

Faculdade de Tecnologia de São Paulo

DEPARTAMENTO DE SISTEMAS ELETRÔNICOS

**Estufa modular com sistema automatizado de controle**

Gustavo Fernando Mastrocollo Gea

Jacson Silveira Ribeiro

São Paulo - SP

2023

# Faculdade de Tecnologia de São Paulo

DEPARTAMENTO DE SISTEMAS ELETRÔNICOS

Gustavo Fernando Mastrocollo Gea

Jacson Silveira Ribeiro

## **Estufa modular com sistema automatizado de controle**

Trabalho de Conclusão de Curso apresentado como requisito parcial para a conclusão do curso Superior de Tecnologia em Eletrônica Industrial da Faculdade de Tecnologia de São Paulo.

Orientador: Prof. Dr. Mauricio Antônio Deffert

São Paulo - SP

2023

## Ficha Catalográfica

Gea, Gustavo F. Mastrocollo; Ribeiro, Jacson Silveira

Estufa modular com sistema automatizado de controle / Gea, Gustavo F. Mastrocollo; Ribeiro, Jacson Silveira – 2023.

50 f.

Monografia – Curso Superior de Tecnologia em Eletrônica Industrial – Faculdade de Tecnologia de São Paulo, 2023.

Orientador(a): Prof. Dr. Mauricio Antônio Deffert

1. Estufa 2. Automatização 3. Controle

Dedicamos este trabalho a todos que buscam constantemente melhorar seu hábito alimentar e que não possuem condições financeiras compatíveis com alimentos orgânicos. Acreditamos que a automação e o controle inteligente de estufas são a chave para impulsionar a produtividade de alimentos saudáveis para todos. Agradecemos a todos os envolvidos neste projeto, que contribuíram com seu conhecimento e dedicação para o desenvolvimento deste sistema automatizado de controle de estufas. Que este trabalho inspire e beneficie muitas pessoas ao redor do mundo, ajudando a promover uma agricultura mais acessível, sustentável e responsável com o meio ambiente.

## *Agradecimentos*

Agradecemos aos nossos familiares e professores por nunca terem desistido de nosso desempenho acadêmico, mantendo sempre o apoio e suporte necessário a nós. Agradecemos também à instituição por nos prover estrutura necessária para nosso desenvolvimento acadêmico.

*"Automatizar tarefas não significa perder a humanidade. Pelo contrário, significa ter mais tempo para se dedicar a atividades que somente seres humanos podem fazer."*

*Satya Nadella, 2017*

## LISTA DE FIGURAS

Figura 1 - <b>Ilustração explicativa de aeroponia</b> .....	Pág. 05
Figura 2 - <b>Ilustração exemplificada de rotação de culturas</b> .....	Pág. 06
Figura 3 - <b>Exemplo de hidroponia</b> .....	Pág. 07
Figura 4 - <b>Sensor LM35</b> .....	Pág. 09
Figura 5 - <b>Módulo Sensor De Umidade De Solo</b> .....	Pág. 09
Figura 6 – <b>LDR</b> .....	Pág. 10
Figura 7 - <b>Microcontrolador Arduino UNO</b> .....	Pág. 12
Figura 8 - <b>LEDs coloridos</b> .....	Pág. 13
Figura 9 - <b>Entrada e saída do protótipo de Santos (2014)</b> .....	Pág. 16
Figura 10 - <b>Protótipo do projeto de Santos (2014)</b> .....	Pág. 17
Figura 11 - <b>Protótipo do trabalho de Lang (2013)</b> .....	Pág. 18
Figura 12 - <b>Interface do ScadaBR de Lang (2013)</b> .....	Pág. 19
Figura 13 - <b>Software desenvolvido pelos autores</b> .....	Pág. 24
Figura 14 – <b>Plataforma Arduino IDE</b> .....	Pág. 25
Figura 15 – <b>Visão Frontal e Traseira do Protótipo em 3D</b> .....	Pág. 28
Figura 16 – <b>Visão Laterais do Protótipo em 3D</b> .....	Pág. 28
Figura 17 – <b>Visão Frontal do Projeto</b> .....	Pág. 29
Figura 18 – <b>Visão Lateral do Projeto</b> .....	Pág. 30
Figura 19 – <b>Parâmetros para Testes</b> .....	Pág. 31
Figura 20 – <b>Gráfico de Temperatura</b> .....	Pág. 31
Figura 20 – <b>Gráfico de Umidade do Solo</b> .....	Pág. 32
Figura 21 – <b>Dados lidos</b> .....	Pág. 33
Figura 22 – <b>Sistema de rega</b> .....	Pág. 34
Figura 23 – <b>Testes</b> .....	Pág. 35
Figura 24 – <b>Protótipo final</b> .....	Pág. 36

## **LISTA DE TABELAS**

<b>Tabela 1 - Análise Comparativa.....</b>	<b>Pág. 20</b>
<b>Tabela 2 – Custos do Projeto.....</b>	<b>Pág. 36</b>



## **LISTA DE ABREVIATURAS**

LED *Light-Emitting Diode*

DC *Direct Current*

LCD *Liquid Crystal Display*

LDR *Light Dependent Resistor*

pH *Potencial (ou potência) Hidrogeniônico*

## RESUMO

O expressivo aumento da população gerou alternativas produtivas, como a existência de plantio em espaços confinados, tais como apartamentos. Neste contexto, a existência de uma estufa pode favorecer produtividade, segurança alimentar e acesso à alimentação de qualidade, aliado com uma forma mais simples e eficiente, além de apresentar um baixo custo de desenvolvimento. Este trabalho apresenta como solução um sistema automatizado para o controle de parâmetros em estufas (como por exemplo temperatura, luminosidade, umidade do solo e ventilação do ambiente). Esse sistema tem como funções: absorver o calor que vem do LED e manter acondicionada em seu interior, controlando a temperatura; conservar as plantas com qualidade e por mais tempo, livre de pragas e efeitos externos, como por exemplo, chuvas e ventos; controle de microclimas em qualquer estação do ano, não havendo necessidade de rotacionar plantações iguais às plantações tradicionais. O resultado do trabalho é a implementação de um sistema automatizado para controle de estufas, baseado na plataforma Arduino e Visual Studio, operado através de um aplicativo instalado em um computador com sistema operacional Windows, proporcionando vantagens como acesso às funções de controle da plantação e monitoramento em tempo real das condições da estufa.

**Palavras-chave:** Estufa, Automatização, Controle.

## **ABSTRACT**

The significant increase in population has generated productive alternatives, such as the cultivation of plants in confined spaces, such as apartments. In this context, the existence of a greenhouse can promote productivity, food security, and access to quality food, along with a simpler and more efficient approach, while also presenting a low cost of development. This paper presents a solution in the form of an automated system for controlling parameters in greenhouses (such as temperature, lighting, soil humidity, and environmental ventilation). This system functions to: absorb heat from the LED and maintain it within the greenhouse, thereby controlling the temperature; preserve plants with quality and for a longer period, free from pests and external effects, such as rain and wind; control microclimates throughout the year, eliminating the need for rotating crops like in traditional farming. The outcome of this work is the implementation of an automated system for greenhouse control, based on the Arduino and Visual Studio platform, operated through an application installed on a computer running the Windows operating system, providing advantages such as access to plant control functions and real-time monitoring of greenhouse conditions.

**Keywords:** Greenhouse, Automation, Control.

## SUMÁRIO

<b>LISTA DE FIGURAS.....</b>	<b>Pág. VIII</b>
<b>LISTA DE TABELAS.....</b>	<b>Pág. IX</b>
<b>LISTA DE ABREVIATURAS.....</b>	<b>Pág. X</b>
<b>1. INTRODUÇÃO .....</b>	<b>Pág. 1</b>
<b>1.1 Objetivos .....</b>	<b>Pág. 2</b>
<b>1.1.1 Objetivo geral .....</b>	<b>Pág. 2</b>
<b>1.1.2 Objetivos específicos .....</b>	<b>Pág. 2</b>
<b>1.1.3 Organização do trabalho.....</b>	<b>Pág. 2</b>
<b>2. FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>Pág. 4</b>
<b>2.1 Técnicas de plantações.....</b>	<b>Pág. 4</b>
<b>2.2 Sensores.....</b>	<b>Pág. 8</b>
<b>2.3 Conceitos básicos de Arduino e sua aplicação em sistemas automatizados.....</b>	<b>Pág. 11</b>
<b>2.4 LEDs.....</b>	<b>Pág. 12</b>
<b>2.5 Linguagens de programação utilizada.....</b>	<b>Pág. 14</b>
<b>2.5.1 Linguagem Wiring.....</b>	<b>Pág. 14</b>
<b>2.5.2 Linguagem C#.....</b>	<b>Pág. 15</b>
<b>2.6 Estado da Arte.....</b>	<b>Pág. 15</b>
<b>2.6.1 Sistema Automatizado para Controle de Umidade e Temperatura em Cultura de Morangos Aplicados aos Pequenos Produtores.....</b>	<b>Pág. 15</b>
<b>2.6.2 Desenvolvimento de um Irrigador Microcontrolado Integrado a um Sistema de Supervisão e Controle.....</b>	<b>Pág. 17</b>

2.6.3	<b>Comparativo</b> .....	Pág. 20
3.	<b>DESENVOLVIMENTO</b> .....	Pág. 22
3.1	<b>Microsoft Visual Studio 2022</b> .....	Pág. 23
3.2	<b>Arduino IDE</b> .....	Pág. 24
3.2.1	<b>Biblioteca de comunicação serial</b> .....	Pág. 26
3.3	<b>Montagem do protótipo</b> .....	Pág. 27
3.4	<b>Testes e Validação do protótipo</b> .....	Pág. 30
3.5	<b>Tabela de Custos</b> .....	Pág. 36
4.	<b>CONCLUSÕES</b> .....	Pág. 37
4.1	<b>Projetos Futuros</b> .....	Pág. 38
5.	<b>REFERÊNCIAS</b> .....	Pág. 40
6.	<b>APÊNDICES</b> .....	Pág. 44
	<b>APÊNDICE A – Bloco 01 de código do Projeto</b> .....	Pág. 44
	<b>APÊNDICE B – Bloco 02 de código do Projeto</b> .....	Pág. 44
	<b>APÊNDICE C – Bloco 03 de código do Projeto</b> .....	Pág. 45
	<b>APÊNDICE D – Bloco 04 de código do Projeto</b> .....	Pág. 46
	<b>APÊNDICE E – Blocos 05, 06 e 07 de código do Projeto</b> .....	Pág. 48
	<b>APÊNDICE F – Bloco 08 de código do Projeto</b> .....	Pág. 49
	<b>APÊNDICE G – Bloco 09 de código do Projeto</b> .....	Pág. 50
	<b>APÊNDICE H - Circuito de Testes simulado no Proteus</b> .....	Pág. 52



# 1. INTRODUÇÃO

Atualmente, grandes problemas atingem a humanidade de modo global, advindos de várias fontes distintas. Um grande problema, por exemplo, é a geração de produtos alimentícios. Os fatores que geram esse estresse incluem o crescimento populacional desproporcional ao crescimento da produção de alimentos, uma vez que a população cresce de forma geométrica e a produção de alimentos cresce de forma aritmética, de acordo com a Teoria Malthusiana (ABRAMOVAY, 2010)(TIVELLI, 2012)(OLIVEIRA & FACHINELLO, 2018). Outros dois grandes fatores incluem o crescente uso de produtos agroquímicos e alto custo de produtos orgânicos, que tendem a ser mais caros que os convencionais por diversos motivos.

As estufas são estruturas fechadas que podem controlar temperatura, umidade, luminosidade e efetivamente pragas e doenças, sendo capazes de se adaptar a diferentes tipos de cultivos e tornando o microclima adequado em todas as estações do ano. O uso de estufas tornou-se uma necessidade para o abastecimento seguro de alimentos no mundo, uma vez que traz mais garantia aos consumidores.

Além disso, "[...] a utilização de uma estufa automatizada é de grande importância para a otimização do cultivo, pois a interferência humana será mínima e dessa forma o clima no interior da estufa estará adaptado rigorosamente às condições necessárias para o melhor desenvolvimento da plantaçã"(GUEDES; SOUSA, 2016).

Nesse sentido, o projeto busca driblar esses fatores que geram problemas e trazer para dentro da casa das pessoas uma ajuda para cultivo de seus alimentos com baixo custo. Através de um microcontrolador, sensores e dispositivos que imitam condições ideais ao do cultivo, será possível atingir a proposta.

## 1.1 **Objetivos**

Este trabalho tem como objetivos o desenvolvimento de protótipo de estufa com sistema automatizado com sensores e dispositivos de controle, assim como projetar, construir e testar o hardware e software para funcionamento deste protótipo.

### 1.1.1 **Objetivo geral**

O objetivo geral consiste no desenvolvimento do protótipo, utilizando materiais de baixo custo em sua estrutura, microcontrolador Arduino e diversos sensores e dispositivos de controle de temperatura, ventilação e umidade, tendo o funcionamento comandado através de uma programação em linguagem C gravada em microcontrolador.

### 1.1.2 **Objetivos específicos**

Os objetivos específicos têm como proposta o projeto, construção e testes do hardware de forma a adequar as melhores condições e posições para um melhor funcionamento, como também o projeto do software em programa específico e simuladores para melhoria contínua.

### 1.1.3 **Organização do trabalho**

No Capítulo 2 será apresentado toda a fundamentação teórica necessária para entendimento do trabalho, destacando-se as características dos sensores, o Arduino e as linguagens de programação utilizadas. Já no Capítulo 3 será apresentado o



desenvolvimento do projeto na plataforma Microsoft Visual Studio 2022, plataforma IDE do Arduino e a montagem do protótipo. Por fim, no Capítulo 4 será apresentada a conclusão do projeto e suas aplicações futuras.

## **2. FUNDAMENTAÇÃO TEÓRICA.**

Para compreensão deste trabalho, além da revisão de bibliografia para determinar dispositivos semelhantes já construídos, são necessários os seguintes fundamentos teóricos: a) automatização de sistemas, sua definição e questões práticas para o projeto; b) funcionamento de sensores (LM35, LM393 e GL5528) e válvulas; c) tecnologias utilizadas em estufas; d) conceitos básicos em Arduino; e) LEDs; f) linguagens utilizadas no trabalho e g) trabalhos de referência.

### **2.1 Técnicas de plantações**

Um sistema automático realiza as funções independentes de ajustes externos, ou seja, a atuação autônoma não depende da ação humana. Já um sistema automatizado requer pequenos ajustes do usuário ou do fabricante para melhor funcionamento e adequação ao uso requisitado. Dessa forma, a automatização busca trazer benefícios à utilização de um sistema, nesse caso de uma estufa, através da utilização de sensores que geram valores ao microcontrolador e este, por sua vez, realiza a comparação e tomada de decisões através de uma programação desenvolvida.

Abaixo pode-se observar técnicas de plantações para se utilizar na automatização, obtendo um melhor plantio e cultivo. A Figura 1 apresenta o modelo de aeroponia que consiste em manter as plantas suspensas no ar, deixando suas raízes livres de terra, pulverizando-as com solução nutritiva (MEGMENG, 2018).

**Figura 01 – Ilustração explicativa de aeroponia**



Fonte: <https://www.anahuac.mx/mexico/noticias/Aeroponia-cultivo-sin-tierra>

A Figura 2 apresenta a técnica de rotação de culturas, que consiste em alterar o cultivo de plantas em uma mesma área por um determinado tempo, visando diminuir a exaustão de nutrientes do solo, desde que as plantações utilizadas na rotação necessitem de diferentes necessidades de adubação (FRANCHINI, 2011).

**Figura 02 – Ilustração exemplificada de rotação de culturas**



Fonte: <https://www.sementesvivas.bio/pt/biblioteca/artigos-e-dicas-de-sementeira/rotacao-cultural>

Na Figura 3 é apresentado a técnica de hidroponia, que consiste em deixar as raízes das plantas imersas em uma solução nutritiva, contendo tudo o que é necessário para seu melhor desenvolvimento (MEGMENG, 2018).

**Figura 03 – Exemplo de hidroponia**



Fonte: <https://plataformahidroponia.com/a-hidroponia/>

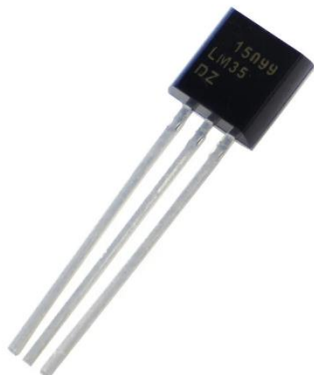
## 2.2 Sensores

Para construção do protótipo, utiliza-se simulador para definição dos circuitos e para o desenvolvimento dos respectivos softwares. As informações necessárias para definição dos principais componentes elétricos podem ser encontradas nos principais manuais (MICROCHIP, 2022)(ARDUINO, 2022)(TEXAS INSTRUMENTS, 2017)(MOTOROLA, 2022)(PHOTOCONDUCTIVE CELLS, 2022)(PARKER, 2002).

Os sensores são dispositivos eletrônicos que permitem medir e monitorar variáveis físicas e químicas em um ambiente. Dentre os sensores mais comuns em projetos eletrônicos estão os sensores de temperatura, umidade do solo e luminosidade.

O LM35 é um sensor de temperatura linear que converte a temperatura em uma tensão analógica proporcional à variação da temperatura. Este sensor é capaz de medir temperaturas em um intervalo de  $-55^{\circ}\text{C}$  a  $150^{\circ}\text{C}$  com precisão de  $0,5^{\circ}\text{C}$ . O LM35 é amplamente utilizado em projetos de automação residencial, controle de temperatura de equipamentos eletrônicos, dentre outros.

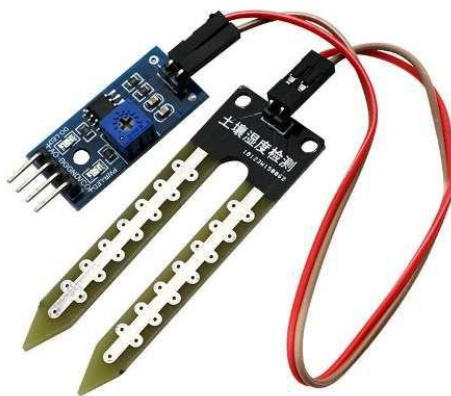
**Figura 04 – Sensor LM35**



Fonte: <https://www.baudaeletronica.com.br/sensor-de-temperatura-lm35.html>

O sensor de umidade do solo LM393 é um sensor capacitivo que permite medir o nível de umidade do solo. Este sensor é capaz de medir a umidade do solo em um intervalo de 0% a 100% com precisão de 5%. O LM393 é amplamente utilizado em projetos de automação agrícola, controle de irrigação de plantas e hortas, dentre outros.

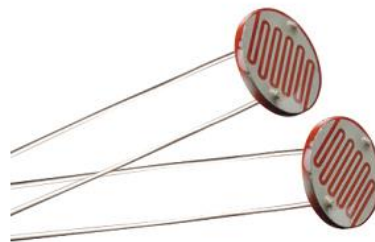
**Figura 05 - Módulo Sensor De Umidade De Solo**



Fonte: <https://www.eletrogate.com/modulo-sensor-de-umidade-de-solo>

O LDR (*Light Dependent Resistor*) é um sensor que permite medir a luminosidade do ambiente. Este sensor é composto por um material semicondutor cuja resistência varia em função da intensidade luminosa. Quanto maior a luminosidade, menor é a resistência do sensor. O LDR é capaz de medir a luminosidade em um intervalo de 0 a 1000 lux e é amplamente utilizado em projetos de automação residencial, controle de iluminação e monitoramento de ambientes.

**Figura 06 – LDR**



Fonte: <https://www.olelectronics.in/product/ldr-light-dependent-resistor/>

Os sensores de temperatura, umidade do solo e luminosidade são componentes importantes em diversos projetos eletrônicos. Eles permitem medir e monitorar variáveis críticas para o bom funcionamento de equipamentos, plantas, hortas e ambientes. Com esses dados, é possível desenvolver sistemas automatizados mais eficientes e sustentáveis, que promovam um melhor aproveitamento dos recursos disponíveis e um menor impacto ambiental.



### **2.3 Conceitos básicos de Arduino e sua aplicação em sistemas automatizados**

Arduino é uma plataforma eletrônica de código aberto que permite a criação de projetos de automação de baixo custo e com alta flexibilidade. Ele é composto por uma placa controladora com processador e uma série de componentes eletrônicos que permitem o controle de entrada e saída de dados em diferentes sistemas (FONSECA; BEPPU, 2010).

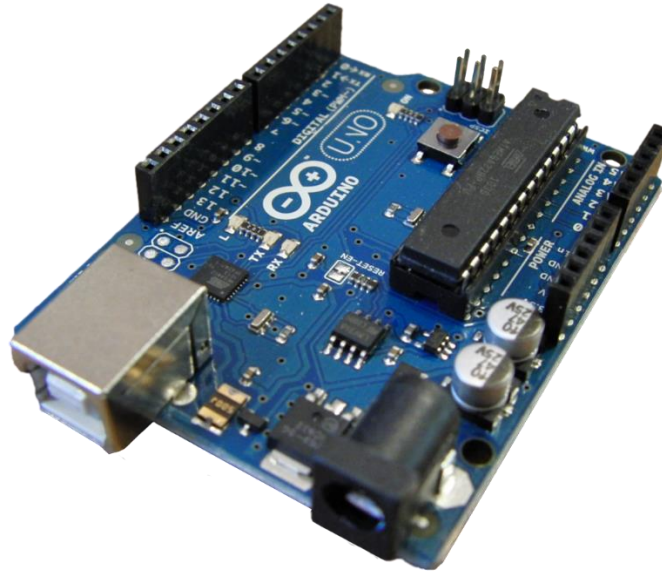
Os conceitos básicos de Arduino incluem a programação do microcontrolador, a configuração de pinos de entrada e saída, a leitura de sensores e a ativação de atuadores. A programação é realizada em uma linguagem de programação própria do Arduino, baseada em C/C++ (MCROBERTS, 2011), que é acessível mesmo para iniciantes em programação.

A aplicação de Arduino em sistemas automatizados é vasta e variada, incluindo aplicações em agricultura, indústria, automação residencial, entre outras. Em sistemas agrícolas, por exemplo, sensores podem ser utilizados para medir a umidade do solo e a temperatura ambiente, e atuadores podem ser usados para acionar irrigação automática ou ventiladores para controle de temperatura.

Na indústria, a plataforma Arduino pode ser usada para controle de processos automatizados, monitoramento de equipamentos e sistemas de segurança. Já na automação residencial, é possível controlar iluminação, temperatura ambiente, eletrodomésticos e sistemas de segurança, tudo isso por meio de dispositivos conectados à internet.

A flexibilidade e a facilidade de programação são as principais vantagens da plataforma Arduino em relação a outras tecnologias de automação. Além disso, o seu baixo custo permite a implementação de projetos de automação em diferentes setores, desde pequenas empresas até grandes indústrias.

**Figura 07 – Microcontrolador Arduino UNO**



Fonte: <https://www.squids.com.br/arduino/index.php/hardware/arduino/placas-arduino/67-arduino-uno-r3>

## 2.4 LEDs

Os LEDs (*Light Emitting Diodes*) são componentes eletrônicos amplamente utilizados em projetos com Arduino, seja para indicar o status de um dispositivo, criar efeitos visuais ou até mesmo para iluminar ambientes. Com sua grande variedade de cores e tamanhos, os LEDs oferecem uma grande flexibilidade para o desenvolvimento de projetos com o microcontrolador.

A utilização de LEDs com o Arduino é bastante simples. Os LEDs são dispositivos de polaridade, ou seja, possuem um lado positivo (ânodo) e um negativo (catodo). Para acender um LED, é necessário fornecer uma corrente elétrica no sentido

correto, do anodo para o catodo. Isso pode ser feito conectando-se o anodo do LED à porta digital do Arduino e o catodo a um resistor limitador de corrente, que é conectado ao GND do microcontrolador. A programação do Arduino pode então ser configurada para alternar a porta digital entre níveis altos e baixos, acendendo e apagando o LED (WIRTH LIMA, 2013).

É importante escolher o resistor limitador de corrente adequado para cada LED, a fim de evitar que a corrente elétrica seja excessiva e danifique o componente. A fórmula para calcular o valor do resistor é a seguinte:

$$R = \frac{(V - V_{led})}{I_{led}}$$

onde R é o valor do resistor em ohms, V é a tensão de alimentação,  $V_{led}$  é a tensão de queda do LED (geralmente em torno de 1,8V para LEDs comuns) e  $I_{led}$  é a corrente máxima que o LED pode suportar (geralmente em torno de 20mA para LEDs comuns).

**Figura 08 – LEDs coloridos**



Fonte: <http://endigital.orgfree.com/goodbit/led.htm>

## **2.5 Linguagens de programação utilizadas**

### **2.5.1 Linguagem Wiring**

A linguagem de programação utilizada em Arduino é uma linguagem derivada do C/C++, chamada Wiring (MCROBERTS, 2011). Essa linguagem foi criada especificamente para simplificar o processo de programação em Arduino, tornando-o mais acessível para iniciantes em programação e também para pessoas com experiência em outras linguagens.

A linguagem Wiring possui uma série de funções e bibliotecas pré-definidas que permitem aos programadores de Arduino controlar facilmente diferentes dispositivos e periféricos eletrônicos, como sensores, motores, displays, entre outros. Essas funções incluem comandos para leitura e escrita de pinos digitais e analógicos, controle de temporizadores, comunicação serial e interação com displays e outros dispositivos (MCROBERTS, 2015).

Uma das principais vantagens da linguagem Wiring é a sua simplicidade e facilidade de uso. Além disso, ela é compatível com uma ampla variedade de placas e dispositivos baseados em Arduino. Outra vantagem da linguagem Wiring é a sua comunidade ativa e de suporte. Como a plataforma Arduino é de código aberto e tem uma grande comunidade de usuários em todo o mundo, existem muitos fóruns, tutoriais e exemplos disponíveis para ajudar os programadores a aprender e usar a linguagem Wiring de forma eficaz.

### **2.5.2 Linguagem C#**

C# é uma linguagem de programação orientada a objetos moderna e poderosa, desenvolvida pela Microsoft em meados dos anos 2000. Ela é uma linguagem derivada do C++ e da linguagem Java, e foi criada com o objetivo de oferecer uma linguagem de programação mais fácil de usar e com recursos avançados para a criação de aplicativos para o sistema operacional Windows pelo Microsoft Visual Studio 2022 (TEHERAN E ROJAS, 2022).

A linguagem C# foi projetada para ser fácil de aprender, permitindo que desenvolvedores iniciantes possam criar aplicativos de alta qualidade. Ela oferece recursos avançados de programação orientada a objetos, como herança, encapsulamento e polimorfismo, além de suporte para programação genérica e delegação.

Uma das características mais importantes do C# é sua integração com a plataforma .NET da Microsoft. Isso permite aos desenvolvedores criar aplicativos .NET que são executados no ambiente de tempo de execução do .NET, o que garante a interoperabilidade entre diferentes aplicativos, componentes e serviços. Além disso, a plataforma .NET oferece uma ampla gama de bibliotecas de classe que simplificam o desenvolvimento de aplicativos complexos.

## **2.6 Estado da Arte**

### **2.6.1 Sistema Automatizado para Controle de Umidade e Temperatura em Cultura de Morangos Aplicados aos Pequenos Produtores**

Santos (2014) propôs um sistema de irrigação automatizado de baixo custo para pequenos produtores rurais, utilizando um sistema microcontrolado Arduino e comunicação por radiofrequência.

O hardware desenvolvido é capaz de controlar a irrigação, monitorar a temperatura e umidade do solo, acionar uma bomba d'água e abrir e fechar a estufa através de um servomotor. O sistema utiliza um conjunto de materiais, incluindo Arduino Nano, sensores de temperatura, umidade do solo e luminosidade, display LCD, módulo de relé, módulo Link de radiofrequência, servomotor e bomba de aspersão. A Figura 09 apresenta o esquema das entradas do protótipo.

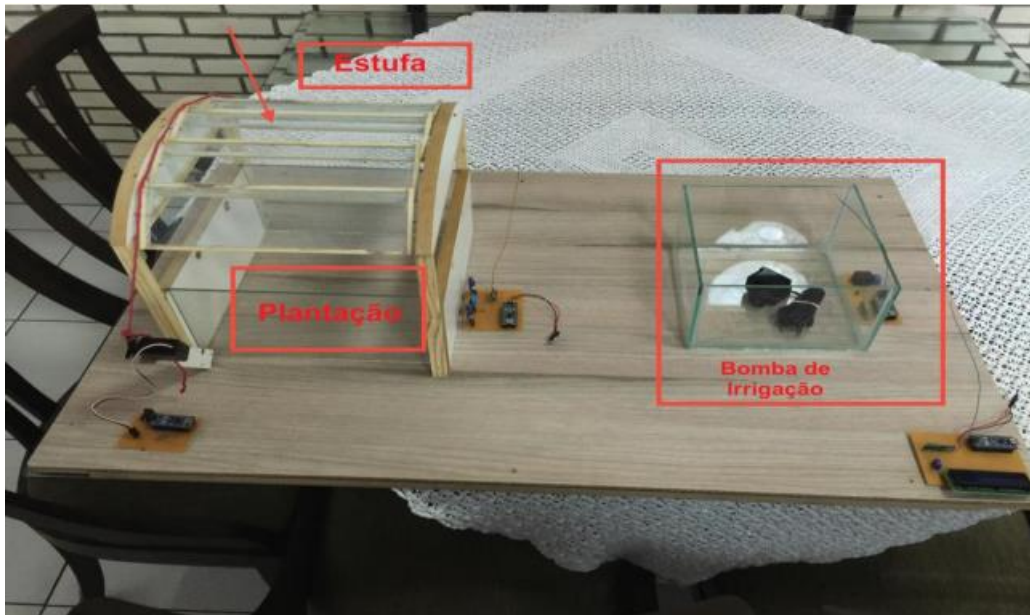
**Figura 09 – Entrada e saída do protótipo**



Fonte: Santos (2014)

Para controlar o sistema, o autor desenvolveu um software em linguagem C, com definições de valores para temperatura e umidade, parâmetros para atuadores e programação do display LCD. O protótipo foi testado em uma plantação simulada, demonstrando sua capacidade de medir umidade e temperatura, como podemos observar na Figura 10.

**Figura 10 – Protótipo do projeto**



Fonte: Santos (2014)

Apesar de atender às necessidades propostas, o autor identificou oportunidades de melhoria para o sistema, incluindo o uso de sensores mais precisos, o desenvolvimento de um programa para desktop para controlar a irrigação, a instalação de câmeras para monitorar a plantação e a implementação de um sistema de reutilização de água da chuva para irrigação.

### **2.6.2 Desenvolvimento de um Irrigador Microcontrolado Integrado a um Sistema de Supervisão e Controle**

O trabalho "Desenvolvimento de um Irrigador Microcontrolado Integrado a um Sistema de Supervisão e Controle" de Dutra, Lang e Waromby (2013) descreve um protótipo de sistema de irrigação automatizado baseado no microcontrolador Arduino. O

sistema é comandado por dispositivos móveis que possuam tecnologia wireless, utilizando um sistema de supervisão e controle desenvolvido no software ScadaBR.

O sistema desenvolvido realiza a irrigação automaticamente de acordo com a temperatura e umidade do ar, utilizando sensores para monitorar as condições ambientais. O sistema conta com um reservatório de água, que tem seus limites mínimo e máximo controlado automaticamente, e seu enchimento é controlado por uma válvula solenoide. O protótipo possui dois modos de funcionamento, manual e automático, permitindo que o usuário escolha se ele quer controlar o sistema de forma manual ou deixá-lo operando automaticamente. Na Figura 11 temos o protótipo apresentado pelo grupo:

**Figura 11 – Protótipo do trabalho**



Fonte: Lang (2013)

O sistema de supervisão e controle é executado no software ScadaBR, que é um software livre e gratuito de código-fonte aberto para automação e controle supervisão.



O sistema pode ser acessado a partir de um desktop, notebook, tablet ou smartphone. A comunicação entre o computador e o controlador Arduino ocorre de forma serial, com os comandos inseridos pelo operador sendo registrados em variáveis e enviados ao controlador para processamento. A Figura 12 apresenta a interface do software.

**Figura 12 – Interface do ScadaBR**



Fonte: Lang (2013)

Os autores destacam que encontraram dificuldades no desenvolvimento do código, pois a linguagem utilizada pelo controlador não era familiar à equipe, e na utilização dos sensores, pois o material fornecido pelos fabricantes não possuía detalhamento de suas programações. No entanto, com o término do projeto, os objetivos propostos foram alcançados. Os autores sugerem a implantação de sensores industriais em um trabalho futuro, para aumentar a precisão e durabilidade do sistema e, conseqüentemente, aumentar o grau de confiabilidade do projeto.

### 2.6.3 Comparativo

É evidente que a plataforma Arduino é amplamente utilizada nos projetos de automação estudados, e sua função principal é realizar a comunicação e controle entre os atuadores, sensores e periféricos. Além disso, uma observação importante feita a partir da Tabela 1 é que os projetos priorizam principalmente o controle dos atuadores, e há pouco uso de um aplicativo para controle do sistema.

A inclusão de um aplicativo para controle oferece praticidade, mobilidade e facilidade no projeto. É importante notar que nenhum dos programas possui uma função de feedback em forma de relatório que permita ao usuário gerenciar os dados armazenados em uma base de dados. A Tabela 1 apresenta uma análise comparativa entre os estudos anteriores e a solução proposta neste trabalho.

**Tabela 01 – Análise Comparativa**

Características	Trabalho de Santos (2014)	Trabalho de Dutra, Lang e Waromby (2013)	Solução Desenvolvida
Controle de Luminosidade	<b>Sim</b>	Não	<b>Sim</b>
Controle da umidade do solo	<b>Sim</b>	<b>Sim</b>	<b>Sim</b>
Controle da Temperatura	Não	Não	<b>Sim</b>

Timer automático para rega	Não	Não	<b>Sim</b>
Status do sistema	<b>Sim</b>	<b>Sim</b>	<b>Sim</b>
Implementação de um software web para controle e monitoramento	Não	Não	<b>Sim</b>
Microcontrolador usado	Arduino Nano	Arduino Uno	Arduino Uno

Fonte: Autores

### 3. DESENVOLVIMENTO

Trata-se de pesquisa aplicada, exploratória, com abordagem combinada. A montagem e testes do circuito necessitam apenas de bancadas convencionais e a parte mecânica, obtida por usinagem, convencional pode ser desenvolvida com ferramentas acessíveis a *hobbists*.

A abordagem utilizada neste projeto privilegia materiais de longa vida e *design* para minimização de consumo, o que torna o resultado mais ambientalmente correto.

Como critério de desenvolvimento, além do baixo custo, facilidade de montagem e de operação, em um sistema automatizado e programável. Esse protótipo deve atender hortaliças de pequeno porte, uma vez que é um dispositivo para uso *indoor*. Espera-se que seja de fácil operação mesmo para pessoas não familiarizadas com equipamentos eletrônicos, em particular programação.

Como condições de contorno tem-se que os componentes eletrônicos devem ser de fácil acesso no mercado. Ou seja, não se utiliza de materiais dependentes de importação. Igualmente as partes mecânicas, devem ser de fácil obtenção e padrão, de modo a evitar usinagem de partes e peças. Além disso, deseja-se também um dispositivo final de pequena dimensão e fácil transporte.

Para realizar o desenvolvimento do projeto, foi utilizado duas plataformas principais para programação:

- Visual Studio 2022 (Microsoft) para criação do GUI (guia de interface do usuário);
- Arduino IDE para desenvolvimento do código fonte a ser utilizado no Arduino Uno, juntamente com a biblioteca externa de comunicação serial (*SoftwareSerial.h*).

### 3.1 Visual Studio 2022

O Microsoft Visual Studio 2022 é a versão mais recente do ambiente de desenvolvimento integrado (*IDE*) da Microsoft, usado para criar aplicativos para Windows, Android, iOS e web, entre outros. Ele é um dos principais *IDEs* usados para desenvolver aplicativos .NET. O Visual Studio 2022 também pode ser usado para desenvolver aplicativos que se comunicam com dispositivos externos, como o Arduino, por meio de uma porta serial.

Para se comunicar com o Arduino, é necessário adicionar a biblioteca Serial ao projeto. Em seguida, é possível usar a classe *SerialPort* para abrir a porta serial e estabelecer a conexão com o Arduino. A classe *SerialPort* também permite a configuração de parâmetros como velocidade de transmissão e tamanho do buffer.

Uma vez que a conexão com o Arduino é estabelecida, o Visual Studio 2022 pode enviar comandos para o Arduino e receber dados de volta. Por exemplo, é possível enviar comandos para acender ou apagar LEDs conectados ao Arduino. Além disso, é possível ler dados de sensores conectados ao Arduino, como temperatura ou umidade, e exibi-los em uma interface gráfica do usuário (*GUI*) criada no Visual Studio 2022.

Neste trabalho foi desenvolvido um *GUI* que alterna entre 3 tipos de plantas pré-programadas via Arduino, onde o usuário consegue visualizar qual o tipo selecionado pelo próprio aplicativo ou também utilizar o modo manual. Além disso, o usuário pode alterar a banda de comunicação e a porta de comunicação que está sendo utilizada pelo Arduino.

Com isso, podemos realizar o controle dos parâmetros da estufa via aplicativo:

**Figura 13 – Software desenvolvido pelos autores**



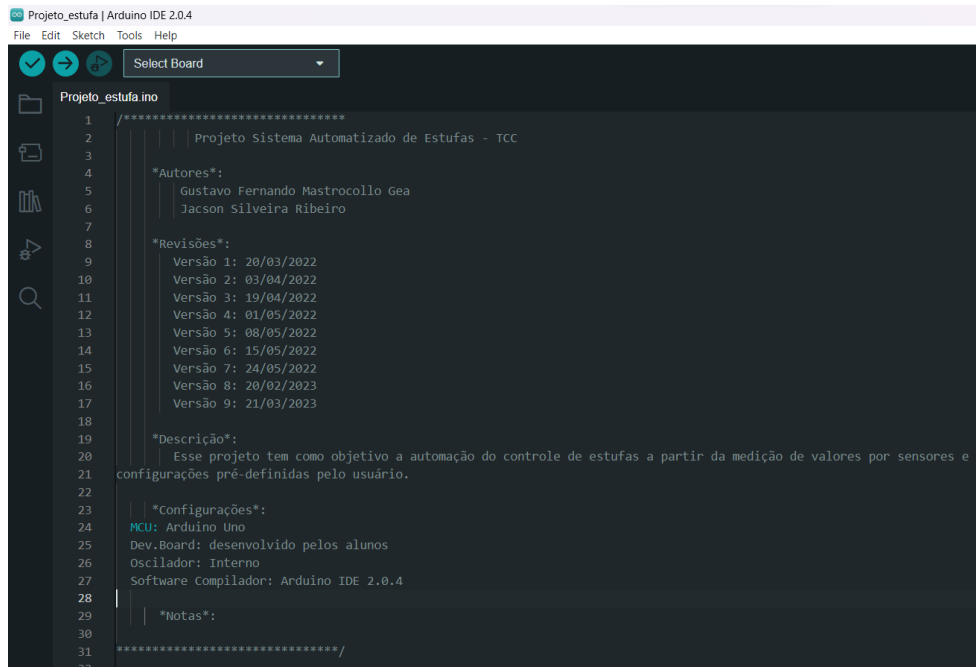
Fonte: Autores

### 3.2 Arduino IDE

A plataforma Arduino IDE (*Integrated Development Environment*) é uma ferramenta de software gratuita e de código aberto utilizada para programar placas Arduino. É uma das principais ferramentas utilizadas por desenvolvedores de todo o mundo para criar projetos eletrônicos que interagem com o mundo físico.

A interface da plataforma Arduino IDE é simples e fácil de usar, mesmo para iniciantes em programação. Ela inclui um editor de código, que permite aos usuários escrever, salvar e carregar programas para as placas Arduino. O IDE também inclui uma janela de monitor serial, que permite a comunicação entre o Arduino e o computador através da porta serial (MCROBERTS, 2011). Na Figura 14 temos um cabeçalho escrito na plataforma como exemplo.

**Figura 14 – Plataforma Arduino IDE**



Fonte: Autores

Após declarar todas as variáveis e portas do Arduino, partimos para a segmentação em blocos do projeto. O primeiro bloco (apêndice A) é composto pela configuração do LCD e dos caracteres iniciais que aparecem ao ligar o Arduino.

O segundo bloco (apêndice B) é composto pela configuração dos sensores. Onde declaramos a entrada de cada um e a configuração específica juntamente com a conversão dos valores. A leitura do sensor de umidade do solo foi ajustada com base no solo utilizado e separada em segmentações de porcentagem. Ou seja, quando o solo está seco, a umidade apresentada é 0%, e quando está úmido, a umidade apresentada é 100%.

No bloco 3 (apêndice C), é onde se situa a configuração da visualização dos parâmetros obtidos pelos sensores no LCD. Na leitura do sensor LDR, foi utilizado faixas de valores para definir se a luminosidade está baixa (*LOW*), média (*GOOD*) ou alta (*HIGH*).

No bloco 4 (apêndice D) temos a comunicação serial que recebe do aplicativo uma variável e a partir dela configura qual configuração entrará em uso. Além disso, a partir dessa variável é possível ligar ou desligar o sistema de rega, ligar ou desligar o sistema de ventilação da estufa ou também ajustar a luminosidade. Entretanto, quando um dos tipos de plantas está selecionado, não é possível realizar o ajuste manual.

Os blocos 5, 6 e 7 (apêndice E) são os blocos dos tipos de plantas. Onde cada planta possui seu tempo de rega específico, temperatura e luminosidade. O ajuste dessas variáveis pode ser facilmente feito, não sendo necessário um conhecimento intermediário ou avançado em programação de microcontroladores.

No bloco 8 (apêndice F) temos a configuração dos modos dos pinos, onde definimos o que são os pinos de entrada e os pinos de saída.

O bloco 9 (apêndice G) é composto pelo *setup* e pelo *loop*, onde definimos qual bloco é de configuração e qual bloco deverá ser mantido em repetição. Nota-se que nessas funções há poucos comandos, sendo composto por outras funções definidas anteriormente, tornando o código de fácil entendimento. Isso pode ser definido como uma programação estruturada (RICARTE, 2013).

### **3.2.1 Biblioteca de comunicação serial**

A biblioteca *SoftwareSerial.h* é uma biblioteca de software do Arduino que permite a comunicação serial em pinos digitais. Ela é útil quando a placa Arduino não possui portas serial suficientes para a comunicação com outros dispositivos.

A biblioteca pode ser usada para criar uma porta serial virtual em qualquer par de pinos digitais do Arduino. Isso significa que a comunicação serial pode ser estabelecida entre o Arduino e outros dispositivos, como módulos de GPS, módulos Bluetooth, entre outros.



O uso da biblioteca é simples e pode ser implementado com apenas algumas linhas de código. Primeiramente, a biblioteca deve ser incluída no início do programa, através do comando "#include <SoftwareSerial.h>". Em seguida, um objeto SoftwareSerial deve ser criado, definindo os pinos digitais que serão usados para a comunicação.

Após a criação do objeto, a comunicação serial pode ser iniciada com o comando "mySerial.begin(9600)", definindo a taxa de transmissão desejada.

A biblioteca SoftwareSerial.h também oferece funções para leitura e escrita de dados pela porta serial virtual criada. Por exemplo, a função "mySerial.read()" pode ser usada para ler os dados recebidos pela porta serial virtual e a função "mySerial.write()" pode ser usada para enviar dados pela porta serial virtual (ARDUINO, 2023).

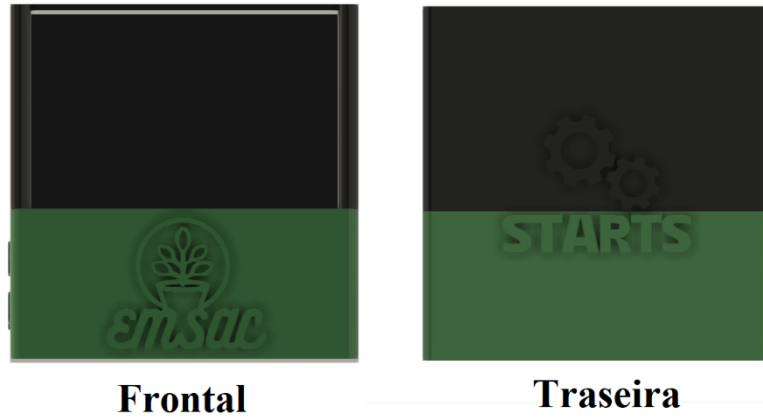
Através dessa comunicação serial no bloco 3 do projeto, o usuário ao realizar a alteração do tipo de planta pelo aplicativo, essa informação é recebida pelo Arduino e altera sua configuração de parâmetros pré-definidos. O usuário também tem a possibilidade de realizar o acionamento manual do sistema de rega, sistema de resfriamento e sistema de luminosidade.

### 3.3 Montagem do Protótipo

Na elaboração do protótipo foi desenvolvido uma maquete em pequena escala feita em impressão 3D, simulando uma estufa. Nela foram instalados os sensores de temperatura, luminosidade e umidade do solo. Foram instalados, também, seus atuadores: uma válvula solenoide 12V para ativação do sistema de rega, um ventilador de 5V com o intuito de regular a temperatura no interior da estufa e um led na cor branca responsável por manter o índice de luminosidade mais elevado em períodos de necessidade.

A Figura 15 apresenta o projeto feito em impressão 3D no software *Autodesk Fusion 360* pela visão frontal e traseira, onde podemos observar um logotipo personalizado para o protótipo e a empresa responsável pela impressão.

**Figura 15 – Visão Frontal e Traseira do Protótipo em 3D**



Fonte: Autores

A Figura 16 apresenta o projeto, no mesmo software citado anteriormente, porém com a visão das laterais, podendo ser observado que a lateral direita contém os nomes dos autores e na lateral esquerda a entrada de ar, onde será fixada a ventoinha.

**Figura 16 – Visão Laterais do Protótipo em 3D**

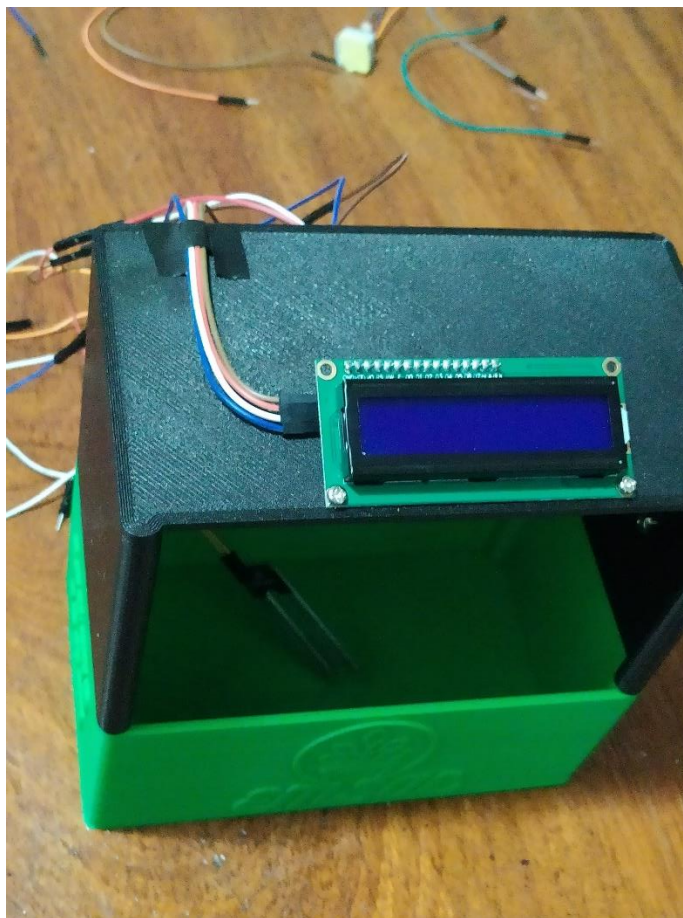


Fonte: Autores

O microcontrolador utilizado (Arduino UNO) e seu esquema elétrico também foram colocados junto ao protótipo para facilitar sua montagem com a estufa. O aplicativo que funcionará para controle da estufa está disponível no notebook no qual o Arduino ficará conectado. Foi usada uma fonte de 12 volts para fornecer a energia necessária para acionar válvula solenoide de irrigação. Para fazer a ligação da ventoinha e da válvula foram utilizados dois relés de 1 canal para Arduino.

A Figura 17 apresenta a visão frontal do protótipo com o início da fixação de alguns componentes (LCD na parte superior e o sensor de umidade do solo na parte interna).

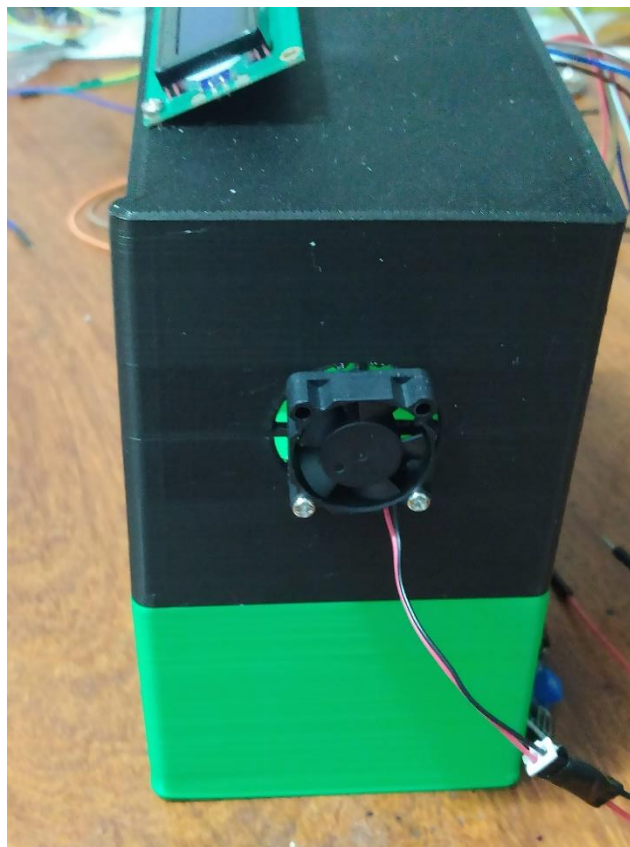
**Figura 17 – Visão Frontal do Projeto**



Fonte: Autores

A Figura 18 apresenta a visão lateral onde foi fixada a ventoinha utilizada no protótipo (diâmetro de 30mm e voltagem 5V).

**Figura 18 – Visão Lateral do Projeto**



Fonte: Autores

### 3.4 Testes e validação do Protótipo

Antes de iniciar o funcionamento de todos os componentes juntos, foi necessário realizar seus testes individualmente. Para isso, foram feitos programas de testes individuais para cada tipo de sensor e atuador, para verificar sua funcionalidade e a maneira que se comportam. Após isso, foram feitos testes envolvendo o sensor e o atuador que será controlado por esse sensor, como por exemplo é o caso do sensor de temperatura e a ventoinha. Para finalizar, os testes foram feitos entre o controle do aplicativo

desenvolvido com os atuadores. A Figura 19 apresenta os parâmetros utilizados para cada tipo de planta, onde o usuário pode selecionar pelo aplicativo qual deseja.

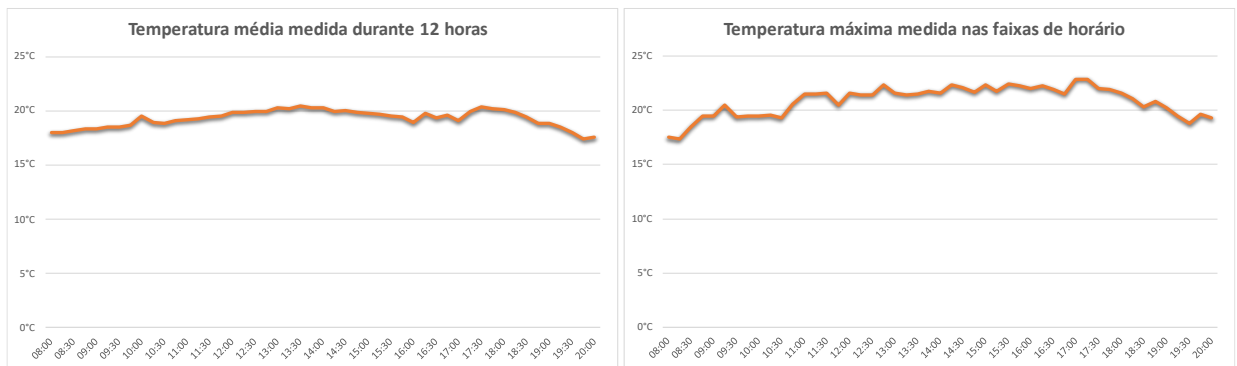
**Figura 19 – Parâmetros para os testes**

Planta 1	Planta 2	Planta 3
<b>Luminosidade</b> ☀ Valor Lido LED > 1000 Máxima capacidade < 999 e > 550 Metade da capacidade < 549 Desligado	<b>Luminosidade</b> ☀ Valor Lido LED > 650 Máxima capacidade < 649 Desligado	<b>Luminosidade</b> ☀ Valor Lido LED Qualquer valor Desligado
<b>Umidade do solo</b> 💧 Valor Lido Válvula < 20% Ligada > 21% Desligada	<b>Umidade do solo</b> 💧 Valor Lido Válvula < 40% Ligada > 41% Desligada	<b>Umidade do solo</b> 💧 Valor Lido Válvula < 60% Ligada > 61% Desligada
<b>Temperatura</b> 🌡 Valor Lido Ventoinha > 25°C Ligada < 24°C Desligada	<b>Temperatura</b> 🌡 Valor Lido Ventoinha > 30°C Ligada < 29°C Desligada	<b>Temperatura</b> 🌡 Valor Lido Ventoinha > 20°C Ligada < 19°C Desligada

Fonte: Autores

A Figura 20 apresenta os gráficos das temperaturas medidas durante o período de 12 horas. Para isso, foram lidos cerca de 44.000 valores pelo *SerialMonitor* do Arduino e realizado a separação entre faixas de 15 em 15 minutos com a média dos valores lidos nessas faixas de horário (12 horas possui 43200 segundos, sendo as faixas de 15 em 15 minutos com a média de 900 em 900 valores lidos).

**Figura 20 – Gráfico de Temperatura**

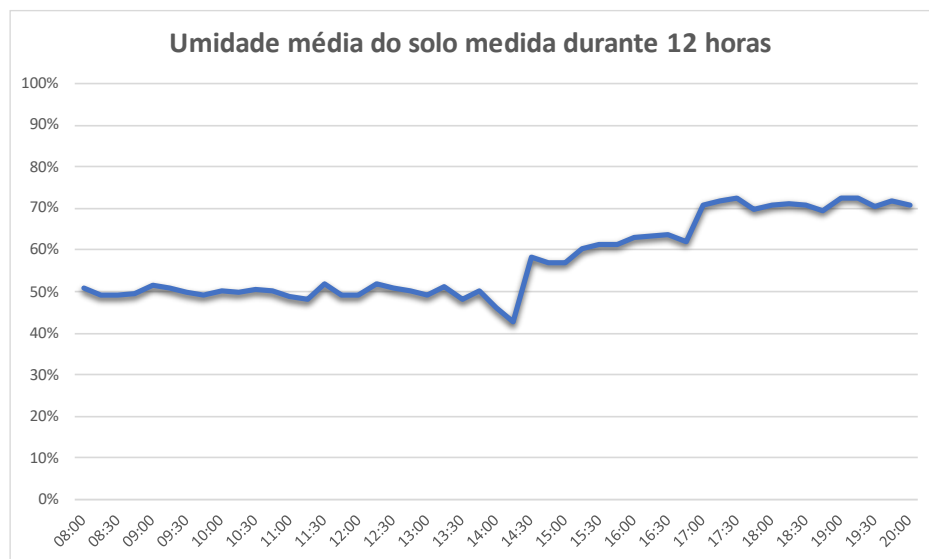


Fonte: Autores

Pode-se observar que a temperatura média foi crescente até por volta do meio-dia e depois estabilizou-se. Já as temperaturas máximas de cada faixa de horário sofreram algumas variações maiores. A temperatura média se manteve constante e sem muitas variações tendo em vista que o código do projeto realiza a média de 1000 leituras antes para exibir a temperatura lida (seja no *SerialMonitor* ou no LCD), juntamente com a proposta da estufa em reter certa quantidade de calor.

A Figura 21 apresenta o gráfico da umidade do solo medida durante o período de 12 horas. A forma no qual os valores foram lidos e separados em faixas de 15 minutos é do mesmo modelo que o da temperatura citado anteriormente. Nota-se que em momentos de pico de umidade (por volta de 14h30 e 17h00), o sistema de rega foi acionado manualmente pelo aplicativo, onde a leitura sobe repentinamente pelo contato do sensor direto com a água e depois estabiliza-se.

**Figura 21 – Gráfico de umidade do solo**

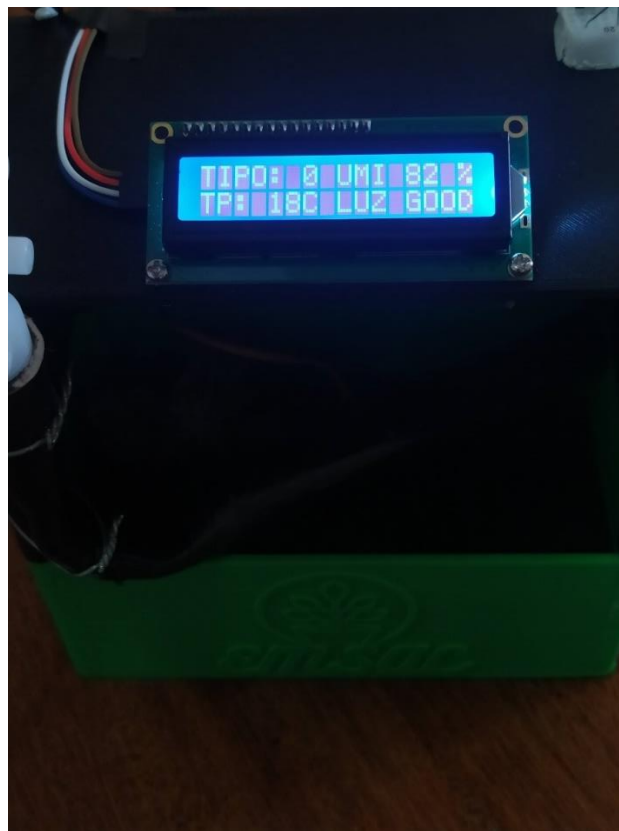


Fonte: Autores

Com todos os testes finalizados, o próximo passo foi de fazer a integração de todos os sistemas e do aplicativo. Após feita a integração de todas as partes do sistema (hardware e software), o protótipo foi mais uma vez testado para ver se iria se comportar

como o esperado. Finalizados os testes, o protótipo estava apto para entrar em funcionamento e, então, foi iniciado para realização da leitura dos sensores e funcionamento manual via aplicativo. A Figura 22 apresenta o protótipo em funcionamento, onde pode-se observar que no momento a umidade do solo estava em 82%, luminosidade boa e temperatura 18°C, modo manual ativado (tipo de planta 0).

**Figura 22 – Dados Lidos**



Fonte: Autores

Após realizar o funcionamento manual, entrou em funcionamento os modos automáticos (tipo de planta 1, 2 e 3). Primeiramente foi o atuador da temperatura que foi acionado, visto que a temperatura estava acima do informado no tipo de planta 1, sendo desligado após o valor de temperatura se estabilizar no indicado. Em seguida, foi testado o sensor de umidade do solo (que estava cravado na terra no interior do protótipo) e indicado que a umidade do solo estava em 50% (dentro do esperado, pois para esse tipo

1 de planta o atuador da rega só irá funcionar quando a umidade estiver em 20% ou menos). Com isso, foi feita a retirada do sensor do solo para prover uma diminuição da umidade, com isso o programa informou uma umidade de 0% e em seguida foi acionado o atuador da rega, recolocando o sensor na terra o programa desligou a irrigação porque a umidade do solo havia voltado ao valor adequado. Também foi testado a luminosidade do sistema, no qual para o tipo 1 de planta é ajustado com base na leitura do sensor. Para isso, foi utilizado o flash de um celular para diminuição do valor do sensor e aumento da luminosidade provida do LED. A Figura 23 apresenta a parte final do sistema de rega do protótipo, onde a água percorre o cano PVC furado para melhor distribuição de água, após a liberação da água pela válvula solenoide,

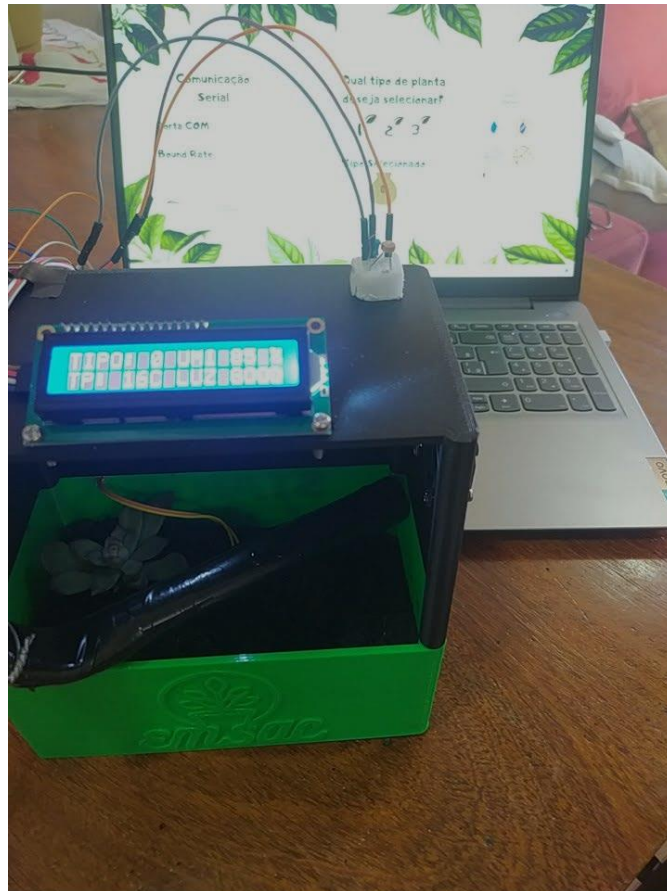
**Figura 23 – Sistema de rega**



Com os testes efetuados foi possível verificar que os todos os requisitos de funcionamento foram atendidos conforme o esperado. A Figura 24 apresenta os testes realizados e os resultados obtidos destes testes, provando que o protótipo é totalmente funcional, sendo observado o controle dos atuadores da estufa pelo aplicativo quando o modo manual está ativo.



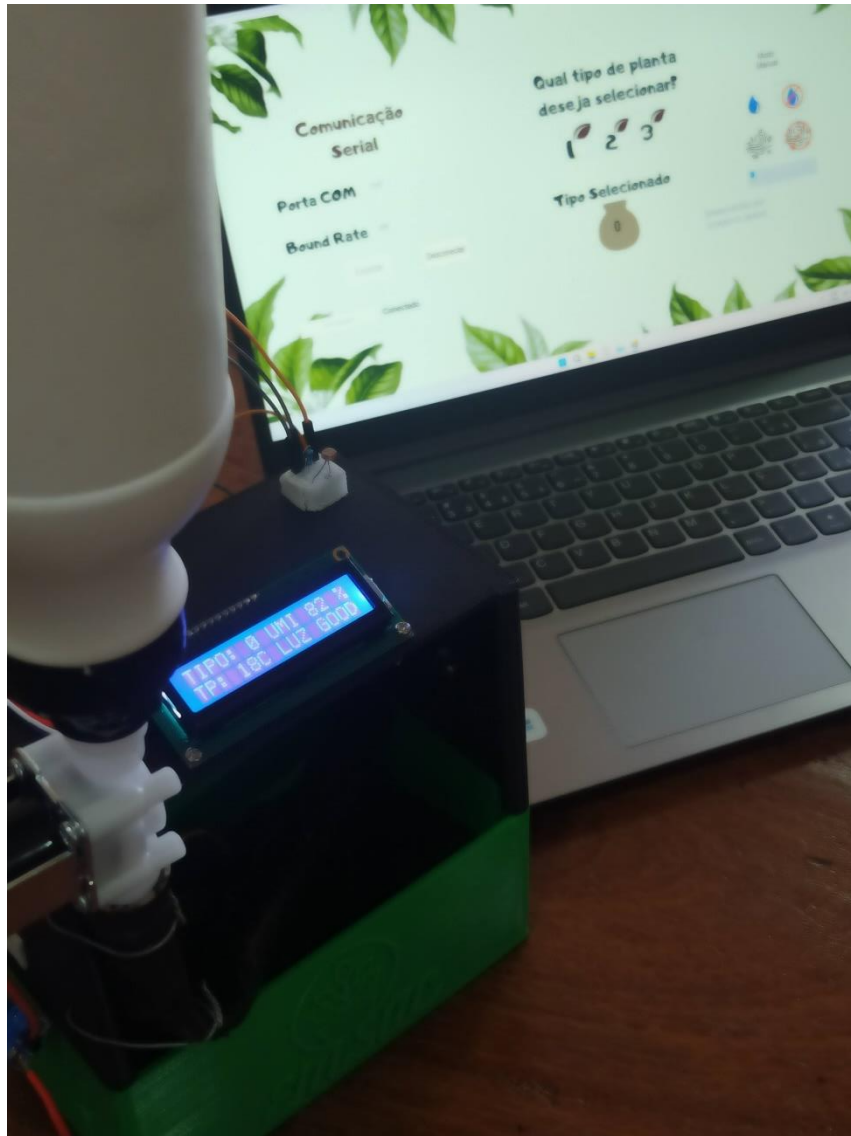
**Figura 24 – Testes**



Fonte: Autores

A Figura 25 apresenta, por outro ângulo, os testes realizados.

**Figura 25 – Protótipo final**



Fonte: Autores

### **3.5 Tabela de Custos**

A Tabela 2 será apresentada a relação de peças utilizadas no projeto e seus respectivos custos.

**Tabela 02 – Custos do projeto**

Sensor TMP36	R\$32,90
Sensor LM393	R\$17,87
Sensor LDR (5x)	R\$15,50
Protoboard (2x) + Jumpers (100x)	R\$39,90
Led 5mm alto brilho (10x)	R\$11,30
Módulo relé (2x)	R\$26,08
Display LCD 16x2	R\$22,99
Fonte DC 12V	R\$12,90
Solenóide 12V	R\$38,90
Arduino Uno R3 (placa compatível)	R\$62,90
Impressão em 3D	R\$95,00
Total	R\$376,24

Fonte: Autores

#### 4. CONCLUSÕES

A agricultura desempenha um papel fundamental no país, sendo a fonte de sustento para a maioria dos pequenos agricultores. Com base nisso, este projeto visou o desenvolvimento de um protótipo automatizado para estufas, que realiza automaticamente o controle da umidade do solo, luminosidade e temperatura. Essa inovação proporciona aos agricultores uma maneira mais conveniente de cuidar do plantio de hortaliças, promovendo o uso adequado dos recursos e reduzindo a necessidade de mão de obra.

Foi desenvolvido um software para o controle do Arduino, onde o usuário pode selecionar um tipo de planta entre os três disponíveis ou realizar seu controle manual, podendo acionar quando quiser o sistema de ventilação e o sistema de rega do sistema. Para visualização dos dados, utilizou-se um LCD que mostra as leituras dos sensores e qual tipo de planta está selecionado. Com base nesses dados, o sistema realiza os acionamentos dos atuadores quando necessário. Foi construído um protótipo de uma estufa simulando uma plantação, onde é possível visualizar toda a estrutura pretendida pelo projeto e suas funcionalidades.

Os dispositivos que compõem o protótipo foram submetidos a testes individuais para compreender suas funcionalidades e a comunicação com o controlador. Para cada componente desenvolvido, um código de programação de teste foi criado. Após o teste individual de todos os itens, deu-se início à integração dos equipamentos, elaborando os códigos definitivos do protótipo. Após a conclusão do protótipo, foram realizados diversos testes e validações. Durante esse processo, foram identificadas algumas necessidades de ajustes tanto no software quanto no hardware, com o objetivo de alcançar os objetivos propostos.

Uma das dificuldades encontradas no desenvolvimento do projeto foi no desenvolvimento do código do aplicativo no Microsoft Visual Studio 2022, onde foi

necessário aprofundar alguns temas como linguagem de programação C# e .NET Framework, juntamente com a programação orientada a objetos.

Outra dificuldade encontrada foi no sistema de irrigação, onde havia a dúvida entre utilizar o sistema com *timers* do Arduino para realizar a rega em tempos certos. Entretanto, foi concluído que em dias quentes a água do solo irá evaporar mais rapidamente, ou seja, a planta necessitará de água mais vezes. Uma vez que o sistema de *timer* não supriria esta dificuldade, houve a mudança do sistema para rega com base na porcentagem de umidade do solo.

No fim foi possível analisar com o protótipo criado, que o sistema atendeu os objetivos estabelecidos, permitindo a criação de um sistema automatizado de controle de estufas para cultivo de plantas, baseado nos comandos realizados via aplicativo web, que permite a interação instantânea com os sensores e atuadores e empregou diferentes temas abordados pelo curso de Eletrônica Industrial, como a lógica de programação, eletrônica, microcontroladores e instrumentação eletrônica e sensores. Além disso, outro objetivo atingido foi a questão de ser um projeto de baixo custo, tendo em vista que uma estufa comercial custa cerca de R\$1.300,00 sem sistemas de rega e ventilação, apenas de sistema de iluminação.

#### 4.1 **Projetos Futuros**

Embora o projeto possa auxiliar o trabalho de agricultura dentro da casa do usuário nos processos de controle de temperatura, luminosidade e rega, ele pode ser evoluído em diversos fatores. Dentre esses fatores, podemos destacar o acréscimo de um sensor que mede a quantidade de água no reservatório e alerte o usuário para que preencha conforme o adequado, acréscimo de um sensor que mede o pH do solo indicando se a quantidade de sais minerais presentes é suficiente para um bom crescimento da planta e, por fim, troca de sensores para outros mais sofisticados e com maior precisão.

Outros pontos a serem melhorados é a elaboração de um aplicativo que permita que o usuário defina suas variáveis de controle da estufa, como por exemplo a temperatura e umidade do solo desejados, juntamente com aprimorar a comunicação dos sensores com o Arduino, substituindo os cabos por transmissão em radiofrequência.

## 5. REFERÊNCIAS

ABRAMOVAY, Ricardo. **Alimentos versus população: está ressurgindo o fantasma malthusiano?**. Cienc. Cult., São Paulo, v. 62, n. 4, p. 38-42, 2010. <[http://cienciaecultura.bvs.br/scielo.php?script=sci\\_arttext&pid=S0009-67252010000400013&lng=en&nrm=iso](http://cienciaecultura.bvs.br/scielo.php?script=sci_arttext&pid=S0009-67252010000400013&lng=en&nrm=iso)>. Acesso em: 09/03/2023

TIVELLI, Sebastião Wilson. **Orgânicos são caros. Por quê.** Pesquisa & Tecnologia, v. 9, n. 1, 2012.

OLIVEIRA, K. C., FACHINELLO, S. L., **Crescimento da demanda mundial de alimentos e restrições do fator terra no Brasil**, Rev. Econ. Sociol. Rural 56 (2), p. 195-212, 2018 <https://doi.org/10.1590/1234-56781806-94790560201>

GUEDES, B. C.; SOUSA, G. F. V. de. **Estufa agrícola automatizada utilizando micro controlador arduíno e comunicação serial**. In: VII Congresso Técnico Científico da Engenharia e da Agronomia. [S.l.: s.n.], 2016.

MICROCHIP. Datasheetcatalog, 2022. **Datasheet do microcontrolador PIC16F877A**. Disponível em: <<http://pdf.datasheetcatalog.net/datasheet/microchip/33023a.pdf>>. Acesso em: 09/03/2023.

ARDUINO. Octopart, 2022. **Datasheet do microcontrolador Arduino UNO**. Disponível em: <<https://datasheet.octopart.com/A000066-Arduino-datasheet-38879526.pdf>>. Acesso em: 12/03/2023.

SNIS159H. Texas Instruments, 2017. **Datasheet do sensor LM35 de temperatura.** Disponível em: <<https://www.ti.com/lit/ds/symlink/lm35.pdf>>. Acesso em 19/03/2023.

MOTOROLA. Adrobotica, 2022. **Datasheet do sensor LM393 de umidade do solo.** Disponível em: <<https://adrobotica.com/wp-content/uploads/2018/02/datasheet-LM393.pdf>>. Acesso em 19/03/2023.

CDS PHOTOCONDUCTIVE CELLS. Filipeflop, 2022. **Datasheet do sensor GL5528 de luminosidade.** Disponível em: <[https://www.filipeflop.com/img/files/download/Datasheet\\_LDR.pdf](https://www.filipeflop.com/img/files/download/Datasheet_LDR.pdf)>. Acesso em: 26/03/2023.

PARKER. Parker, 2002. **Datasheet da válvula Solenoide.** Disponível em: <[https://www.parker.com/parkerimages/br/download/cic/pdf/cat\\_4201\\_2.pdf](https://www.parker.com/parkerimages/br/download/cic/pdf/cat_4201_2.pdf)>. Acesso em: 08/04/2022.

FONSECA, E. G. P. da; BEPPU, M. M. **Apostila Arduino.** Niterói-RJ: Universidade Federal Fluminense Centro Tecnológico, 2010. 23 p.

MCROBERTS, Michael. **Beginning Arduino**, v. 1, n. 1, 2011.

MCROBERTS, Michael. **Beginning Arduino**, v. 2, n. 1, 2015.



DUTRA, Lang e WAROMBY, 2013. Augusto Fellipe Alves, Arthur Guilherme e Paulo Henrique. **Desenvolvimento de um Irrigador Microcontrolado Integrado a um Sistema de Supervisão e Controle**. Curitiba: Universidade Tecnológica Federal do Paraná.

SANTOS (2014), Lorena Beatriz. **Sistema Automatizado para Controle de Umidade e Temperatura em Cultura de Morangos Aplicados aos Pequenos Produtores**. Brasília: Centro Universitário de Brasília.

RICARTE, Ivan L. M. 2013. **Programação de Sistemas: Uma Introdução – Programação Estruturada**. Campinas: Universidade Estadual de Campinas.

ARDUINO, 2023. *SoftwareSerial Library*. Disponível em: <https://docs.arduino.cc/learn/built-in-libraries/software-serial>. Acesso em: 06/05/2023.

MEGMENG, LI, LI, TANG, 2018. Gu, Qiansheng, Xiaoqiang, Bin. ***Growth Responses and Root Characteristics of Lettuce Grown in Aeroponics, Hydroponics, and Substrate Culture***. Disponível em: <https://www.mdpi.com/2311-7524/4/4/35/htm#B1-horticulturae-04-00035>. Acesso em: 13/06/2023.

FRANCHINI, J.C, 2011. COSTA, J. M. da; DEBIASI, H.; TORRES, E.; **Importância da rotação de culturas para a produção agrícola sustentável no Paraná**. Documentos 327. Embrapa Soja. Londrina, PR.

WIRTH LIMA, Almir. **Eletricidade e eletrônica básica**. v. 1, n. 4, 2013.

TEHERAN E ROJAS. *Hands-On Visual Studio 2022*. v. 1, n. 1, 2022.

## 6. APÊNDICES

### APÊNDICE A – Bloco 01 de códigos do projeto

```
void inicializacao(){ //inicialização do LCD e com o setup pronto, só irá entrar os valores depois

lcd.setCursor(0,0);
//      0123456789012345
lcd.print(" PROJETO ESTUFA ");
lcd.setCursor(0,1);
lcd.print("   FATEC-SP   ");
delay(2000);
lcd.clear();

lcd.setCursor(0,0);
lcd.print("   AUTORES   ");
lcd.setCursor(0,1);
lcd.print("GUSTAVO E JACSON");
delay(1000);
lcd.clear();

lcd.setCursor(0,0);
//      0123456789012345
lcd.print("TIPO: ");
lcd.setCursor(8,0);
lcd.print("UMI");
lcd.setCursor(15,0);
lcd.print("%");
lcd.setCursor(0,1);
lcd.print("TP:");
lcd.setCursor(6,1);
lcd.print("C");
lcd.setCursor(8,1);
lcd.print("LUZ");
}
```

### APÊNDICE B – Bloco 02 de códigos do projeto

```
void sensorTMP36(){ //bloco para funcionamento do sensor de temperatura
float celsius = myTMP36.getTempC(); //VARIÁVEL LOCAL QUE ARMAZENA A TEMPERATURA EM GRAUS CELSIUS
//OBTIDA ATRAVÉS DA FUNÇÃO myTMP36.getTempC()

for (int i=0;i<1000;i++) { //Para evitar as grandes variações de leitura do componente são feitas 1000 leituras onde calculamos a média
  temperatura = temperatura + celsius;
}
temperatura = temperatura/1000; // calcula a média das leituras

delay(1000);
}

void sensorLM393(){ //bloco para funcionamento do sensor de umidade do solo
umidadesolo = constrain(analogRead(pin_lm393),analogSoloMolhado,analogSoloSeco); //MANTÉM umidadesolo DENTRO DO INTERVALO (ENTRE analogSoloMolhado E analogSoloSeco)
umidadesolo = map(umidadesolo,analogSoloMolhado,analogSoloSeco,percSoloMolhado,percSoloSeco); //EXECUTA A FUNÇÃO "map" DE ACORDO COM OS PARÂMETROS PASSADOS
delay(1000);
}

void sensorLDR(){
luminosidade=analogRead(pin_ldr); //Lê o valor do sensor ldr e armazena na variável luminosidade
delay(1000);
}
```

## APÊNDICE C – Bloco 03 de códigos do projeto

```
void resultadosLCD(){  
  
    lcd.setCursor(12,0);  
    lcd.print("  ");  
    lcd.setCursor(12,0);  
    lcd.print(umidadesolo);  
  
    lcd.setCursor(6,0);  
    lcd.print(tipodeplanta);  
  
    lcd.setCursor(4,1);  
    lcd.print("  ");  
    lcd.setCursor(4,1);  
    lcd.print(temperatura,0);  
  
    if(luminosidade>1000)  
    {  
        lcd.setCursor(11,1);  
        lcd.print("  ");  
        lcd.setCursor(12,1);  
        lcd.print("LOW");  
    }  
    if(luminosidade<999 && luminosidade>550)  
    {  
        lcd.setCursor(11,1);  
        lcd.print("  ");  
        lcd.setCursor(12,1);  
        lcd.print("GOOD");  
    }  
    if(luminosidade<549)  
    {  
        lcd.setCursor(11,1);  
        lcd.print("  ");  
        lcd.setCursor(12,1);  
        lcd.print("HIGH");  
    }  
}
```

## APÊNDICE D – Bloco 04 de códigos do projeto

```
void tipos_plantas(){  
  
  Serial.available();  
  cmd=Serial.read();  
  |  
  if(cmd=='a'){ //configuração do comando para ativar modo de planta 1  
    botaoPlanta_1=true;  
    botaoPlanta_2=false;  
    botaoPlanta_3=false;  
  }  
  
  if(cmd=='b'){ //configuração do comando para ativar modo de planta 2  
    botaoPlanta_2=true;  
    botaoPlanta_1=false;  
    botaoPlanta_3=false;  
  }  
  
  if(cmd=='c'){ //configuração do comando para ativar modo de planta 3  
    botaoPlanta_3=true;  
    botaoPlanta_1=false;  
    botaoPlanta_2=false;  
  }  
  
  if(cmd=='x'){ //configuração do comando para ativar modo manual  
    botaoPlanta_3=false;  
    botaoPlanta_1=false;  
    botaoPlanta_2=false;  
    tipodeplanta=0;  
    dutyLUZ=0;  
    analogWrite(Led,dutyLUZ);  
  }  
  
  if(cmd=='z'){ //configuração do comando para ativar a rega  
    digitalWrite(Motor, LOW); //Altera estado do pino para ativar a solenoide  
  }  
}
```

```

if(cmd=='y'){ //configuração do comando para desativar a rega
digitalWrite(Motor, HIGH); //Altera estado do pino para desativar a solenoide
}

if(cmd=='d'){ //configuração do comando para ativar o fan
digitalWrite(Ventoinha, LOW); //Altera estado do pino para ativar o fan
}

if(cmd=='e'){ //configuração do comando para desativar o fan
digitalWrite(Ventoinha, HIGH); //Altera estado do pino para desativar o fan
}

if(cmd=='f'){ //configuração do comando para aumentar a luz
dutyLUZ=dutyLUZ+25;
analogWrite(Led,dutyLUZ); //Altera estado do pino para aumentar a luz
}

if(cmd=='g'){ //configuração do comando para diminuir a luz
dutyLUZ=dutyLUZ-25;
analogWrite(Led,dutyLUZ); //Altera estado do pino para aumentar a luz
}
}

```

## APÊNDICE E – Blocos 05, 06 e 07 de códigos do projeto

```

void Planta1(){ //Planta 1 possui temperatura ideal por volta de até 25°C e luminosidade ajustável com a externa
dutyLUZ=0;
  if(luminosidade>1000)
  {
dutyLUZ=255;
analogWrite(Led,dutyLUZ);
  }
  if(luminosidade<999 && luminosidade>550)
  {
dutyLUZ=100;
analogWrite(Led,dutyLUZ);
  }
  if(luminosidade<549)
  {
dutyLUZ=0;
analogWrite(Led,dutyLUZ);
  }

  tipodeplanta=1;

  if(temperatura>25)
  {
digitalWrite(Ventoinha, LOW);
  }
  else
  {
digitalWrite(Ventoinha, HIGH);
  }

  if(umidadesolo<20) ///se umidade do solo abaixo de 20%, liga a valvula solenoide até estabilizar
  {
digitalWrite(Motor, LOW); //Altera estado do pino
  }
  else
  {
digitalWrite(Motor, HIGH); //Altera estado do pino
  }
}

```

```

void Planta2(){ //Planta 2 possui temperatura ideal por volta de até 30°C e luminosidade alta
dutyLUZ=0;
  if(luminosidade>650)
  {
dutyLUZ=255;
analogWrite(Led,dutyLUZ);
  }
  else
  {
dutyLUZ=0;
analogWrite(Led,dutyLUZ);
  }

  tipodeplanta=2;

  if(temperatura>30)
  {
digitalWrite(Ventoinha, LOW);
  }
  else
  {
digitalWrite(Ventoinha, HIGH);
  }

  if(umidadesolo<40) ///se umidade do solo abaixo de 40%, liga a valvula solenoide até estabilizar
  {
digitalWrite(Motor, LOW); //Altera estado do pino
  }
  else
  {
digitalWrite(Motor, HIGH); //Altera estado do pino
  }
}

```

```

void Planta3(){ //Planta 3 possui temperatura ideal por volta de até 20°C e luminosidade apenas diaria
  dutyLUZ=0;
  analogWrite(Led,dutyLUZ);

  tipodeplanta=3;

  if(temperatura>20)
  {
    digitalWrite(Ventoinha, LOW);
  }
  else
  {
    digitalWrite(Ventoinha, HIGH);
  }

  if(umidadesolo<60) //se umidade do solo abaixo de 60%, liga a valvula solenoide até estabilizar
  {
    digitalWrite(Motor, LOW); //Altera estado do pino
  }
  else
  {
    digitalWrite(Motor, HIGH); //Altera estado do pino
  }
}

```

## APÊNDICE F – Blocos 08 de códigos do projeto

```

void modos_pin(){ //void para configurar os modos dos pinos
  pinMode(pin_lm35, INPUT); // Define o sensor como uma entrada de sinal
  pinMode(pin_ldr, INPUT); // Define o sensor como uma entrada de sinal
  pinMode(pin_lm393, INPUT); // Define o sensor como uma entrada de sinal
  pinMode(Led, OUTPUT); // Define o LED como uma saída de sinal
  pinMode(Ventoinha, OUTPUT); // Define a VETOINHA como uma saída de sinal
  pinMode(Motor, OUTPUT); // Define o MOTOR como uma saída de sinal
}

```



## APÊNDICE G – Blocos 09 de códigos do projeto

```
void setup() {  
  
  Serial.begin(9600); // Inicia a comunicação  
  
  lcd.init();          // Inicializando o LCD  
  lcd.backlight();    // Ligando o BackLight do LCD  
  
  modos_pin();  
  
  inicializacao(); //void para o setup do LCD  
  
}  
  
void loop() {  
  
  sensorTMP36();  
  sensorLM393();  
  sensorLDR();  
  resultadosLCD();  
  tipos_plantas();  
  
  if (botaoPlanta_1==true){  
    Planta1();  
  }  
  
  if (botaoPlanta_2==true){  
    Planta2();  
  }  
  
  if (botaoPlanta_3==true){  
    Planta3();  
  }  
  
}
```

## APÊNDICE H – Circuito de Testes simulado no Proteus

