Tecnologia no agronegocio - Equipamentos autonomos**. 2020. Disponível em: <https://blog.aegro.com.br/tecnologia-no-agronegocio/#2-Tecnologia-no-agronegocio-Equipamentos-autonomos>. Acesso em: 04/11/2021.
- SILVA, F. C. de S. *et al.* **Mensuração direta do pH do solo em campo por meio de eletrodos íon-seletivos**. 2013. Dissertação (Mestrado) — Universidade de São Paulo. Disponível em: <http://www.teses.usp.br/teses/disponiveis/11/111152/tde-19122013-103919/>.
- SOBREIRA FILHO, M. G. Manual de cultivo das hortaliças em horta doméstica, educativa e comunitária. 2012. Acesso em: 01/10/2021.
- TERMOQUÍMICA: entenda a troca de calor nas reações nesse resumo! 05/07/2020. Artigo. Disponível em: <https://www.stoodi.com.br/blog/quimica/termoquimica/>. Acesso em: 10/08/2020.
- VISIOLI, A. **Prático de Controle PID** . [S.l.]: Springer Science & Business Media, 2006. v. 3. P. 310.
- WANG, G.; WANG, L. The Bang-Bang principle of time optimal controls for the heat equation with internal controls. **Control Letters**, v. 56, n. 11-12, p. 709 – 713, 2007. Disponível em: <http://dx.doi.org/10.1016/j.sysconle.2007.06.001>.
- ZAMBROZI JUNIOR, P. *et al.* **Sensor de umidade microeletronico baseado e capacitor com dieletrico higroscopico**. 2005. Dissertação (Mestrado) — Universidade Estadual de Campinas . Faculdade de Engenharia Eletrica e de Computação. Disponível em: <http://libdigi.unicamp.br/document/?code=vtls000374592>.

Anexos

ANEXO 1 - CÓDIGO DO ARDUINO EM C++

```

//#include <millis.h>
#include <SPI.h>
#include <SD.h>           // SD
#include <Servo.h>

#include <Crescer.h> // Biblioteca de tempo sem delay
Tempora minutoEscreverSD; // Declaração da variável no biblioteca Cescer
Tempora phPassoServo; // Declara variável de intervalo para o Servo motor movimentada
Tempora phControlando; // Tempo para conferir o ph enquanto ele é controlado
//Tempora testeNoite; // Declara variável de intervalo de tempo para testar se anoiteceu
// Millis
unsigned long tempo_controle_temperatura = 0;

// ENTRADAS
#include <OneWire.h>           // inTemperatura
#include <DallasTemperature.h>
#define inTemperatura 40

// the hardware CS pin (10 on most Arduino boards,
// 53 on the Mega) must be left as an output or the SD
// library functions will not work.
#define aguaReabastecida 10 // Sensor que detecta presença de água no recipiente para
const int CSpin = 53;
int cartao = 0;
int contagem_vetores = 0; // Conta minutos até completar 1 dia
bool criar_vetor = false; // se true, está na hora de criar vetor; isso evita que fique muito grande
bool tempo_de_escrever = false; // bateu 1 min, tá na hora de escrever
bool escreveu = true; // deu certo de conectar ao SD e escrever, pode parar de tentar
bool sd_ou_micro = true; // se true, liga led do SD; se false, liga led do MicroSD
#define LigarSD 20 // Alterar aqui a porta do acionamento do SD
#define LigarSD_led 21 // Led que mostra que o SD está sendo usado
#define LigarMicroSD 39 // 22 // Alterar aqui a porta do acionamento do MicroSD
bool AUX_LED_SD = false;
#define LigarMicroSD_led 23 // Led que mostra que o MicroSD está sendo usado
String vetor = ""; // holds the data to be written to the SD card
//float sensorReading1 = 0.00; // value read from your first sensor
//float sensorReading2 = 0.00; // value read from your second sensor

```



```
//float sensorReading3 = 0.00; // value read from your third sensor
File sensorData;
//File bancoDados;

#define inUmidade A0          // inUmidade
int analogUmidadein;
#define inOutsideUmidade A3 // TIRAR
int analogUmidadeout;       // TIRAR
#define inLuminosidade A2    // inLuminosidade
int analogLuminosidadein;
#define inOutsideLuminosidade A4 // TIRAR
int analogLuminosidadeout;   // TIRAR
int lumiNoite[] = {0,0,0,0,0}; // "Nos ultimos 5 minutos, foi lido 0 de iluminação"
bool noite = false; // Inicia como dia.
bool auxNoite = false;
bool auxNoiteTrocar = false;
bool umidControl = false; // Indica que controle de umidade está acontecendo
bool lumiControl = false; // Indica que controle de luminosidade está acontecendo
#define inPh A1              // inPh
double analogPh;
double Vph;
float phin;
bool phControlInit = false; //
bool phControl = false;    //
bool moveDown = false;     // 5 Booleanas de etapas do processo de controle de ph
bool phMedir = false;      //
bool moveUp = false;       //
//float phout;
int numLoop = 0; // Conta minutos até 60 e depois reseta
int numArquivo = 1; // Conta horas até 24 e depois reseta
String nomeArquivo="";
int phServoPos;
int phServoPosOutWater = 0; // <<-- ALTERAR AQUI ANGULOS DO SERVO
int phServoPosInWater = 180;
int histWater[] = {0,0,0,0,0}; // 5 min para armazenar um breve histórico de presen
float phVet[31]; // Vetor que carrega as 30 leituras de ph mais recentes
// Se alterar tamanho do vetor, alterar também na leitura de ph e no controle de ph

int i; // Usado como iteração em diversos loops
```

```
// SAÍDAS
Servo phServo;
int delay_portas = 2000; // [ms]
#define PortaAbrir 3          // PortaAbrir
#define PortaFechar 4        // PortaFechar
#define fdc_aberto_cima 11    // Fim de curso da porta de cima aberta
#define fdc_fechado_cima 12   // Fim de curso da porta de cima fechado
#define fdc_aberto_baixo 13   // Fim de curso da porta de baixo aberta
#define fdc_fechado_baixo 14  // Fim de curso da porta de baixo fechado
#define Lampada 15           // Lampada
// Porta 53 para o SD
#define BombaDagua 5
#define ElevadorSensorPh 6 // definido embaixo
#define AdicionarVinagre 7
#define MisturadorVinagre 8
#define Coolers 9

// BANCO DE DADOS
int idealUmid; // Umidade
int margemUmid = 20; // Alterar aqui a margem de folga de umidade <---
int loop_umidade;
int idealTemp; // Temperatura
int margemTemp;
float TemperaturaAtual = 0;
bool portas = false; // Portas fechadas
double K_p, K_i, K_d; // Ganhos de PID
double PID = 0;
double temperatura_old;
double erro;
double integral, integral_old;
double diferencial;
double Ts; // Converte tempo de [ms] para [s]
int idealLumi; // Luminosidade
int margemLumi = 0; // Alterar aqui a margem de folga de luminosidade <---
float idealPh; // O controle deve atingir um valor abaixo deste // ph
bool Aux_break = false;

OneWire oneWire(inTemperatura);
```

```
DallasTemperature sensors(&oneWire);
DeviceAddress insideTemperatura = { 0x28, 0xFF, 0x64, 0x1D, 0xF4, 0x68, 0x47, 0x19
// 0x28, 0xE1, 0x20, 0x9D, 0x04, 0x00, 0x00, 0x1A };
//DeviceAddress outsideTemperatura = { 0x28, 0xE5, 0x82, 0x9D, 0x04, 0x00, 0x00, 0x

// COMEÇA BANCO DE DADOS (pré Setup)
Sd2Card card;
const int chipSelect = 53;
File myFile;
//Serial1
String nomePlanta; // String, retornada pelo usuário
float areaPlantio = 0; // Float, retornado pelo usuário
String vet_data;
//int idealTemp;
//int margemTemp;
//int idealUmid;
//int idealLumi;
//float idealPh;
String adubo;
int tempoCultivo;
int prodArea;
String unidade;
// TERMINA BANDO DE DADOS (pré Setup)

void setup(){

// COMEÇA BANCO DE DADOS -----
Serial1.begin(9600);
while (!Serial1) {
    ; // wait for Serial1 port to connect. Needed for native USB port only
}
Serial1.print("Initializing SD card...");
if (!SD.begin(chipSelect)) {
    Serial1.println("initialization failed!");
    while (1);
}
Serial1.println("initialization done.");
myFile = SD.open("banco.txt");
```

```
if (!myFile) {
  Serial1.println("error opening test.txt");
}
//String var;
Serial1.println("Qual planta deseja cultivar?");
while(nomePlanta == ""){nomePlanta = Serial1.readStringUntil('\n');delay(10);}
while (nomePlanta != vet_data){
  vet_data = myFile.readStringUntil('\n');
  vet_data.remove(vet_data.length()-1); // Remove ultimo caractere \n
  if (vet_data == "fim"){
    for (int i = 0; i < 30; i++){ // "limpar" tela
      Serial1.println();
    }
    Serial1.println("Planta não encontrada.");
    Serial1.println("Certifique-se de que digitou corretamente,");
    Serial1.println("ou solicite para que esta planta");
    Serial1.println("seja adicionada ao banco de dados.");
    Serial1.println();
    Serial1.println("Qual planta deseja cultivar?");
    myFile.seek(0); // Retorna ao inicio do arquivo para recomeçar a busca
    nomePlanta = "";
    while(nomePlanta == ""){nomePlanta = Serial1.readStringUntil('\n');delay(10);}
  }
}
Serial1.println("A planta escolhida foi:");
Serial1.println(vet_data);
vet_data = myFile.readStringUntil('\n');
//Serial1.println(vet_data); // Printa vetor de variáveis que sucede nome da plant
vet_data = vet_data.substring(2);

String varInt = vet_data;
varInt.remove(2);
vet_data = vet_data.substring(3);
idealTemp = varInt.toInt();
//Serial1.println(idealTemp);

varInt = vet_data;
varInt.remove(1);
vet_data = vet_data.substring(2);
```

```
margemTemp = varInt.toInt();
//Serial1.println(margemTemp);

varInt = vet_data;
varInt.remove(2);
vet_data = vet_data.substring(3);
idealUmid = varInt.toInt();
//Serial1.println(idealUmid);

varInt = vet_data;
varInt.remove(2);
vet_data = vet_data.substring(3);
idealLumi = varInt.toInt();
idealPh = 0.1*idealLumi;
//Serial1.println(idealPh);

varInt = vet_data;
varInt.remove(2);
vet_data = vet_data.substring(3);
idealLumi = varInt.toInt();
//Serial1.println(idealLumi);

adubo = vet_data; // String
adubo.remove(8);
vet_data = vet_data.substring(9);
//Serial1.println(adubo);

varInt = vet_data;
varInt.remove(2);
vet_data = vet_data.substring(3);
tempoCultivo = varInt.toInt();
//Serial1.println(tempoCultivo);

varInt = vet_data;
varInt.remove(3);
vet_data = vet_data.substring(4);
prodArea = varInt.toInt();
//Serial1.println(prodArea);
```

```
unidade = vet_data; // String
unidade.substring(1);
//vet_data = vet_data.substring(3); // Processo finalizado
//Serial1.println(unidade);

// PRINT DO SETUP
Serial1.println();
Serial1.println("SetUp das informações da planta do Banco de Dados:");
Serial1.print("Temperatura = ");
Serial1.print(idealTemp);
Serial1.println(" °C");
Serial1.print("Umidade = ");
Serial1.print(idealUmid);
Serial1.println(" %");
Serial1.print("PH = ");
Serial1.println(idealPh);
//Serial1.print("Luminosidade = ");
//Serial1.print(idealLumi);
//Serial1.println(" %");
Serial1.println(); // Pula linha

//String adubo = "4-14-8"; // String do SD
Serial1.print("Coloque o adubo ");
Serial1.print(adubo);
Serial1.println(" no solo.");
Serial1.println();

Serial1.println("Defina a área de plantio em m2 (usar . para decimal):");
while(areaPlantio == 0.00 || areaPlantio == -1.00){areaPlantio = Serial1.parseFloat();
Serial1.println("Cálculo da produtividade:");
Serial1.print("(10 m2) -----> (");
//String prodArea = "75"; // Inteiro, do SD
//String unidade = "kg"; // String, do SD
//int intProdArea = prodArea.toInt();
//int intAreaPlantio = areaPlantio.toInt();
Serial1.print(prodArea);
Serial1.print(" ");
Serial1.print(unidade);
Serial1.println(")");
```

```
Serial1.print("(");
Serial1.print(areaPlantio);
Serial1.print(" m2) -----> (");
Serial1.print(prodArea*areaPlantio/10);
Serial1.print(" ");
Serial1.print(unidade);
Serial1.println(")");
Serial1.println();

Serial1.print("Estimativa de produtividade: ");
Serial1.print(prodArea*areaPlantio/10);
Serial1.print(" ");
Serial1.print(unidade);
Serial1.print(" em ");
//String tempoCultivo = "70"; // Em dias, do SD
Serial1.print(tempoCultivo);
Serial1.println(" dias.");
Serial1.println();

Serial1.println("SetUp concluído!");
Serial1.println("Dispositivo em cultivo.");
Serial1.println("Pode desconectar seu aparelho Bluetooth!");
Serial1.println();
Serial1.println("FIM!");

idealUmid *= 10; // Converte de % para um valor proporcional ao que será lido
idealLumi *= 10;

// close the file:
myFile.close();
// TERMINA BANDO DE DADOS -----

Serial.begin(9600); // inTemperatura
sensors.begin();
sensors.setResolution(insideTemperatura, 10);
//sensors.setResolution(outsideTemperatura, 10);

//Atribuição da porta do Servo Motor do sensor de ph
phServo.attach(ElevadorSensorPh); // Anexa à porta 6
```

```
phServo.write(phServoPosInWater); // É melhor iniciar com ele fora d'água

//Temporizadores sem delay
minutoEscreverSD.defiSP(60*1000); // 60 segundos
phPassoServo.defiSP(10); // Apenas para o Servo motor não ir rápido demais
phControlando.defiSP(20*1000); // A cada 20 segundos, será adicionado ácido, nesse

pinMode(aguaReabastecida, INPUT); // Sensor de presença de água

pinMode(inUmidade, INPUT); // inUmidade
pinMode(inLuminosidade, INPUT); // inLuminosidade
pinMode(inPh, INPUT); // inPh

pinMode(fdc_aberto_cima, INPUT); // Fim de curso da porta de cima aberta
pinMode(fdc_fechado_cima, INPUT); // Fim de curso da porta de cima fechado
pinMode(fdc_aberto_baixo, INPUT); // Fim de curso da porta de baixo aberta
pinMode(fdc_fechado_baixo, INPUT); // Fim de curso da porta de baixo fechado

// SAÍDAS
pinMode(PortaAbrir, OUTPUT); // PortaAbrir
pinMode(PortaFechar, OUTPUT); // PortaFechar

pinMode(BombaDagua, OUTPUT);
//pinMode(ElevadorSensorPh, OUTPUT);
pinMode(AdicionarVinagre, OUTPUT);
pinMode(MisturadorVinagre, OUTPUT);
pinMode(Coolers, OUTPUT);

// Atualiza variáveis para evitar acionamentos indevidos
temperatura_old = idealTemp;
integral_old = 0;
}

void printTemperatura(DeviceAddress deviceAddress){ // inTemperatura
float tempC = sensors.getTempC(deviceAddress);
if (tempC == -127.00){
//Serial.print("Erro ao ler temperatura !");
TemperaturaAtual = 0;
}
else{
```



```

    TemperaturaAtual = tempC;
    /*
    Serial.print("C: ");
    Serial.print(tempC);
    Serial.print(" F: ");
    Serial.print(DallasTemperature::toFahrenheit(tempC));
    */
}
}

void loop(){
    /*
    //Este pedacinho de código foi substituído por temporizador sem delay
    numLoop++;
    if (numLoop >=100){
        numLoop = 0;
        numArquivo++;
    }
    */
    // delay(500); Ts = Ts + 500; // inTemperatura

    //Serial.print("Lendo temperaturas...\n\r");
    sensors.requestTemperatures(); // LINHA IMPORTANTE <--- <--- <---
    //Serial.print("Temperatura Sensor 1: ");
    printTemperatura(insideTemperatura);
    //Serial.print("\n\r");
    //Serial.print("Temperatura Sensor 2: ");
    //printTemperatura(outsideTemperatura);

    analogUmidadein = analogRead(inUmidade); // inUmidade
    analogLuminosidadein = analogRead(inLuminosidade); // inLuminosidade
    if (phControl == true){ // Lê ph apenas se está em andamento o controle do ph
        analogPh = analogRead(inPh); // inPh
        Vph = analogPh*5/1024;
        phin = 7 + ((2.5 - Vph)/0.18);
        for (i = 28; i >=0; i--){ // Vetor carrega últimas 30 medidas de ph
            phVet[i+1] = phVet[i];
        }
        phVet[0] = phin;
    }
}

```

```
}
/*
analogPh = analogRead(inPh); // inPh
Vph = analogPh*5/1024;
phin = 7 + ((2.5 - Vph)/0.18);
*/
/*
Serial.println("\tPH: ");
Serial.println(phin,3);
Serial.println("\tUmidade: ");
Serial.println(analogUmidadein);
Serial.println("\tLuminosidade: ");
Serial.println(analogLuminosidadein);
*/

analogUmidadeout = analogRead(inOutsideUmidade); // inUmidade
analogLuminosidadeout = analogRead(inOutsideLuminosidade); // inLuminosidad
//analogPh = analogRead(inPh); // inPh
//Vph = analogPh*5/1024;
//phout = 7 + ((2.5 - Vph)/0.18);
/*
Serial.println("\tPH: ");
Serial.println(phout,3);
Serial.println("\tUmidade: ");
Serial.println(analogUmidadeout);
Serial.println("\tLuminosidade: ");
Serial.println(analogLuminosidadeout);*/

/*
// CSV print (contem medidas de dentro e de fora da estufa de cada uma das 4 variá
vetor = String(analogUmidadein) + "," + String(analogUmidadeout) + "," + String(in

//vetor_minuto = vetor
//vetor_hora = vetor_hora + vetor_minuto

nomeArquivo = "Data" + String(numArquivo) + ".csv"; // Carrega o nome do arquivo
*/

if (minutoEscreverSD.Saida(1)){ // Quando completa contagem de 1 min, entra no if
```

```
minutoEscreverSD.Saida(0); // Reseta contagem logo em seguida para evitar perdas d
numLoop++;

// Teste de presença de água nova no recipiente de irrigação
if (digitalRead(aguaReabastecida) == HIGH && phControlInit == false && phControl =
  for (int i = 0; i < 5; i++){
    if (histWater[i] == 0){ // Se todas as leituras dos últimos 5 min são de ausên
      phControlInit = true; // Então temos água nova, precisamos controlar o ph.
    }
    else{
      phControlInit = false; // Caso contrário, não precisamos.
      break;
    }
  }
}
for (int i = 3; i >= 0; i--){
  histWater[i+1] = histWater[i]; // Rearranja o hitórico das leituras de água dos
}
histWater[0] = 0;
if(digitalRead(aguaReabastecida) == HIGH){histWater[0] = 1;} // Salva a leitura at

// Teste de anoitecer/amanhecer
for (int i = 3; i >= 0; i--){
  lumiNoite[i+1] = lumiNoite[i]; // Rearranja as leituras de luminosidade dos ulti
}
lumiNoite[0] = analogLuminosidadein; // Atualiza a leitura recente de luminosidade
// Testar se está noite:
for (int i = 0; i < 5; i++){
  if (lumiNoite[i] < 100){ // Valor estipulado para a iluminação do anoitecer
    auxNoite = true; // Se os ultimos 5 min de leitura de luminosidade indicam que
  }
  else{
    auxNoite = false; // Caso contrário, não está noite
    break; // Para de verificar e não atualiza a variavel booleana noite
  }
} // Encerra o teste do anoitecer
if (auxNoite == true){
  noite = true; // Atualiza a variável para noite
}
```

```
// Testar se está de manhã:
for (int i = 0; i < 5; i++){
    if (lumiNoite[i] > 400){ // Valor estipulado para a iluminação do amanhecer
        auxNoite = false; // Se os ultimos 5 min de leitura de luminosidade indicam qu
    }
    else{
        auxNoite = true; // Caso contrário, está noite
        break; // Para de verificar e não atualiza a variavel booleana noite
    }
} // Encerra o teste do amanhecer
if (auxNoite == false){
    noite = false; // Atualiza a variável para dia
}

if (numLoop >= 60){
    numLoop = 0;
    numArquivo++; // A cada 1 hora, muda o nome do arquivo que receberá vetores de l
    nomeArquivo = "Data" + String(numArquivo) + ".csv"; // Carrega o nome do arquivo
}
criar_vetor = true;
tempo_de_escrever = true;
escreveu = false;
}

if (escreveu == false){
    if (tempo_de_escrever == true){
        contagem_vetores ++; // Conta minutos até completar 1 dia
        if (contagem_vetores >= (24*60)){
            contagem_vetores = 0; // Reseta contagem de minutos
            /*
            if (sd_ou_micro == true){
                sd_ou_micro == false;
            }
            else{
                sd_ou_micro == true;
            }*/
        }
        /*
        if (sd_ou_micro == true){
```

```

    digitalWrite(LigarMicroSD_led, LOW);
    digitalWrite(LigarSD_led, HIGH);
}
else{
    digitalWrite(LigarSD_led, LOW);
    digitalWrite(LigarMicroSD_led, HIGH);
}*/
tempo_de_escrever = false;
}

if (criar_vetor == true){
    //vetor = String(analogUmidadein) + ";" + String(analogUmidadeout) + ";" + Strin
    int contTemp = 0; if(portas == true){contTemp = 1;} // Saída booleana do control
    int contUmid = 0; if(umidControl == true){contUmid = 1; umidControl = false;} //
    int contLumi = 0; if(lumiControl == true){contLumi = 1; lumiControl = false;} //
    int contPh = 0; if(phControl == true){contPh = 1;} // Saída booleana do controle
    vetor = String(numArquivo) + ";" + String(numLoop) + ";" + String(TemperaturaAtu
    // vetor = hora; minuto; ...
    //      temperatura; umidade; luminosidade; ph;... // Sensores
    //      temperatura; umidade; luminosidade; ph      // Atuadores
    criar_vetor = false;
}

/*
if (sd_ou_micro == true){
    LigarMicroSD
    delay(200);
}*/

Serial.print("Initializing SD card...");
pinMode(CSpin, OUTPUT);          //SD
if (!SD.begin(CSpin)) {
    Serial.println("Card failed, or not present");
    return;
}
Serial.println("card initialized.");

sensorData = SD.open(String(nomeArquivo), FILE_WRITE); // Se der erro retirar Stri
if (sensorData){

```

```
sensorData.println(vetor);
sensorData.close(); // close the file
escreveu = true; // Informa ao código que foi bem sucedido e voltará a escrever no
if (AUX_LED_SD == true){
    digitalWrite(LigarMicroSD, HIGH);
    AUX_LED_SD = false;
}else{
    digitalWrite(LigarMicroSD, LOW);
    AUX_LED_SD = true;
}

}

else{
Serial.println("Error writing to file !");
escreveu = false; // Continua a tentar escrever no SD já no próximo loop principal
}
}

/*
sensorData = SD.open(String(nomeArquivo), FILE_WRITE); // Se der erro retirar Stri
if (sensorData){
sensorData.println(vetor);
sensorData.close(); // close the file
}
else{
Serial.println("Error writing to file !");
}
*/

// SAÍDAS //

//Abrir Portas
digitalWrite(PortaFechar, LOW);
digitalWrite(PortaAbrir, HIGH);
delay(delay_portas); //Ts = Ts + delay_portas;
digitalWrite(PortaAbrir, LOW);

//Fechar Portas
digitalWrite(PortaAbrir, LOW);
```

```
digitalWrite(PortaFechar, HIGH);
delay(delay_portas); //Ts = Ts + delay_portas;
digitalWrite(PortaFechar, LOW);
/*
//Abrir Sombrite
digitalWrite(SombriteFechar, LOW);
digitalWrite(SombriteAbrir, HIGH);
delay(delay_portas); Ts = Ts + delay_portas;
digitalWrite(SombriteAbrir, LOW);

//Fechar Sombrite
digitalWrite(SombriteAbrir, LOW);
digitalWrite(SombriteFechar, HIGH);
delay(delay_portas); Ts = Ts + delay_portas;
digitalWrite(SombriteFechar, LOW);
*/

// CONTROLE //

// Controle de Umidade
// Aqui esperamos algumas voltas no loop para realizar uma nova leitura de umidade
// Uma espera para a umidade fluir no solo

if(loop_umidade >= 10){ // Depois de quantas voltas no loop volto a verificar a umidade
    if(analogUmidadein < (idealUmid - margemUmid)){
        umidControl = true;
        digitalWrite(BombaDagua, HIGH);
    }
    if(analogUmidadein > (idealUmid + margemUmid)){
        digitalWrite(BombaDagua, LOW);
    }
    loop_umidade = 0; // Reseta contagem do loop para atualizar controle de umidade
}
if(loop_umidade == 1){ // Depois de quantas voltas no loop desliga-se a bomba d'água
    digitalWrite(BombaDagua, LOW);
}
loop_umidade++;
```

```
// Controle de Temperatura
```

```
if(insideTemperatura > (idealTemp - margemTemp)){
  if(portas == false){ // Se portas fechadas, então:
    digitalWrite(PortaFechar, LOW);
    analogWrite(PortaAbrir, 255);
    delay(50);
    analogWrite(PortaAbrir, 230);
    delay(15);
    analogWrite(PortaAbrir, 180);
    delay(10);
    analogWrite(PortaAbrir, 130);
    delay(15);
    analogWrite(PortaAbrir, 115);
    delay(30);
    analogWrite(PortaAbrir, 100);
    delay(30);
    analogWrite(PortaAbrir, 60);

    //digitalWrite(PortaAbrir, HIGH);
    //delay(delay_portas); Ts = Ts + delay_portas;
    while(digitalRead(fdc_aberto_cima) == LOW || digitalRead(fdc_aberto_baixo) ==
      delay(10);
    }
    digitalWrite(PortaAbrir, LOW);
    portas = true; // Portas abertas!
  }
  Ts = millis() - tempo_controle_temperatura; // Captura dt
  tempo_controle_temperatura = millis(); // Define tempo inicial do novo dt
  Ts = Ts/1000; // Converte Ts de [ms] para [s]
  erro = idealTemp - TemperaturaAtual; // Diferença entre a Temp ideal e a atual

  integral = integral_old + Ts*erro; // Ts está em segundos
  diferencial = (temperatura_old - TemperaturaAtual)/Ts; //

  PID = K_p*erro + K_i*integral + K_d*diferencial;

  temperatura_old = TemperaturaAtual;
  integral_old = integral;
```



```
    //Ts = 1; // Reseta Ts
}
else{
    if(portas == true){ // Se portas abertas, então:
        digitalWrite(PortaAbrir, LOW);
        analogWrite(PortaFechar, 255);
        delay(50);
        analogWrite(PortaFechar, 230);
        delay(15);
        analogWrite(PortaFechar, 180);
        delay(10);
        analogWrite(PortaFechar, 130);
        delay(15);
        analogWrite(PortaFechar, 115);
        delay(30);
        analogWrite(PortaFechar, 100);
        delay(30);
        analogWrite(PortaFechar, 60);

        //digitalWrite(PortaFechar, HIGH);
        //delay(delay_portas); Ts = Ts + delay_portas;
        while(fdc_fechado_cima == LOW || fdc_fechado_baixo == LOW){
            delay(10);
        }
        digitalWrite(PortaFechar, LOW);
        portas = false; // Portas fechadas!
    }
}

// Controle de Lumidosidade
if (noite == false){ // Se está dia, mas com baixa luminosidade, liga
    // O teste de dia ou noite foi para o IF que se inicia a cada minuto, acima.
    //if(analogLuminosidadein <= ideallLumi - margemLumi){
        lumiControl = true;
        digitalWrite(Lampada, HIGH);
    //}
    //else{
    // digitalWrite(Lampada, LOW);
    //}
```

```
}
else{ // Se está noite, não queremos lâmpada ligada
  digitalWrite(Lampada, LOW);
}

// Controle de ph
// Ler ph somente quando repor água e controlar ph --> OK
// A verificação de água nova está no IF que se inicia a cada minuto, acima.
if (phControlInit == true){ // Este IF é responsável por iniciar o controle de ph
  phServoPos = phServoPosOutWater; // A próxima etapa começa com o sensor fora d'á
  phControlInit = false; // Já demos início ao controle de ph
  phControl = true; // Indica que o processo de controle de ph está em andamento
  moveDown = true; // Inicia o imergir do sensor (para dentro d'água)
}
if (phPassoServo.Saida(1) && moveDown == true && phServoPos < phServoPosInWater){
  phPassoServo.Saida(0);
  phServoPos++;
  phServo.write(phServoPos);
  if (phServoPos >= phServoPosInWater){
    moveDown = false; // Já imergiu
    phMedir = true; // Começar etapa de medição e controle do ph
  }
}
//
//Imergiu o sensor (para dentro d'água)
//
digitalWrite(MisturadorVinagre, HIGH); // Liga o misturador
//
Aux_break = false; // break artificial falso
if (Aux_break == false || (phMedir == true && phControlando.Saida(1))){ // A cada
  phControlando.Saida(0); // Reinicia contador
  phin = 0;
  for (i = 0; i < 30; i++){
    phin += phVet[i];
  }
  phin = phin/30; // Finaliza o cálculo da média do ph (de 30 medidas)
  //
  if (phin >= idealPh){ // Se ainda não atingimos o objetivo de ph...
    digitalWrite(AdicionarVinagre, HIGH);
```

```
    if ((phin - idealPh) > 1){
        delay(3000); // Se erro de ph maior que 1, 3 segundos de injeção de ácido
    }
    else{
        delay(500); // Apenas meio segundo de injeção de ácido
    }
    digitalWrite(AdicionarVinagre, LOW);
}
else{
    phMedir = false; // Objetivo de ph atingido com sucesso!!!
    moveUp = true; // Inicia o emergir do sensor (para fora d'água)
    Aux_break = true; // break artificial
}
}
//
digitalWrite(MisturadorVinagre, LOW); // Desliga o misturador
//
//Começar a emergir o sensor (para fora d'água)
//
if (phPassoServo.Saida(1) && moveUp == true && phServoPos > phServoPosOutWater){ /
    phPassoServo.Saida(0);
    phServoPos--;
    phServo.write(phServoPos);
    if (phServoPos <= phServoPosOutWater){
        moveUp = false; // Já emergiu
        phControl = false; // Termina aqui o processo de controle do ph
    }
} // FIM do Controle de ph

} // Fecha Void Loop, o loop principal
```