

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
BACHARELADO EM ENGENHARIA ELETRÔNICA**

VALDEMIR MESKIV

**AUTOMAÇÃO DE UMA ESTUFA AGRÍCOLA DESTINADA A
PRODUÇÃO DE MUDAS DE EUCALIPTO**

TRABALHO DE CONCLUSÃO DE CURSO

**CAMPO MOURÃO
2020**

VALDEMIR MESKIV

**AUTOMAÇÃO DE UMA ESTUFA AGRÍCOLA DESTINADA A PRODUÇÃO DE
MUDAS DE EUCALIPTO**

**Automation Of An Agricultural Greenhouse For The Production Of Eucalyptus
Seedlings**

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Márcio Rodrigues da Cunha

CAMPO MOURÃO

2020



TERMO DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO
INTITULADO

**AUTOMAÇÃO DE UMA ESTUFA AGRÍCOLA DESTINADA A
PRODUÇÃO DE MUDAS DE EUCALIPTO**

DO DISCENTE

VALDEMIR MESKIV

Trabalho de Conclusão de Curso apresentado no dia 30 de novembro de 2020 ao Curso Superior de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná, Campus Campo Mourão. O(A) discente foi arguido(a) pela Comissão Examinadora composta pelos professores abaixo assinados. Após deliberação, a comissão considerou o trabalho aprovado com alterações.

Prof. Dr. Gilson Junior Schiavon

Avaliador(a) 1
UTFPR

Prof. Me. Reginaldo Nunes De Souza

Avaliador(a) 2
UTFPR

Prof. Dr. Marcio Rodrigues da Cunha

Orientador(a)
UTFPR

AGRADECIMENTO

Gostaria de agradecer em primeiro lugar a Deus, pela vida, e por permitir ultrapassar todos os obstáculos ao longo do curso.

Agradeço também aos meus pais e meu irmão por todo o apoio e compreensão durante essa jornada.

Agradeço também ao meu orientador o Professor Doutor Márcio Rodrigues da Cunha, pela ajuda desde a escolha do tema até a conclusão do trabalho.

Agradeço também aos meus parentes, amigos, colegas de curso e professores que de alguma forma me ajudaram a atingir os meus objetivos.

E agradeço especialmente a minha esposa Patrícia Wynnek Meskiv por toda a ajuda, conselhos e incentivos, pois sem ela eu não teria conseguido chegar nesse momento.

A todos muito obrigado!

RESUMO

Este projeto tem por objetivo a automatização de uma estufa destinada a produção de mudas de eucalipto, para possibilitar o controle das variáveis internas dessa estufa. Essa estufa tem como o objetivo a produção de mudas clonadas de eucalipto, pois estas árvores clonadas apresentam uma maior produtividade em relação as mudas convencionais. O projeto conta com dois ESP's32 que são conectados através do protocolo ESP-NOW, e sensores que fazem a captação dos dados de temperatura, umidade e luminosidade da estufa. Um dos ESP's32 faz a leitura constante dos sensores, mostra os valores obtidos em um display, esses valores são mostrados apenas localmente, ele faz a análise desses dados, que são enviados através do protocolo ESP-NOW para o outro ESP32 ligar ou desligar os atuadores. Após testes realizados chegou-se à conclusão de que o projeto funcionou adequadamente e atingiu os objetivos propostos, podendo ser expandido para a utilização em um projeto em escala real.

Palavras-chave: Microcontrolador ESP32. Protocolo ESP-NOW. Estufa. Eucalipto.

ABSTRACT

This project aims to automate a greenhouse for the production of eucalyptus seedlings, to enable control of the internal variables of that greenhouse. This greenhouse aims to produce cloned eucalyptus seedlings, as these cloned trees have a higher productivity compared to conventional seedlings. The project has two ESP's32 that are connected through the ESP-NOW protocol, and sensors that capture the temperature, humidity and luminosity data of the greenhouse. One of the ESP's32 constantly reads the sensors, shows the values obtained on a display, these values are shown only locally, it analyzes these data, which are sent via the ESP-NOW protocol to the other ESP32 on or off the actuators. After tests carried out, it was concluded that the project worked properly and achieved the proposed objectives, which can be expanded for use in a full-scale project.

Keywords: ESP32 microcontroller. ESP-NOW protocol. Greenhouse. *Eucalyptus*.

LISTA DE FIGURAS

Figura 1 Placa de desenvolvimento ESP32	18
Figura 2 Arduino IDE	20
Figura 3 Sensor DHT 22	21
Figura 4 Sensor LDR	22
Figura 5 Módulo relé	23
Figura 6 Sensor de vazão	23
Figura 7 Diagrama de blocos do sistema	24
Figura 8 Diagrama elétrico do ESP32 mestre	25
Figura 9 Diagrama elétrico do ESP32 escravo	25
Figura 10 Instalação do pacote ESP32 na IDE Arduino	27
Figura 11 Tela de configuração	27
Figura 12 Site para baixar pacote do ESP32	28
Figura 13 Tela gerenciador de placas	28
Figura 14 Instalando o pacote ESP32	29
Figura 15 Selecionando a placa ESP 32 DeyModule	29
Figura 16 Instalando a biblioteca do sensor DHT 22	30
Figura 17 Instalando a biblioteca do display	30
Figura 18 Fluxograma completo do sistema	31
Figura 19 Fluxograma ESP32 master	31
Figura 20 Código mestre, inclusão das bibliotecas e pinos	32
Figura 21 Código mestre, função setup	33
Figura 22 Código mestre, função InitESPNow	33
Figura 23 Código mestre, leitura dos sensores	34
Figura 24 Código mestre, comparação da umidade	35
Figura 25 Código mestre, envio dos dados	35
Figura 26 Código mestre, função loop	36
Figura 27 Fluxograma ESP32 master	36
Figura 28 Código escravo, inclusão das bibliotecas	37
Figura 29 Código escravo, função OnDataRecv	37
Figura 30 Fluxograma de acionamento dos atuadores	38
Figura 31 Foto do protótipo da estufa pronta	38

Figura 32 Atuadores utilizados no projeto. A) Ventilador; B) Bomba d'água; C) Lâmpada	39
Figura 33 Display LCD	39
Figura 34 Gráfico do sensor LDR	41
Figura 35 Acionamento lâmpada/led	42
Figura 36 Gráfico do sensor DHT 22 para temperatura	42
Figura 37 Acionamento dos ventiladores	43
Figura 38 Gráfico do sensor DHT 22 para umidade	44
Figura 39 Acionamento da bomba d'água	44
Figura 40 Gráfico do sensor de vazão	45
Figura 41 Gráfico do sistema de umidade completo	45
Figura 42 A) Luminosidade em excesso; B) Luminosidade muito baixa	46
Figura 43 Led indicando o sistema de ventilação ligado	47
Figura 44 Led indicando o sistema de irrigação ligado	47
Figura 45 Dados apresentados no display LCD	48

LISTA DE QUADROS

Quadro 1 Características dos componentes	26
---	----

SUMÁRIO

1 INTRODUÇÃO	12
1.1 Objetivos	14
1.1.1 Objetivo geral	14
1.1.2 Objetivos específicos	14
1.2 Justificativa	14
2 REFERENCIAL TEÓRICO.....	16
2.1 Temperatura	16
2.2 Umidade	17
2.3 Ventilação	17
2.4 Luminosidade	17
2.5 Irrigação	18
2.6 Placa eletrônica ESP32	18
2.7 IDE Arduino	19
2.8 Protocolo ESP-NOW	20
2.9 Sensor DHT 22	21
2.10 Sensor LDR	22
2.11 Módulo relé	22
2.12 Sensor de vazão de água	23
3 MATERIAIS E MÉTODOS	24
3.1 Desenvolvimento do <i>Hardware</i>	24
3.2 Desenvolvimento do <i>Software</i>	26
3.3 Código mestre	31
3.4 Código escravo	36
3.5 Construção da estufa	38
3.6 Planejamento de testes	39
4 RESULTADOS	41
4.1 Luminosidade	41
4.2 Temperatura	42
4.3 Umidade	43
4.4 Simulações	46
4.4.1 Luminosidade	46
4.4.2 Temperatura	46

4.4.3 Umidade	47
4.4.4 Display	48
5 CONCLUSÃO	49
REFERÊNCIAS	50
APÊNDICE I - CÓDIGO MESTRE	55
APÊNDICE II - CÓDIGO ESCRAVO	61

1. INTRODUÇÃO

Com a evolução do ser humano, desde a época dos homens da caverna até os dias atuais, o homem desenvolveu a capacidade de pensar e raciocinar, buscando meios de melhorar e facilitar a sua vida, com a criação de máquinas e equipamentos, como por exemplo, a máquina a vapor, computadores, controladores lógicos programáveis (CLP), entre outras invenções que ocorreram ao longo da evolução humana (SILVA, 2013).

Neste contexto, uma das áreas desenvolvidas pelo ser humano foi a agricultura, seja para alimentação, produção de energia, construção, dentre outras aplicações. Dentre as evoluções tecnológicas observadas na agricultura, destaca-se aqui os ambientes de cultivo controlado, as chamadas estufas, as quais visam manter os parâmetros ótimos de cultivo, além de proteger o cultivo de fatores ambientes externos.

Segundo LITJENS (2009), as primeiras estufas modernas surgiram na Europa no século XIII, pois eles necessitavam proteger as plantas do inverno rigoroso. Logo, as estufas se espalharam pelo mundo, e evoluíram com a tecnologia.

Nos últimos 150 anos houve diversos avanços na ciência básica e aplicada, permitindo maiores conhecimentos sobre solo e meio ambiente, e por consequência obteve-se grande desenvolvimento na agricultura. A alta tecnologia quando está unida a geração e a utilização de informações, pode permitir que se tenha uma produção maior, porém com menos destruição do meio ambiente, pois assim é possível obter uma produção com menos água, adubo e mão de obra (REIFSCHNEIDER, HENZ, RAGASSI, et al, 2010).

O Brasil possuiu muitas variações climáticas, como amplitudes térmicas, altas temperaturas, frio extremo, períodos de seca, temporais de granizo, doenças e pragas nas lavouras. O uso de estufas com sistema automatizado vem crescendo cada vez mais, pois assim o produtor tem maior facilidade nos cuidados da produtividade (FERNANDES, PREUSS, SILVA, 2017).

Visto que com a estufa automatizada é possível ter um ambiente protegido, com diminuição de riscos, diminuição de mão de obra e redução da perda dos recursos. A automação nestas estruturas fechadas permite o controle da umidade, luminosidade e temperatura, ou seja, proporciona maior desenvolvimento da

produtividade e como consequência qualidade dos produtos com maior eficácia e uso lógico dos recursos (FERNANDES, PREUSS, SILVA, 2017).

Segundo BISON (2004), a partir de 1904, iniciou-se os primeiros estudos com a cultura do eucalipto. Porém, esses plantios apresentavam baixa qualidade. Por isso, em 1941, iniciou-se o programa de melhoramento genético do eucalipto para a Companhia Paulista de Estradas de Ferro, com o objetivo de melhorar a uniformidade das plantas, reduzir as falhas nas plantações, melhorar o crescimento em altura e diâmetro, aumentar a capacidade de brotação após o corte e aumentar a produtividade.

As árvores de eucalipto clonadas, contribuíram para significativo aumento no volume de madeira e produção de celulose por hectare. Bem como a produtividade da madeira que era de 20 m³/ha/ano em 1960, passou para 40 m³/ha/ano em 1998 (BISON, 2004).

Segundo ABRAF (2011), no ano de 2010, árvores como o eucalipto e o pinus representavam uma área plantada de 6.510.693 ha, dos quais 73% correspondente a área de plantios de eucalipto e o restante a plantios de pinus.

A produção de celulose brasileira teve um crescimento de 4,5% no ano de 2010, em relação ao ano anterior. Fazendo o país chegar ao 4º lugar do ranking mundial dos produtores de celulose, totalizando uma produção de 14,1 milhões de toneladas (ABRAF, 2011).

O eucalipto é originário da Austrália, e possui mais de 600 espécies. Seu plantio é amplo no Brasil, por causa do seu crescimento rápido, baixo custo e madeira de boa qualidade. A espécie escolhida para escolha de valores de umidade e temperatura da estufa foi o *Eucaliptus urograndis*, híbrido resultante do cruzamento entre o *Eucaliptus grandis* x *Eucalipto urophylla*, tem como característica o desenvolvimento acelerado, e tolerância as doenças e as adversidades climáticas (Benatti et al., 2013). Pelas suas qualidade e características ele é amplamente cultivado no Brasil.

Logo, este trabalho articulará o uso de tecnologia aplicada ao cultivo de eucalipto em estufas, contribuindo com o desenvolvimento tecnológico dessa área agrícola.

1.1. Objetivos

1.1.1 Objetivo Geral

Implementar a automação de uma estufa agrícola usada para produção de mudas de eucalipto. O projeto consiste em analisar e controlar os parâmetros de uma estufa, que são, temperatura, umidade e luminosidade, deixando-os em uma faixa adequada para o melhor enraizamento e crescimento das plantas. Os atuadores utilizados para tal controle são: ventiladores e bomba d'água, além da lâmpada, que é uma carga resistiva. Para interligar os sensores e atuadores foi utilizado o microcontrolador ESP32.

1.1.2 Objetivos Específicos

Para atingir o objetivo geral, os objetivos específicos devem ser alcançados, que englobados resultam no objetivo geral do projeto. Esses objetivos são:

- 1)** Avaliar os sensores necessários para a automação da estufa;
- 2)** Avaliar a implementação do sistema usando o microcontrolador ESP32 e o protocolo ESP-NOW;
- 3)** Monitorar a luminosidade, temperatura e umidade da estufa por meio dos sensores definidos;
- 4)** Processar os dados obtidos dos sensores;
- 5)** Enviar sinais para o acionamento dos atuadores da estufa;

1.2 Justificativa

Atualmente, muitas empresas necessitam de matéria prima extraída da natureza, tanto para a comercialização como para serem usadas em fornos e caldeiras como lenha (SEBRAE, 2013). Como não é possível fazer essa extração de florestas nativas, uma saída é utilizar as árvores de eucalipto. Justamente por não poder extrair árvores da natureza, o consumo de eucalipto aumentou, que mesmo sendo uma

árvore de crescimento rápido, viu-se a necessidade de aumentar a produção desse cultivo.

Uma das alternativas é a produção de mudas clonadas, feitas a partir de uma matriz. Essas mudas, além de ter um crescimento mais rápido que as mudas convencionais, também têm uma maior padronização das árvores, ou seja, as árvores crescem praticamente iguais, não tendo árvores menores e maiores espalhadas na plantação, tendo assim uma maior produção.

O processo de obtenção de mudas, a partir de partes de uma planta, chama-se estaquia. Consiste em cortar galhos de 08 a 10 cm de comprimento e deixar de um a três pares de folhas, que devem ser recortadas pela metade. Depois essas mudas devem ser mantidas em um ambiente fechado com alta umidade do ar, ou seja, uma estufa (SANTOS, XAVIES, OLIVEIRA, et al, 2005). Para a espécie de eucalipto escolhida essa umidade do ar na estufa deve ficar na faixa de 70 % a 100 %, e a temperatura deve estar entre 23 ° e 30 °C.

Apesar da definição de estufa ser manter o calor em seu interior, uma estufa agrícola tem como objetivo, além de controlar a temperatura, a criação de um microclima completo a fim de favorecer o crescimento das plantas produzidas na estufa.

Este microclima é um conjunto de variáveis ambientais como temperatura, luminosidade, umidade relativa do ar, nível de CO₂, dentre outras (LITJENS, 2009). As variáveis desse microclima podem ser controladas a fim de melhorar ainda mais a produção.

As estufas não são usadas, única e exclusivamente, para produção de alimentos, mas são usadas também para cultivo de plantas ornamentais, flores e plantas medicinais, produção de mudas, dentre outras aplicações (MARANGONI, MOREIRA, SOUZA, 2014).

As principais vantagens da produção de mudas de eucalipto em estufa são: economia de água, proteção contra intempéries e mudas de melhor qualidade.

2 REFERENCIAL TEÓRICO

As estufas são instrumentos de proteção ambiental para produção de plantas, como hortaliças e flores. Por definição, estufas são estruturas construídas com diversos materiais, como madeira, concreto, ferro, alumínio, etc. São cobertas com materiais transparentes que permitam a passagem da luz solar para crescimento e desenvolvimento das plantas (REIS, 2005).

Além de proteção das plantas em relação a parâmetros meteorológicos adversos, como a precipitação pluviométrica, ela gera e aprisiona o calor dentro do seu interior, ou seja, não há troca de ar entre o interior e o exterior, sendo assim a energia que entra pela radiação solar aquece o ambiente interno e não é perdida com as correntes ascendentes, que dissipariam o calor (GLÓRIA, NAKA, BELONI, et al, 2015).

Grande parte das plantas são sensíveis às mudanças bruscas dos elementos do tempo. As orquídeas, por exemplo, necessitam de condições apropriadas para o seu desenvolvimento e alguns fatores, como luminosidade, água, temperatura, umidade e ventilação, determinam o sucesso de seu crescimento (PLOUGHMAN,2007).

2.1 Temperatura

A temperatura influencia diretamente em vários processos biológicos dos vegetais, assim como a brusca variação desse parâmetro pode acarretar vários problemas nos processos de transpiração, respiração, fotossíntese, germinação, crescimento, floração e frutificação (CERMEÑO, 2005). Cada planta necessita de uma faixa de temperatura específica para que seu ciclo de desenvolvimento não seja alterado. Fora dessas faixas de valores os resultados podem acabar resultando até a morte da planta.

2.2 Umidade

A umidade relativa do ar representa a quantidade de água em forma de vapor presente na atmosfera em relação à quantidade máxima que poderia existir nas mesmas condições de temperatura (CGE,2011). Altos valores de umidade relativa do ar dificultam o processo de evaporação, já valores muito baixos interferem na etapa de fotossíntese da planta. Parâmetros como ventilação, nebulização e temperatura podem auxiliar no controle da umidade relativa do ar (SOLER e PALAU, 2012).

2.3 Ventilação

A ventilação é a troca de ar do meio externo com o meio interno da estufa, e tem como objetivo principal retirar o ar quente de dentro da estufa para controlar a temperatura, controlando também a presença de fungos e insetos e a umidade. Essa ventilação pode ser feita de duas formas. A primeira forma de ventilação se caracteriza por aberturas laterais ou superiores existentes na estufa, e a segunda é feita com um ventilador mecânico que faz a troca de ar da estufa com o ambiente externo (SOLER e PALAU, 2012).

2.4 Luminosidade

A luz é indispensável para a sobrevivência das plantas, pois é através dela que a planta realiza a fotossíntese, pela qual a planta transforma a molécula de água e de dióxido de carbono em energia química, liberando o oxigênio para purificação do ar (HOPKINS, 2006, p.4-8).

Embora a luz seja vital para o crescimento da planta, o seu excesso pode prejudicar ou até matar o vegetal.

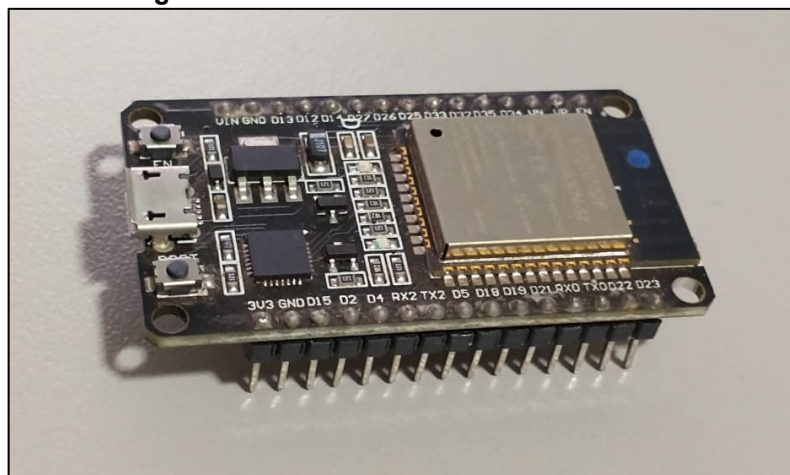
2.5 Irrigação

A água é essencial não só para o crescimento, como para a vida da planta. Existem várias formas de realizar a irrigação em uma estufa. As principais formas são: por gotejamento, por aspersão e por nebulização, que seria a melhor forma de fazer essa irrigação, pois proporciona menor dano as plantas, além de uma melhor distribuição de água (BEZERRA, 2003).

2.6 Placa eletrônica ESP32

ESP32 é uma plataforma de *hardware* livre (Figura 1), usado para a criação de dispositivos que permitam interação com o ambiente utilizando sensores como entrada. A plataforma utiliza-se de uma camada simples de *software* implementada na placa que utiliza a linguagem C/C++. Por ser uma plataforma de *hardware* livre existem diversos fabricantes desta placa e esses fabricantes utilizam nomes diferentes para suas placas, porém todas elas possuem a mesma finalidade (Vagapov, 2017).

Figura 1: Placa de desenvolvimento ESP32.



Fonte: Autoria própria (2020).

O ESP32 assim como seu antecessor o ESP8266 é um microcontrolador utilizado para projetos de automação. O ESP32 conta com 2 núcleos de 240 MHz dentro de um microprocessador de 32 bit`s, 448 kb de ROM para funções internas,

520 kb de SRAM que é utilizado para o código, e é constituído por dois módulos de conectividade um Bluetooth e o outro Wi-Fi (programar, 2017).

Comparando-o com o ESP8266, o ESP32 possui muito mais memória e processamento, além dos dois núcleos que permitem executar dois códigos em paralelo, coisa que raramente acontece nos microcontroladores comuns (programar, 2017).

A documentação do ESP32-WROOM-32 (ESPRESSIF, 2018c) informa que há no total 32 portas digitais. Cada porta opera em uma tensão de 0 V para nível lógico baixo ou 3,3 V para nível lógico alto.

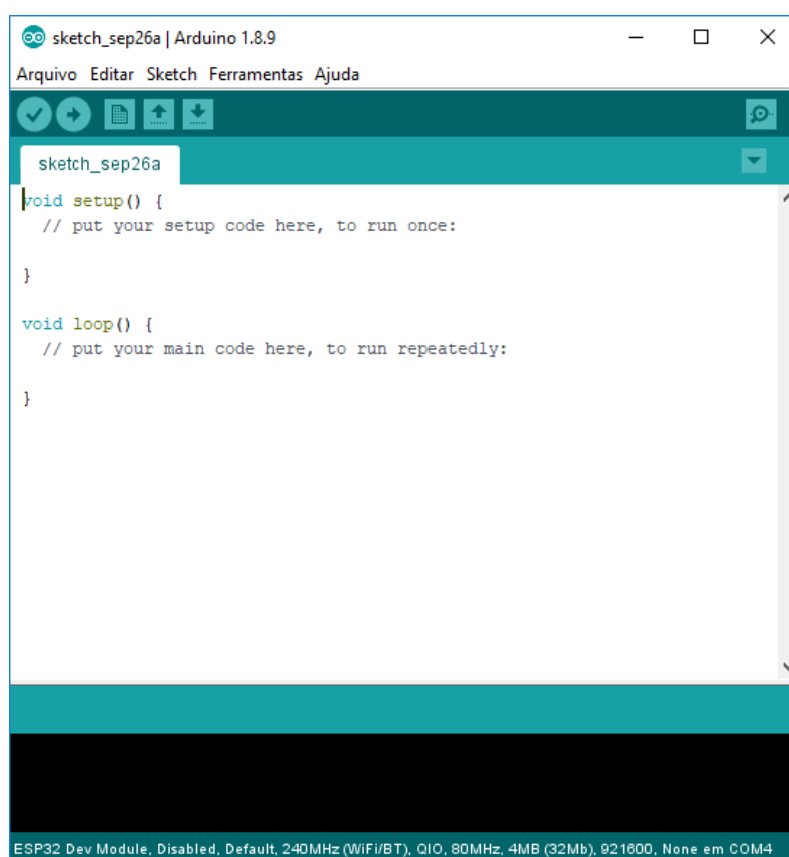
A placa possui 18 entradas analógicas para conversão digital, as quais fornecem uma resolução de 12 bits na escala de 0 a 3,3 V. Estas entradas serão utilizadas para a aquisição dos dados dos sensores que vem em um formato analógico, convertendo-os para uma escala digital entre 0 e 4095 (12 bits) para posteriormente serem manipulados no *software* (ESPRESSIF, 2018b).

Os pinos de tensão da placa de acordo com a documentação do ESP-WROOM-32 (ESPRESSIF, 2018c) são:

- 5 V: Tensão de saída com 5 V fornecida pela placa.
- 3,3 V: Tensão de saída com 3,3 V fornecida pela placa com corrente máxima de 80 mA.
- GND: Pino conhecido como terra.

2.7 IDE Arduino

Para programar e executar o código foi utilizado o IDE Arduino (Figura 2). Este *software* é um editor de texto, que verifica se a programação está correta e faz o upload do código para a placa (BUKMAN, 2016).

Figura 2: Arduino IDE.

Fonte: Autoria própria (2020).

A sintaxe da linguagem Arduino é basicamente C/C++, possui funções simples e específicas para trabalhar com as portas do Arduino. São necessárias duas funções elementares para seu funcionamento: `setup ()` e `loop ()` (ARDUINO).

2.8 Protocolo ESP-NOW

O ESP-NOW é um protocolo que permite que vários dispositivos se comuniquem entre si sem utilizar um agente externo para ocorrer essa comunicação. Ele é semelhante à conectividade sem fio utilizada em mouses e teclados, sendo necessário o emparelhamento entre os dispositivos para ocorrer a comunicação. Após o emparelhamento a conexão é segura e surpreendentemente rápida (MARQUES, 2020).

O ESP-NOW é um protocolo de comunicação M2M (Machine To Machine) ou seja, entre dois ESPs 32(mestre e escravo), porém os dispositivos podem se tornar mestres e escravos ao mesmo tempo, interagindo entre si (MARQUES, 2020).

Os ESPs32 conseguem se comunicar até uma distância de 165 metros, isso só com a antena interna do dispositivo (KOYANAGI, 2018).

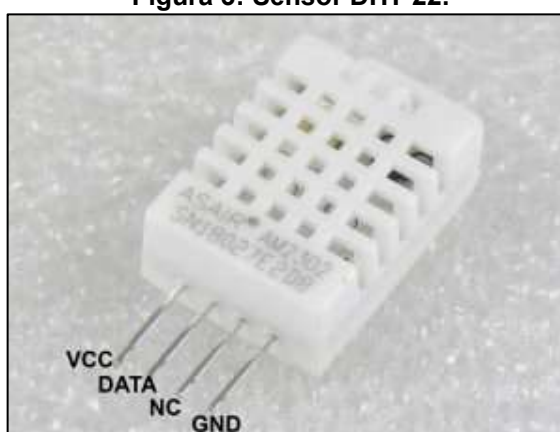
2.9 Sensor DHT 22

Para realizar as medições de umidade do ar e da temperatura, será utilizado o sensor DHT22 (Figura 3). Ele funciona com um sensor capacitivo de umidade e um termistor para medir o ar em volta dele, entregando suas leituras em forma de sinal digital para o pino de dados do ESP32 (JUNIOR, 2016).

As características dele são:

- Tensão de funcionamento de 3,3 a 5 V;
- Faixa de medição de umidade: 0-100 %;
- Opera com faixas de leitura de temperatura de -40 a 80 °C, com uma variação de $\pm 0,5$ %;
- Tamanho: 27mm x 59mm x 13.5mm;

Figura 3: Sensor DHT 22.



Fonte: Murta, 2019.

Considerando a pinagem vista na Figura 3, as configurações do sensor são feitas da seguinte forma:

1. VCC de 3,3 V;
2. Dados;
3. Não utilizado por não possuir nenhuma função;
4. GND.

2.10 Sensor LDR

Para efetuar as medições de luminosidade da estufa, utilizou-se um LDR (*Light Dependant Resistor*) similar ao mostrado na Figura 4, que é um resistor cuja resistência varia de acordo com a incidência de luz. Em ambientes escuros, sua resistência é elevada. Porém, conforme a luz incide sobre ele, sua resistência diminui (JUNIOR, 2016).

Figura 4: Sensor LDR.

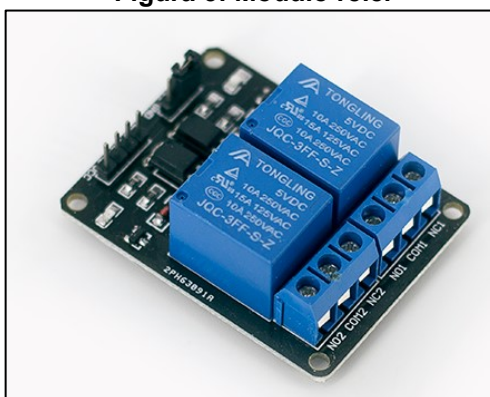


Fonte: Lopes, 2018.

O sensor LDR, pode ser utilizado de maneira satisfatória para indicar a presença ou ausência de luz.

2.11 Modulo relé

Relé é uma chave eletromecânica que liga ou desliga quando uma pequena tensão é aplicada a seus terminais de controle (Figura 5). Relés eletromecânicos foram utilizados por possuírem uma rápida velocidade de comutação e, ao receber o sinal do ESP32, são ativados imediatamente, possibilitando que a carga de 127 V necessária para o funcionamento dos aparelhos utilizados no projeto seja provida ao circuito (JUNIOR, 2016).

Figura 5: Modulo relé.

Fonte: Thomsen, 2013.

2.12 Sensor de vazão de água

O Sensor de fluxo de água é constituído por um corpo de plástico, um rotor, e um sensor de efeito Hall, conforme Figura 6. Quando a água flui através do rotor, ele irá girar e influenciará diretamente na frequência de pulsos, e através desses pulsos que são entregues pelo sensor de efeito hall é possível saber qual a vazão de água que passa por ele.

Figura 6: Sensor de vazão.

Fonte: Straub, 2016.

3. MATERIAIS E MÉTODOS

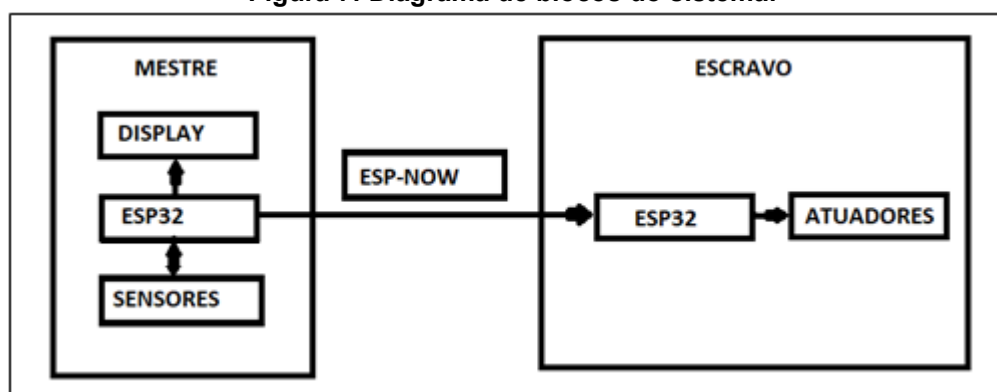
O projeto foi desenvolvido em etapas, cada uma feita separadamente uma da outra, sendo tudo integrado no fim do projeto.

O projeto foi planejado de acordo com as seguintes etapas:

- Desenvolvimento do *Hardware*;
- Desenvolvimento do *software*;
- Construção da estufa;
- Testes.

Abaixo tem-se o diagrama de blocos do sistema (Figura 7).

Figura 7: Diagrama de blocos do sistema.



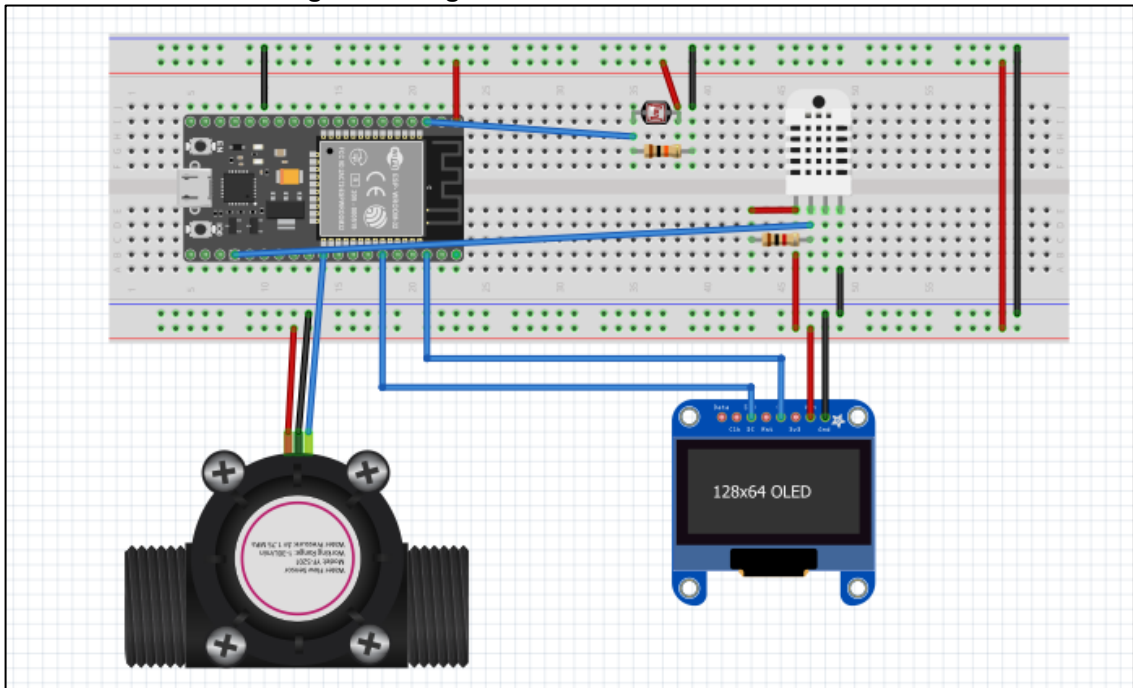
Fonte: Autoria própria (2020).

3.1. Desenvolvimento do *Hardware*

A escolha dos componentes e materiais baseou-se, principalmente, nos atributos como: alta qualidade, desempenho e durabilidade.

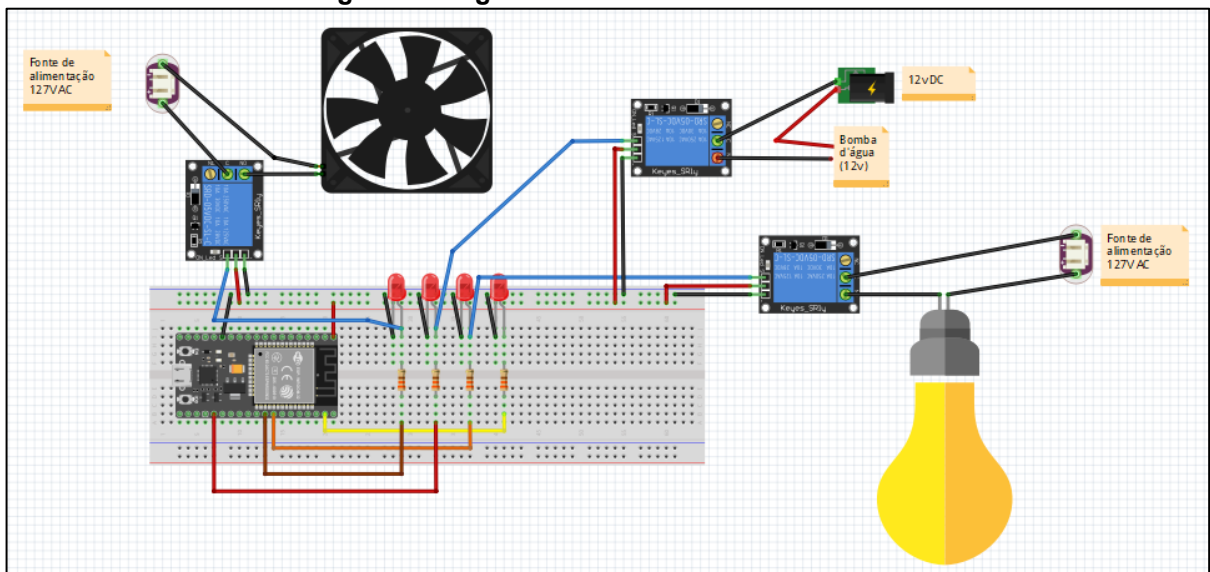
Já com todos os componentes, microcontrolador, sensores e atuadores foi desenvolvido o *Hardware* do projeto. Sendo feito em *protoboard* onde foi interligado todos os componentes do protótipo. O diagrama elétrico do protótipo pode ser visto nas Figuras 8 e 9.

Figura 8: Diagrama elétrico do ESP32 mestre.



Fonte: Autoria própria (2020).

Figura 9: Diagrama elétrico do ESP32 escravo.



Fonte: Autoria própria (2020).

No Quadro 1 são mostrados os componentes utilizados no projeto.

Quadro 1: Características dos componentes.

Componente	Características
ESP 32	Microprocessador dual core Tensilica Xtensa 32-bit LX6 Clock ajustável de 80 MHz até 240 MHz Tensão de operação: 3,3 VDC Possui SRAM de 512 kb Possui ROM de 448 kb Possui memória flash externa de 32 Mb Corrente máxima por pino é de 12 mA Possui 36 GPIOs GPIOs com função PWM / I2C e SPI
DHT 22	Tensão de operação: 3-5 VDC Faixa de medição de umidade: 0 a 100% UR Faixa de medição de temperatura: -40° a +80 °C Corrente: 2,5 mA max durante uso, em stand by de 100 uA a 150 uA Precisão de umidade de medição: ± 2,0% UR Precisão de medição de temperatura: ± 0,5 °C Tempo de resposta: 2 s
Sensor LDR	Tensão máxima: 150 VDC Potência máxima: 100 mW Resistência no escuro: 1 MΩ Resistência na luz: 10-20 KΩ
Modulo relé	Tensão de operação: 5 VDC Contato do relé permite tensão de até 250 VAC a 10 A Tempo de resposta: 5~10 ms
Sensor de vazão	Tensão de operação: 5-24V Corrente máxima: 15 mA Faixa de fluxo: 1-30 L/min Exatidão: ± 5 %

Fonte: Autoria própria (2020).

3.2. Desenvolvimento do *software*

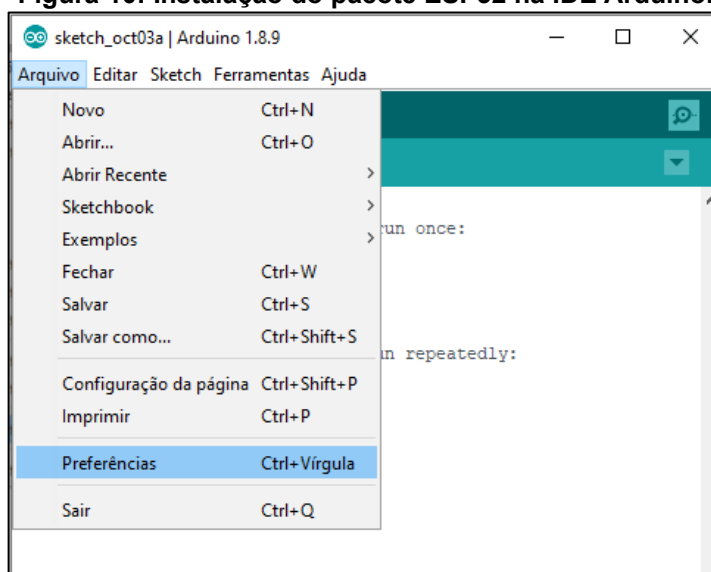
Para o desenvolvimento do *software* foi utilizado a IDE Arduino, por ter compatibilidade com o ESP32. Mas para ter essa compatibilidade foi preciso fazer alguns ajustes e instalações.

Já com a Arduino IDE instalada, fez-se os seguintes passos.

Com a IDE Arduino aberta, foi acessado a aba preferências (Figura 10).

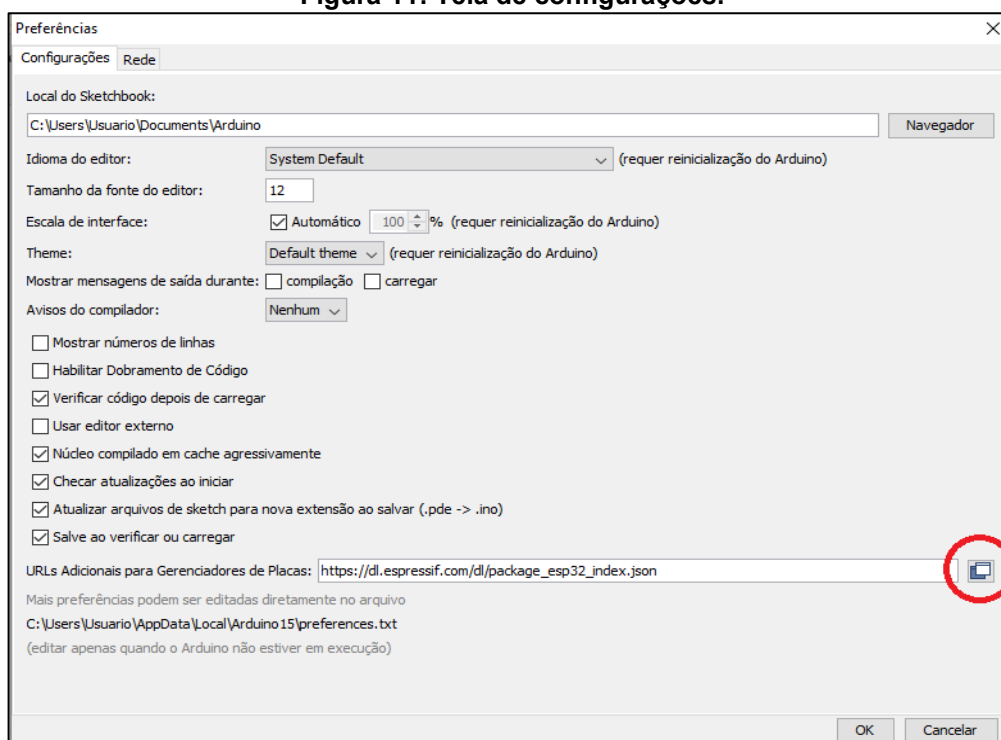
Dentro da aba preferências acessou-se o botão destacado da imagem (Figura 11), o qual abriu uma tela para digitar o link do pacote do ESP32 para a IDE Arduino.

Figura 10: Instalação do pacote ESP32 na IDE Arduino.



Fonte: Autoria própria (2020).

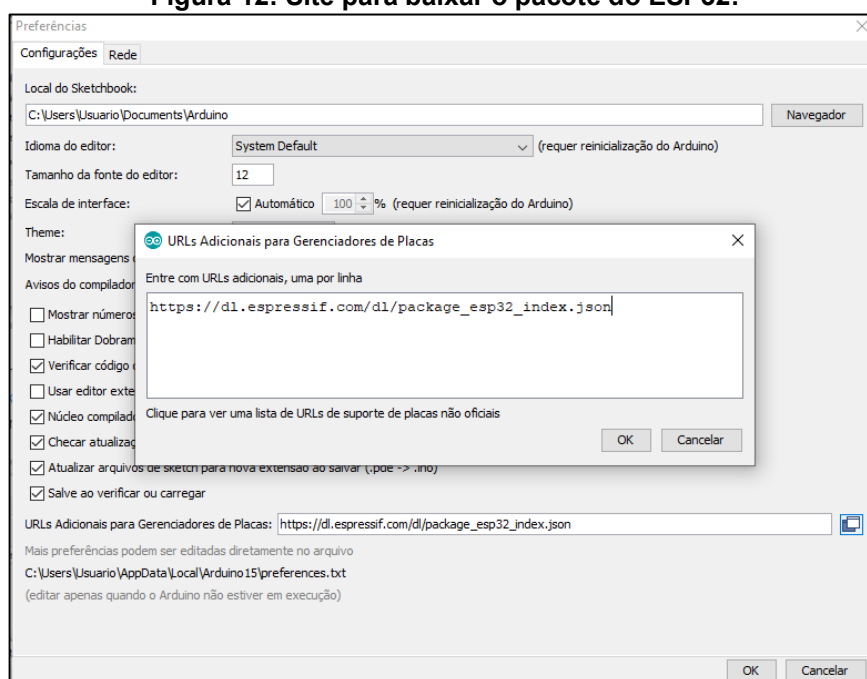
Figura 11: Tela de configurações.



Fonte: Autoria própria (2020).

Na Figura 12 pode-se ver o link do site para baixar o pacote ESP32. O site da Figura 12 é https://dl.espressif.com/dl/package_esp32_index.json. Após isso foi clicado em ok, então essa aba foi fechada.

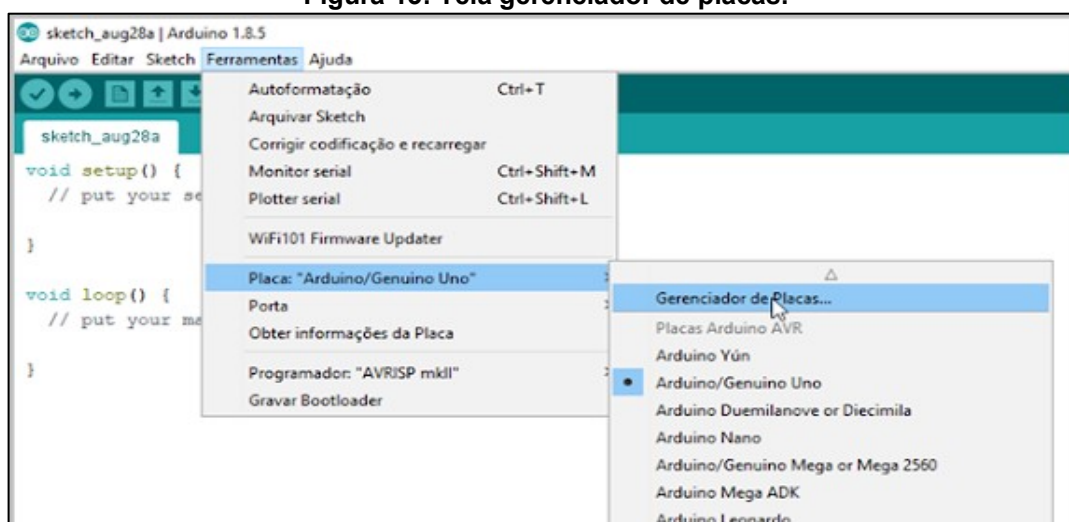
Figura 12: Site para baixar o pacote do ESP32.



Fonte: Autoria própria (2020).

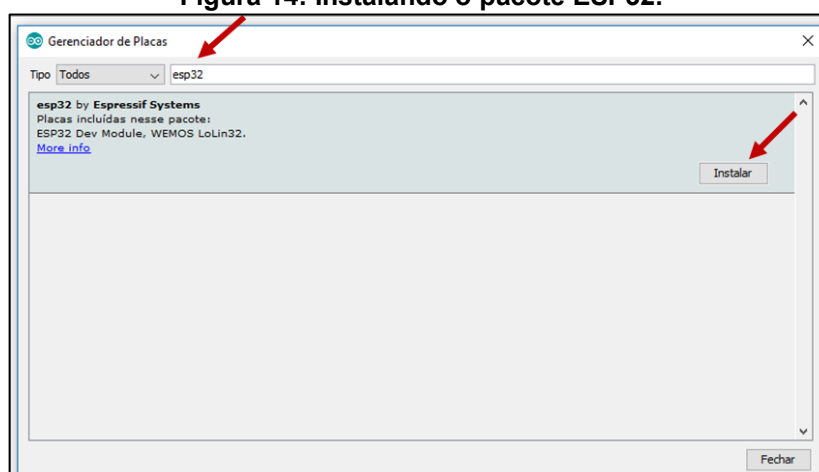
Depois na aba ferramentas, foi aberto o gerenciador de placas, conforme a Figura 13. Entrou-se no campo de busca e foi digitado esp32. Selecionou-se o resultado da pesquisa descrito como ESP32 *by Espressif Systems*, após isso foi clicado na opção instalar conforme a imagem (Figura 14).

Figura 13: Tela gerenciador de placas.



Fonte: Autoria própria (2020).

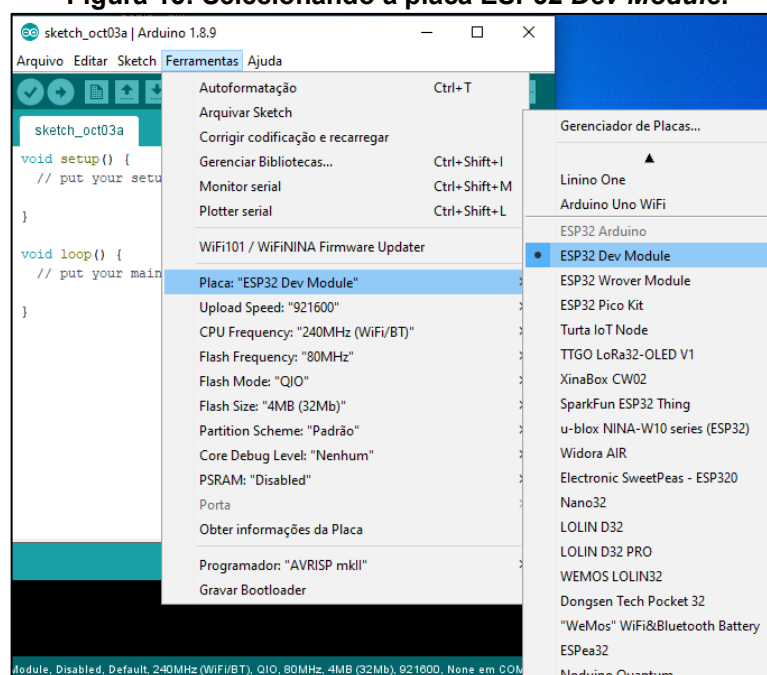
Figura 14: Instalando o pacote ESP32.



Fonte: Autoria própria (2020).

Após os procedimentos anteriores a placa ESP32 *Dev Module* foi listada pela Arduino IDE na opção placas instaladas (Figura 15).

Figura 15: Selecionando a placa ESP32 Dev Module.



Fonte: Autoria própria (2020).

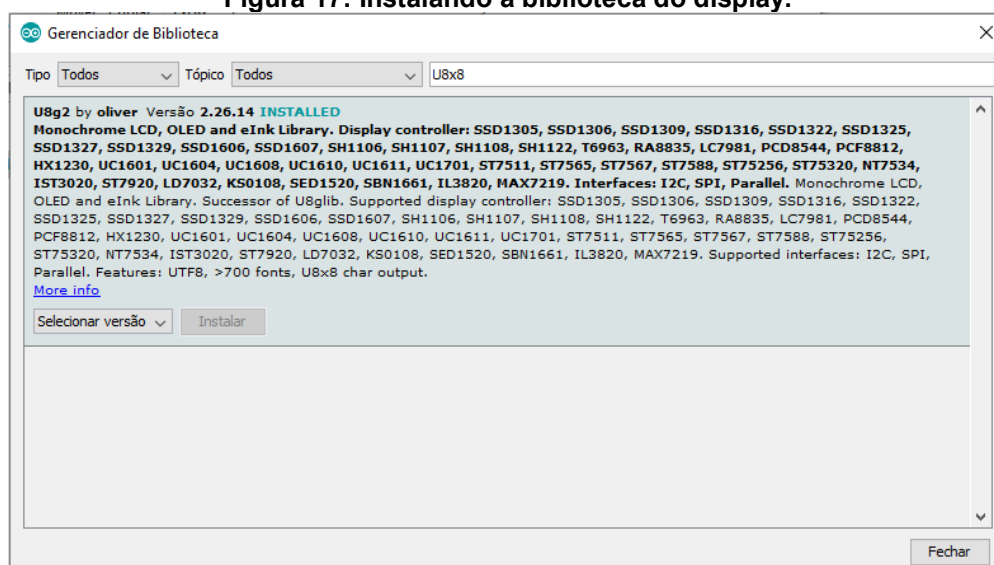
Depois foram instaladas as bibliotecas necessárias para cada sensor, nesse caso foi necessário instalar apenas a biblioteca para o DHT22 e para o display LCD, conforme as Figuras 16 e 17.

Figura 16: Instalando a biblioteca do sensor DHT22.



Fonte: Autoria própria (2020).

Figura 17: Instalando a biblioteca do display.

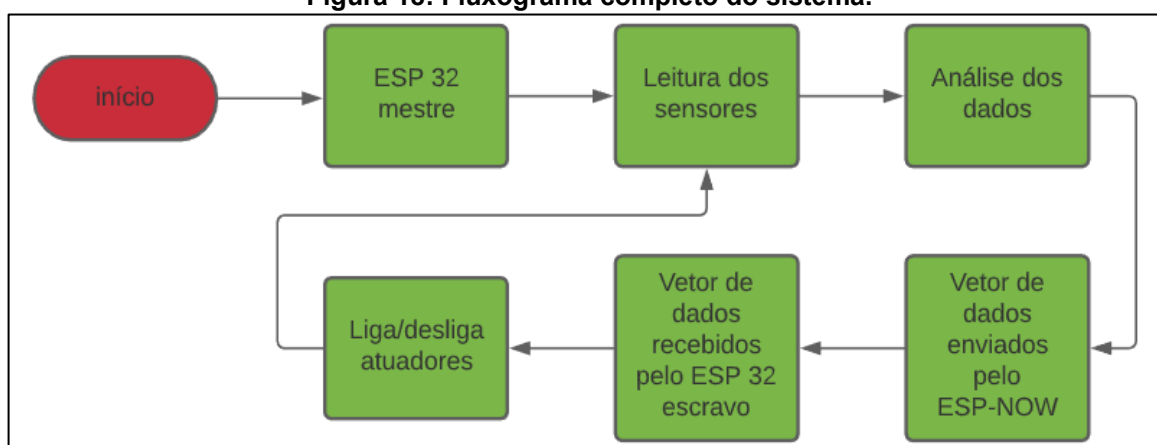


Fonte: Autoria própria (2020).

Já com as bibliotecas instaladas, pode-se começar a fazer a programação do ESP32. Como foi utilizado o protocolo ESP-NOW, precisou-se de dois ESPs32, um como mestre, que vai fazer a leitura dos sensores e apresentar no display, e o escravo, que através do protocolo ESP-NOW recebe os valores lidos pelos sensores, e liga ou desliga os atuadores.

Na Figura 18 observa-se o fluxograma completo do sistema.

Figura 18: Fluxograma completo do sistema.

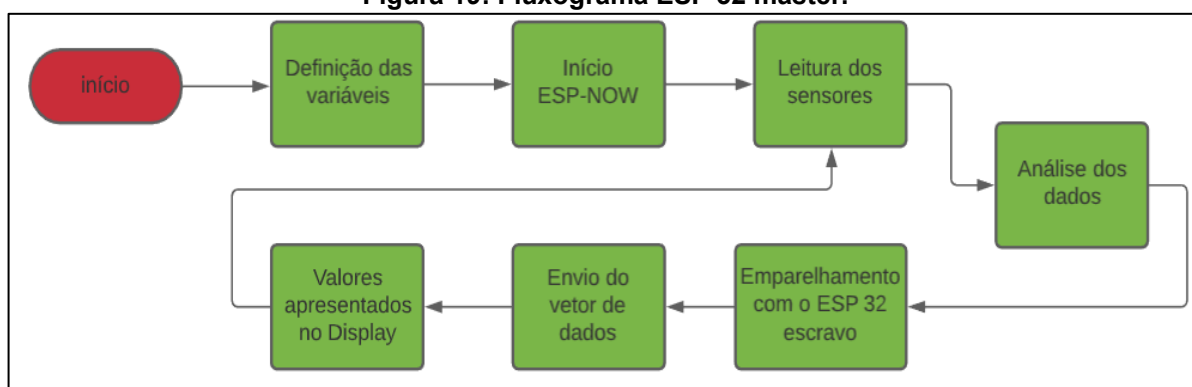


Fonte: Autoria própria (2020).

3.3. Código mestre

Na Figura 19 encontra-se o fluxograma do código do ESP 32 master.

Figura 19: Fluxograma ESP 32 master.



Fonte: Autoria própria (2020).

O *software* do projeto é constituído de dois códigos, o mestre e o escravo, que se comunicam através do protocolo ESP-NOW.

Primeiro define-se as bibliotecas que serão utilizadas, o canal para conexão dos ESP32, os pinos de leitura dos sensores e as variáveis que serão preenchidas com a leitura dos sensores. Chama a função que faz a interrupção do sensor de vazão, inicializa o display (Figura 20). E cria-se um vetor como os pinos que serão lidos e enviados para o escravo, no escravo deve-se ter o mesmo vetor com os mesmos pinos.

Figura 20: Código mestre, inclusão das bibliotecas e pinos.



```

master-pronto2 | Arduino 1.8.9
Arquivo Editar Sketch Ferramentas Ajuda

master-pronto2
#include <esp_now.h>
#include <WiFi.h>
#include <DHT.h>
#include <U8x8lib.h>

int sensorValue;
#define DHTPIN 15
#define DHTTYPE DHT22
#define CHANNEL 1
#define MIN_HUMIDITY 90
#define MAX_TEMP 29
#define MIN_TEMP 25
#define INTERVAL 1000
DHT dht(DHTPIN, DHTTYPE);
const int portaVazao = GPIO_NUM_5;
static void atualizaVazao();
volatile int pulsos_vazao = 0;
float vazao = 0;
float liters = 0;
float mililiters = 0;

void IRAM_ATTR gpio_isr_handler_up(void* arg)
{
    pulsos_vazao++;
    portYIELD_FROM_ISR();
}
U8X8_SSD1306_128X64_NONAME_SW_I2C display(SCL, SDA, U8X8_PIN_NONE);

uint8_t gpios[] = {2, 18, 19, 23};

```

Fonte: Autoria própria (2020).

Para o mestre conseguir enviar as informações para o escravo, é necessário ter o seu endereço (*mac address*), ou coloca-se {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF}, utilizando a segunda opção ele vai enviar as informações para todos os escravos que tiverem conectados, nesse caso, como tem apenas um ESP32 escravo, apenas ele vai receber as informações do mestre. No void setup inicializa o DHT 22, o display e o sensor de vazão (Figura 21).

Coloca-se o ESP32 em modo *station*, para poder usar o protocolo ESP-NOW, e chama-se o protocolo ESP-NOW.

É declarada a função que será responsável por informar o status do envio das informações do mestre para o escravo. Os pinos são colocados em modo de leitura, e chama-se a função Send.

Então é verificado o protocolo ESP-NOW, se foi iniciado corretamente, conforme a Figura 22. Depois são inicializados e configurados alguns parâmetros no display, como a fonte utilizada, e inicializa a função que faz a leitura da vazão de água.

Figura 21: Código mestre, função setup.

```

master-pronto2 | Arduino 1.8.9
Arquivo Editar Sketch Ferramentas Ajuda

master-pronto2 $

uint8_t macSlaves[6] = {{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF}};
float humidity = 0;
float temp = 0;

void setup() {
  Serial.begin(115200);
  dht.begin();
  setupDisplay();
  iniciaVazao((gpio_num_t) portaVazao);

  gpioCount = sizeof(gpios)/sizeof(uint8_t);

  WiFi.disconnect();
  WiFi.mode(WIFI_STA);

  InitESPNow();

  int slavesCount = sizeof(macSlaves)/6/sizeof(uint8_t);

  for(int i=0; i<slavesCount; i++){
    esp_now_peer_info_t slave;
    slave.channel = CHANNEL;
    slave.encrypt = 0;
    memcpy(slave.peer_addr, macSlaves[i], sizeof(macSlaves[i]));
    esp_now_add_peer(slave);
  }

  esp_now_register_send_cb(OnDataSent);

  for(int i=0; i<gpioCount; i++){
    pinMode(gpios[i], OUTPUT);
  }
}

```

Fonte: Autoria própria (2020).

Figura 22: Código mestre, função InitESPNow.

```

master-pronto2 | Arduino 1.8.9
Arquivo Editar Sketch Ferramentas Ajuda

master-pronto2 $

send();
}

void InitESPNow() {
  if (esp_now_init() == ESP_OK) {
    Serial.println("ESPNow Init Success");
  }
  else {
    Serial.println("ESPNow Init Failed");
    ESP.restart();
  }
}

void setupDisplay(){
  display.begin();
  display.setPowerSave(0); //modo powerSave (0-Off ? 1-On)
  display.setFont(u8x8_font_torussansbold8_u); //fonte utilizada
}

void iniciaVazao(gpio_num_t Port){
  gpio_set_direction(Port, GPIO_MODE_INPUT);
  gpio_set_intr_type(Port, GPIO_INTR_NEGEDGE);
  gpio_set_pull_mode(Port, GPIO_PULLUP_ONLY);
  gpio_intr_enable(Port);
  gpio_install_isr_service(0);
  gpio_isr_handler_add(Port, gpio_isr_handler_up, (void*) Port);
}

void send(){
  uint8_t values[gpioCount];
}

```

Fonte: Autoria própria (2020).

Tem-se a função Send, que faz as leituras dos pinos, e a variável que armazena os valores lidos pela função Send. Faz-se a leitura da temperatura e umidade pelo sensor DHT 22. E calcula a vazão de água (Figura 23). Faz-se a leitura do LDR, e compara-se o valor obtido para acionar a lâmpada ou o led.

Figura 23: Código mestre, leitura dos sensores.



```

master-pronto2 | Arduino 1.8.9
Arquivo Editar Sketch Ferramentas Ajuda
Carregar

master-pronto2 $

humidity = dht.readHumidity();
temp = dht.readTemperature();
    Serial.println(humidity);
    Serial.println(temp);

vazao = pulsos_vazao/5.5;
    mililiters = vazao / 60;    // Calculo de mililitros
    liters = liters + mililiters;
    pulsos_vazao = 0;
    Serial.println("Leitura do Sensor de Vazao:");
    Serial.println(liters);

sensorValue = analogRead(A0); // read analog input pin 0
Serial.print(sensorValue, DEC); // prints the value read
Serial.print(" \n"); // prints a space between the numbers
if(analogRead(A0) > 3300){ //SE LEITURA DO PINO FOR IGUAL A 1 (HIGH), FAZ
    digitalWrite(gpios[3], HIGH); //ACENDE O LED
}
else if(analogRead(A0) < 2800){ //SENÃO, FAZ
    digitalWrite(gpios[3], LOW); //DESLIGA O LED
}

if(analogRead(A0) < 1200){ //SE LEITURA DO PINO FOR IGUAL A 1 (HIGH), FAZ
    digitalWrite(gpios[2], HIGH); //ACENDE A LUZ
}
else if (analogRead(A0) > 1800){ //SENÃO, FAZ
    digitalWrite(gpios[2], LOW); //DESLIGA A LUZ
}

```

Fonte: Autoria própria (2020).

Depois na Figura 24 é comparado o valor lido da umidade para ligar ou deligar a bomba d'água. O mesmo é feito para a temperatura, compara-se para ver se será ligado ou não o sistema de ventilação

Na Figura 25 então lê-se o estado de cada pino e armazena-os em um vetor, que será enviado ao escravo, faz o emparelhamento dos ESP's mestre e escravo e faz o envio das informações.

A função OnDataSent verifica se o envio foi realizado com sucesso ou não.

Figura 24: Código mestre, comparação umidade.

```

master-pronto2 | Arduino 1.8.9
Arquivo Editar Sketch Ferramentas Ajuda
Carregar

master-pronto2 $
if(humidity < MIN_HUMIDITY)
{
  Serial.println("Umidade mínima atingida, enviando sinal para ligar o relê");
  digitalWrite(gpios[0], HIGH);
  if(liters >5){
    digitalWrite(gpios[0], LOW);
    liters = 0;
    Serial.println(liters);
  }
}
if(humidity > MIN_HUMIDITY)
{
  Serial.println("Umidade máxima atingida, enviando sinal para desligar o relê");
  if(liters >5){
    digitalWrite(gpios[0], LOW);
    liters = 0;
  }
}
if(temp > MAX_TEMP)
{
  Serial.println("TEMPERATURA máxima atingida, enviando sinal para ligar o relê");
  digitalWrite(gpios[1], HIGH);
}
else if(temp < MIN_TEMP)
{
  Serial.println("TEMPERATURA mínima atingida, enviando sinal para desligar o relê");
  digitalWrite(gpios[1], LOW);
}
}

```

Fonte: Autoria própria (2020).

Figura 25: Código mestre, envio dos dados.

```

master-pronto2 | Arduino 1.8.9
Arquivo Editar Sketch Ferramentas Ajuda
Carregar

master-pronto2 $

for(int i=0; i<gpioCount; i++){
  values[i] = digitalRead(gpios[i]);
  Serial.println(values[i]);
}

uint8_t broadcast[] = {0xFF, 0xFF,0xFF,0xFF,0xFF,0xFF};
esp_err_t result = esp_now_send(broadcast, (uint8_t*) values, sizeof(values));
Serial.print("Send Status: ");
if (result == ESP_OK) {
  Serial.println("Success");
}
else {
  Serial.println("Error");
}
}

void onDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
  char macStr[18];
  snprintf(macStr, sizeof(macStr), "%02x:%02x:%02x:%02x:%02x:%02x",
  mac_addr[0], mac_addr[1], mac_addr[2], mac_addr[3], mac_addr[4], mac_addr[5]);
  Serial.print("Sent to: ");
  Serial.println(macStr);
  Serial.print("Status: ");
  Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Success" : "Fail");
  send();
}

```

Fonte: Autoria própria (2020).

O loop apenas inicializa o display, e a função “showOnDisplay” é responsável por enviar as informações para o display (Figura 26).

Figura 26: Código mestre, função loop.

```

master-pronto2 | Arduino 1.8.9
Arquivo Editar Sketch Ferramentas Ajuda
master-pronto2 $

void loop() {
  showOnDisplay();
}

void showOnDisplay(){
  String strname = "EUCALIPTU'S";
  String strname1 = "GREENHOUSE";
  String strHumidity = "UMIDADE: " + String(humidity);
  String strTemp = "TEMP: " + String(temp);
  String strVazao = "VAZAO: " + String(liters);
  String strLdr = "LDR: " + String(sensorValue);
  display.clearLine(0);
  display.drawString(2, 0, strname.c_str());
  display.drawString(2, 1, strname1.c_str());
  display.drawString(0, 3, strHumidity.c_str());
  display.drawString(0, 4, strTemp.c_str());
  display.drawString(0, 5, strVazao.c_str());
  display.drawString(0, 6, strLdr.c_str());
}

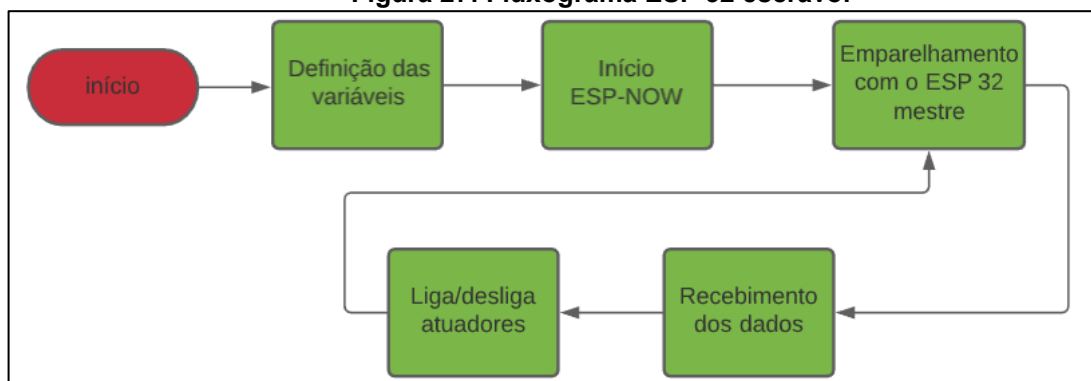
```

Fonte: Autoria própria (2020).

3.4 Código escravo

Na Figura 27 é demonstrado o funcionamento detalhado do código do ESP 32 escravo, apresentado em um fluxograma.

Figura 27: Fluxograma ESP 32 escravo.



Fonte: Autoria própria (2020).

O código do escravo é feito de forma semelhante ao mestre, primeiramente incluem-se as bibliotecas, declara-se os mesmos pinos que serão lidos no mestre, e os coloca em modo de saída (Figura 28).

Na Figura 29 inicia-se o protocolo ESP-NOW, verifica se os dados chegaram corretamente com a função OnDataRecv e troca os estados de saída dos pinos conforme foram recebidos do mestre.

Figura 28: Código escravo, inclusão das bibliotecas.



```

slave-pronto | Arduino 1.8.9
Arquivo Editar Sketch Ferramentas Ajuda
Verificar
slave-pronto $
#include <esp_now.h>
#include <WiFi.h>

uint8_t gpios[] = {2, 18, 19, 23};

int gpioCount;

void setup() {
  Serial.begin(115200);

  gpioCount = sizeof(gpios)/sizeof(uint8_t);

  WiFi.mode(WIFI_STA);
  InitESPNow();
  esp_now_register_recv_cb(OnDataRecv);

  for(int i=0; i<gpioCount; i++){
    pinMode(gpios[i], OUTPUT);
  }
}

```

Fonte: Autoria própria (2020).

Figura 29: Código escravo, função OnDataRecv.



```

slave-pronto | Arduino 1.8.9
Arquivo Editar Sketch Ferramentas Ajuda
Carregar
slave-pronto $
}

void InitESPNow() {
  if (esp_now_init() == ESP_OK) {
    Serial.println("ESPNow Init Success");
  }
  else {
    Serial.println("ESPNow Init Failed");
    ESP.restart();
  }
}

void OnDataRecv(const uint8_t *mac_addr, const uint8_t *data, int data_len) {
  char macStr[18];
  //Copiamos o Mac Address origem para uma string
  snprintf(macStr, sizeof(macStr), "%02x:%02x:%02x:%02x:%02x:%02x",
    mac_addr[0], mac_addr[1], mac_addr[2], mac_addr[3], mac_addr[4], mac_addr[5]);
  //Mostramos o Mac Address que foi a origem da mensagem
  Serial.print("Received from: ");
  Serial.println(macStr);
  Serial.println("");

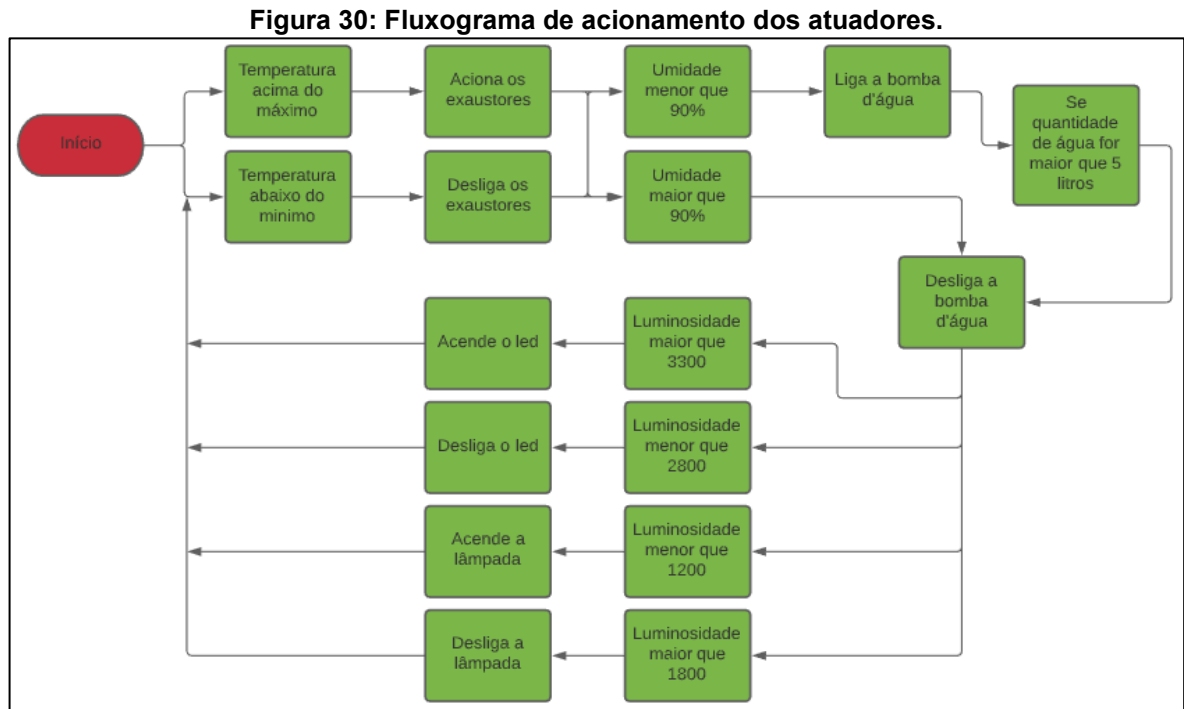
  for(int i=0; i<gpioCount; i++){
    digitalWrite(gpios[i], data[i]);
  }
}

void loop() {
}

```

Fonte: Autoria própria (2020).

Na Figura 30 tem-se o fluxograma do acionamento dos atuadores, bem como o acionamento da lâmpada ou do led.

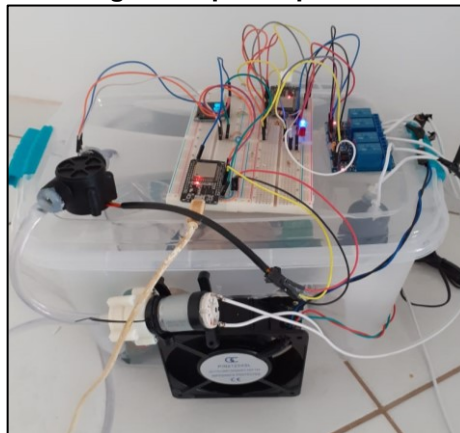


Fonte: Autoria própria (2020).

3.5 Construção da estufa

O protótipo da estufa foi feito em uma caixa plástica conforme a Figura a seguir, nela foram instalados todos os sensores e atuadores do sistema (Figura 31).

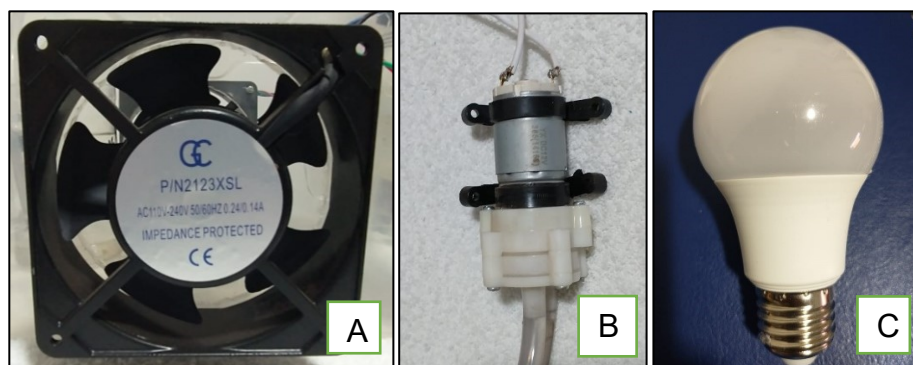
Figura 31: Imagem do protótipo da estufa pronta.



Fonte: Autoria própria (2020).

A Figura a 32 mostra os atuadores, bomba d'água e ventilador e a lâmpada que foram utilizados no projeto.

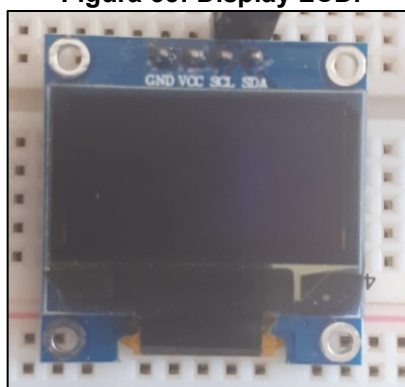
Figura 32: Atuadores utilizados no projeto. A) Ventilador; B) Bomba d'água; C) Lâmpada.



Fonte: Autoria própria (2020).

Display utilizado para apresentar localmente os valores lidos pelos sensores ao usuário (Figura 33). Serão apresentados no display os valores de temperatura, umidade e vazão de água, e também os valores lidos pelo LDR.

Figura 33: Display LCD.



Fonte: Autoria própria (2020).

3.6 Planejamento de testes

Os testes foram realizados para verificar possíveis erros no sistema, para isso os sensores foram testados em várias situações para garantir seu funcionamento. Os dados foram coletados a cada 0,5 segundos.

As condições para ligar e desligar os atuadores podem ser vistas a seguir:

- Luminosidade: a luminosidade lida pelo sensor varia de 0 a 4095, pois ele é um sensor analógico, com pouca luminosidade a leitura do sensor se aproxima mais de 0 e quando a luminosidade é alta a leitura se aproxima de 4095. A partir dessas informações foram escolhidos os valores para ligar e desligar os atuadores. Se a luminosidade na estufa

for menor que 1200, então a lâmpada dentro da mesma será acesa, e quando chegar a 1800 a lâmpada será apagada. Bem como se a luminosidade for maior que 3300 deverá acender um led, bem como ligar o relé, informando que a luminosidade incidente na estufa está muito alta, e quando essa leitura abaixar de 2800 o led vai apagar. Nos testes, o sensor foi deixado em uma sala iluminada por uma lâmpada, após algum tempo foi desligada a lâmpada da sala, fazendo a luminosidade cair, depois com uma lanterna, foi iluminado diretamente no sensor e por fim o sensor foi coberto retirando quase toda a incidência de luz. Os dados foram coletados por quase 4 minutos.

- Temperatura: a temperatura ideal para a espécie de eucalipto escolhida é entre 23 ° e 30 ° C, a partir desses valores foram escolhidos os valores para ligar e desligar o sistema de ventilação, se a temperatura se elevar e passar dos 29 °C o sistema de ventilação vai ser acionado, ficando ligado até a temperatura abaixar do mínimo especificado que é 25 °C, ao atingir 25 °C os ventiladores são desligados. Nos testes a variação da temperatura foi bem grande, primeiro o sensor foi deixado em temperatura ambiente, após algum tempo foi ligado uma fonte de calor para elevar sua temperatura, depois novamente sua temperatura voltou a temperatura ambiente. Os dados foram coletados por 6,5 minutos.
- Umidade: a umidade ideal para a espécie de eucalipto escolhida é entre 70 % e 100 %, então para não haver muita variação de umidade o valor escolhido para ligar o sistema de irrigação é quando a umidade baixar de 90 %, então quando a umidade relativa do ar dentro da estufa atingir uma marca menor que 90 %, o sistema de irrigação será acionado, quando o total de litros de água aspersada for igual a 5 litros, a bomba d'água é desligada porém se a umidade ainda for menos que 90 %, o sistema ligará novamente, e se a umidade estiver maior que 90 % o sistema permanecerá desligado. Para verificar as condições descritas acima o sistema foi ligado com a umidade do ar ambiente, e após algum tempo foi jogado vapor no sensor fazendo a umidade do ar subir, depois a umidade voltou a umidade do ar ambiente. Os dados foram coletados por aproximadamente 4,5 minutos

4. RESULTADOS

Nesse capítulo são apresentados os resultados das simulações realizadas, tais simulações foram realizadas com o intuito de se representar as variações de luminosidade, temperatura e umidade do ar presentes na estufa. Os gráficos apresentados foram feitos no software Excel.

4.1. Luminosidade

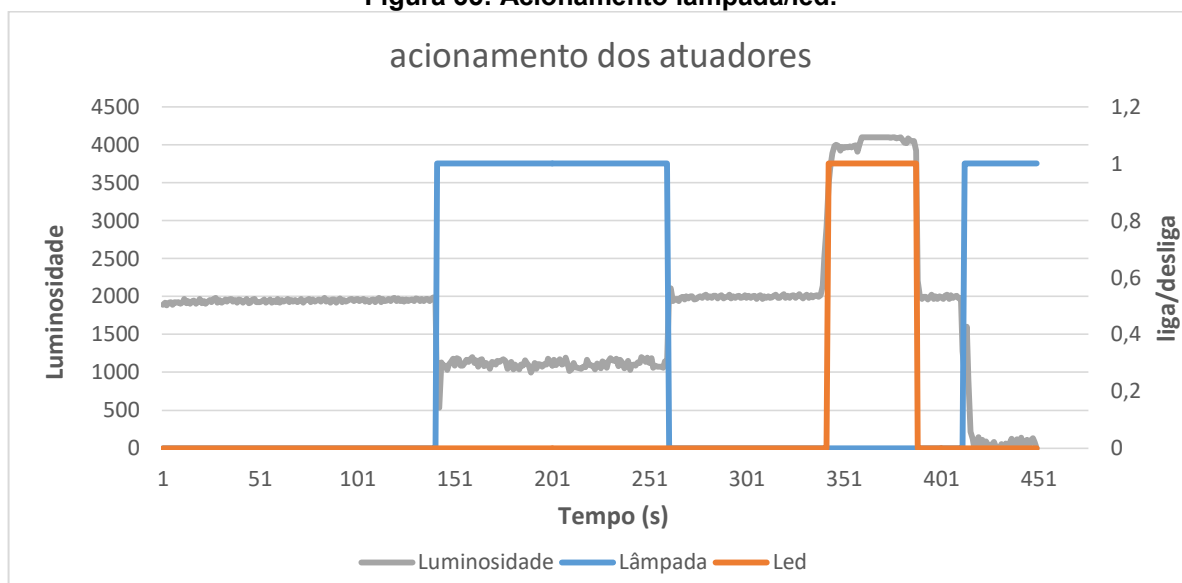
Como pode ser observado na Figura 34, primeiro o protótipo foi deixado em uma sala com uma lâmpada ligada, ficando assim até por volta de 140 segundos, quando a lâmpada foi desligada, nesse exato momento a lâmpada da estufa foi acionada ficando ligada até 260 segundos, quando novamente a lâmpada da sala foi ligada. Depois no instante 340 segundos do gráfico foi ligada uma lanterna diretamente sobre o LDR, neste momento foi acionado o led indicador de luminosidade excessiva, ficando acionado até 390 segundos. Por volta de 415 segundos o sensor LDR foi tapado completamente, novamente ligando o sistema de iluminação da estufa. Na Figura 35 pode-se observar o momento em que os atuadores foram acionados.

Figura 34: Gráfico do sensor LDR.



Fonte: Autoria própria (2020).

Figura 35: Acionamento lâmpada/led.

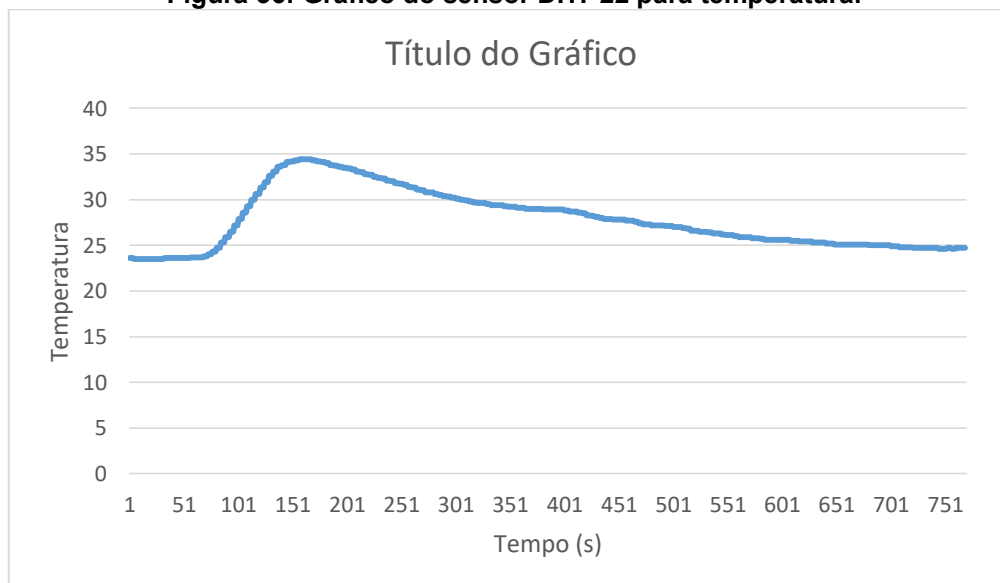


Fonte: Autoria própria (2020).

4.2. Temperatura

Na Figura 36 é possível visualizar a resposta do sensor DHT 22 para a temperatura em função do tempo, primeiro o sensor começou em temperatura ambiente, depois por volta de 70 segundos do gráfico foi inserido uma fonte de calor fazendo a temperatura se elevar chegando ao pico de quase 35 °C. Porém no ponto 109 quando a temperatura passou dos 29 °C foram acionados os ventiladores, ficando acionados até a temperatura abaixar de 25 °C, que ocorreu em 701 segundo.

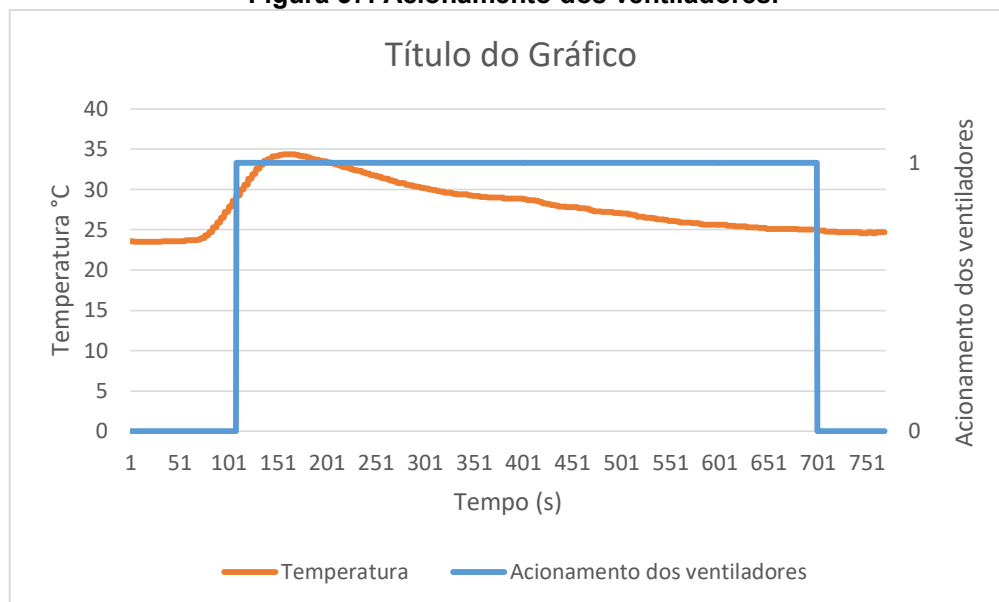
Figura 36: Gráfico do sensor DHT 22 para temperatura.



Fonte: Autoria própria (2020).

Na Figura 37 é possível observar o momento em que os ventiladores são acionados (29 °C) e o momento em que são desligados (25 °C).

Figura 37: Acionamento dos ventiladores.



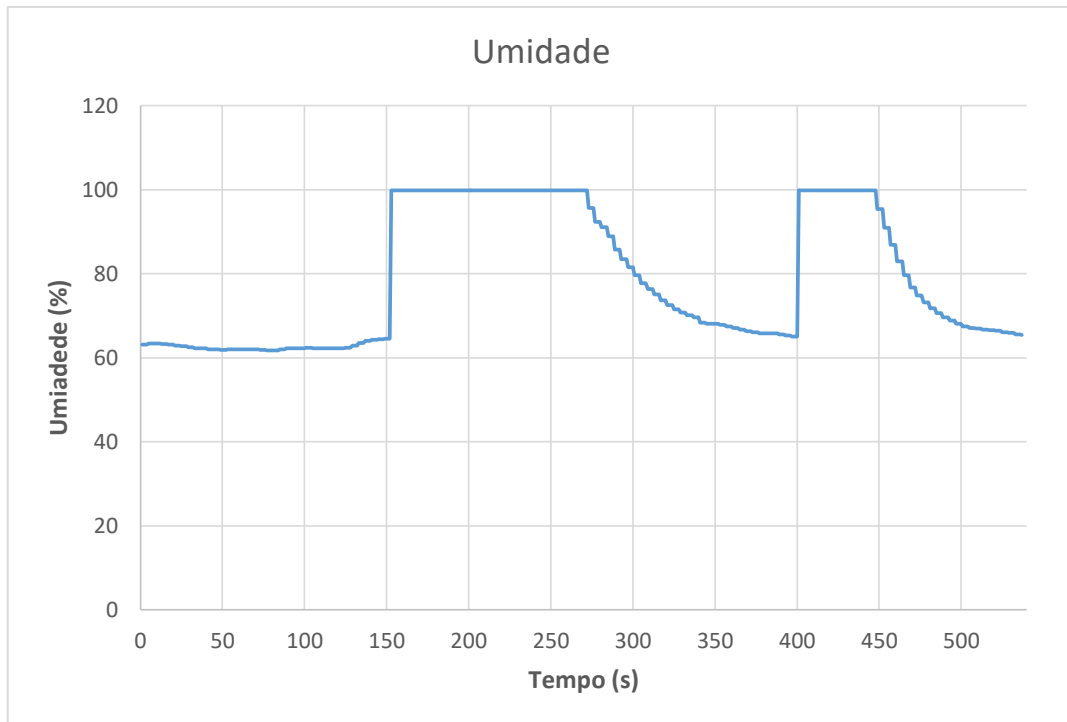
Fonte: Autoria própria (2020).

4.3. Umidade

Se a umidade dentro da estufa for inferior a 90%, o sistema de irrigação vai ser acionado, ficando ligado até ser aspersado 5 litros de água, como é possível ver na Figura 39. Depois de 150 segundos é possível observar que a umidade foi para 99,9% (Figura 38), porém o sistema foi desligar apenas por volta de 190 segundos (Figura 39), que foi o ponto em que o sensor de vazão chegou a 5 litros de água aspersada. O sistema foi ligar novamente apenas quando a umidade voltou a baixar de 90% no ponto de 285 segundos (Figura 39), depois o processo se repete da mesma forma.

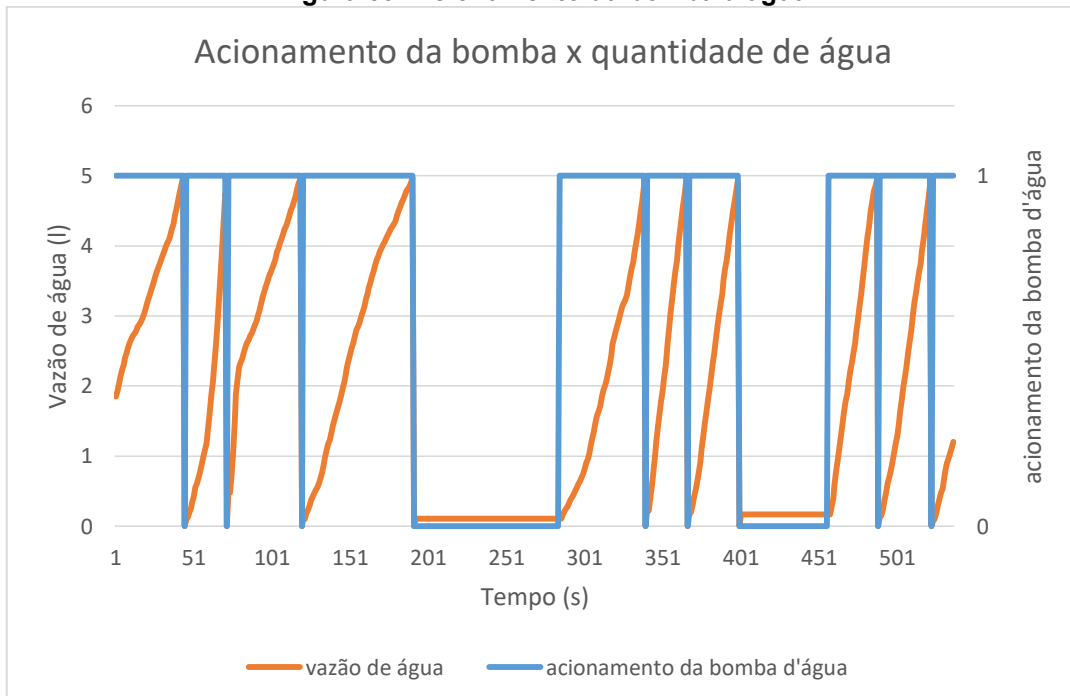
Na Figura 40, pode-se observar o gráfico do sensor de vazão em relação ao gráfico da umidade relativa do ar, também é possível observar uma imprecisão do sensor de vazão, pois nem todas as leituras ocorreram com o mesmo instante de tempo. Na Figura 41 tem-se todos os gráficos do sistema de umidade em uma mesma figura, para facilitar a visualização do gráfico do sensor de vazão em função do acionamento da bomba d'água.

Figura 38: Gráfico do sensor DHT 22 para umidade.



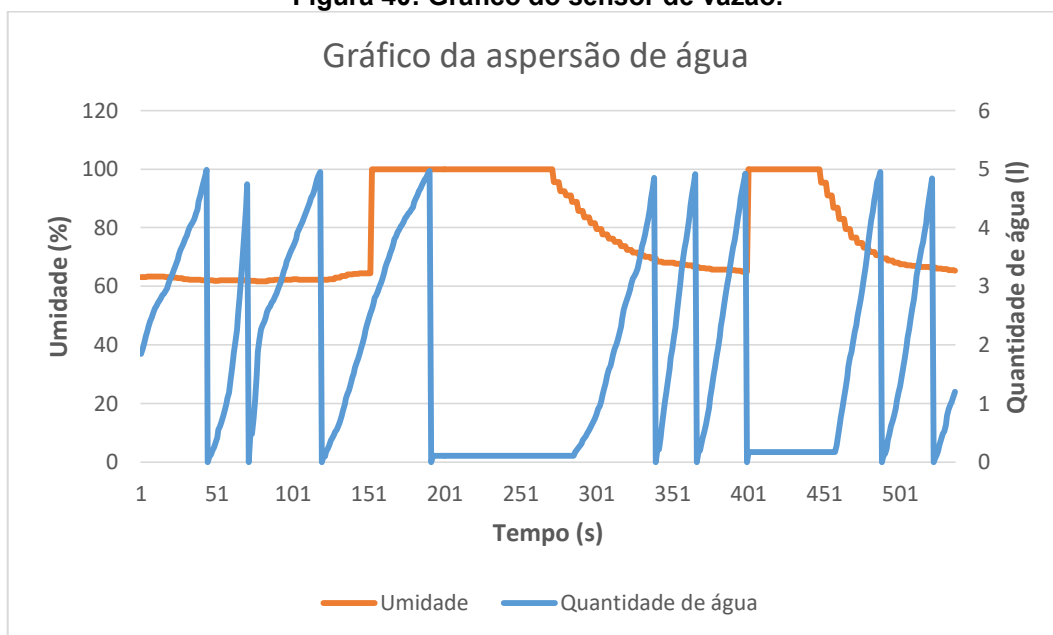
Fonte: Autoria própria (2020).

Figura 39: Acionamento da bomba d'água.



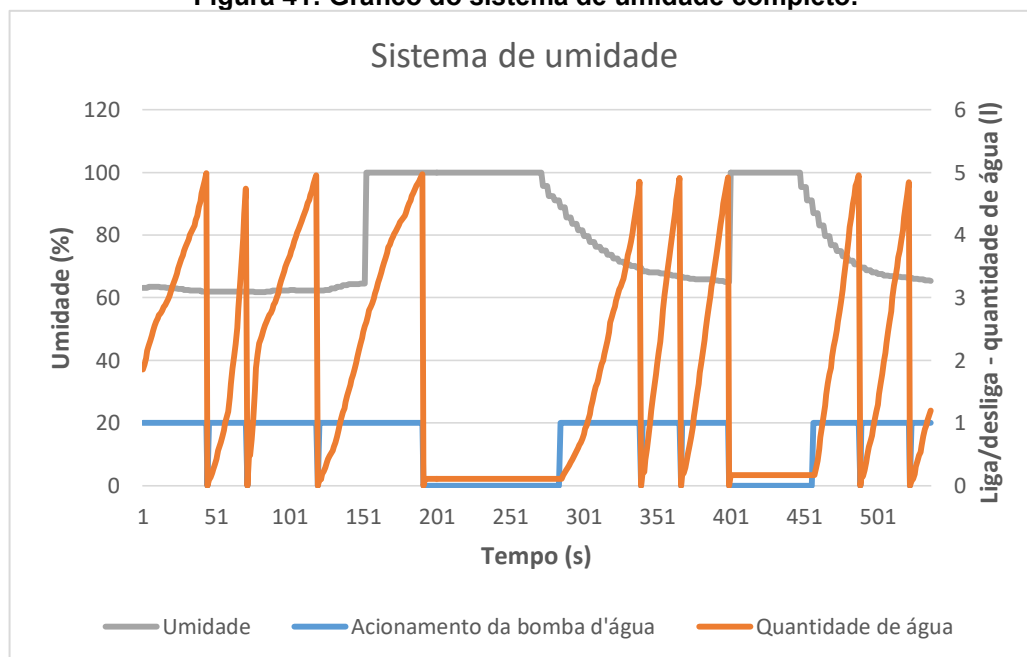
Fonte: Autoria própria (2020).

Figura 40: Gráfico do sensor de vazão.



Fonte: Autoria própria (2020).

Figura 41: Gráfico do sistema de umidade completo.



Fonte: Autoria própria (2020).

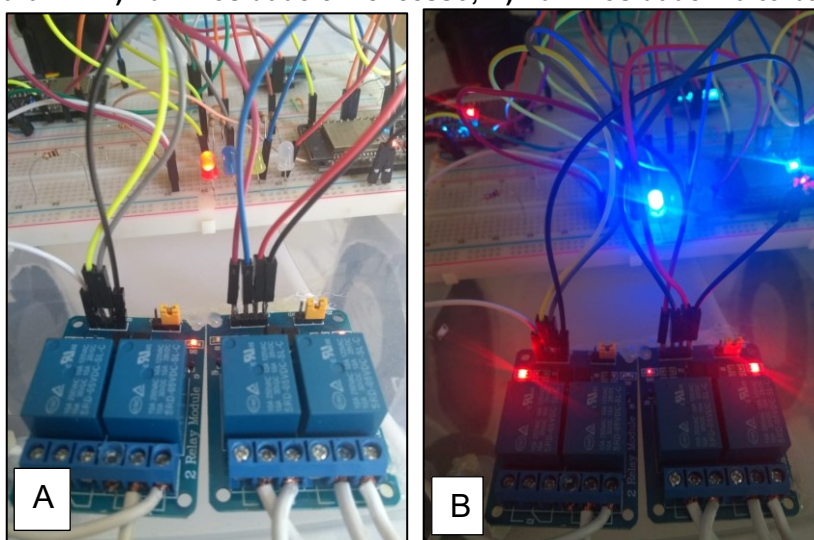
4.4. SIMULAÇÕES

Os resultados foram obtidos através das simulações, onde foi considerado as condições apresentadas anteriormente.

4.4.1. Luminosidade

A Figura 42-A mostra o momento em que o sensor LDR apresenta uma leitura maior que 3300, acendendo o led e ligando o relé indicando que a iluminação estava maior que o indicado, nesse caso a cobertura da estufa deveria ser coberta por um sombrite para não ocasionar a queima das mudas. Por outro lado, na Figura 42-B mostra quando o sensor foi coberto, retirando a luminosidade, nesse caso a lâmpada interna da estufa foi ligada.

Figura 42: A) Luminosidade em excesso; B) Luminosidade muito baixa.



Fonte: Autoria própria (2020).

4.4.2. Temperatura

Quando a temperatura passou dos 29 °C o sistema de ventilação foi ligado (Figura 43), ficando ligado até a temperatura baixar dos 25°C.

Figura 43: Led indicando o sistema de ventilação ligado.

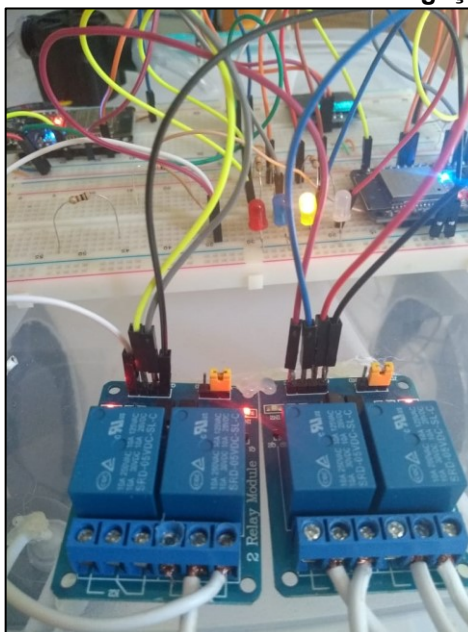


Fonte: Autoria própria (2020).

4.4.3. Umidade

Como dito anteriormente, quando a umidade do ar na estufa fica abaixo de 90%, o sistema de irrigação é acionado, como mostra a Figura 44.

Figura 44: Led indicando o sistema de irrigação ligado.



Fonte: Autoria própria (2020).

4.4.4. Display

Todos os dados lidos pelos sensores são apresentados no display LCD (Figura 45).

Figura 45: Dados apresentados no display LCD.



Fonte: Autoria própria (2020).

5. CONCLUSÃO

A automatização de estufas ainda é pouco difundida no país, devido ao alto custo da sua implementação. Portanto o objetivo do projeto foi fazer um protótipo simples e de bom desempenho.

Para isso foi indispensável a utilização de um microcontrolador confiável e com grande poder de processamento, como o ESP 32.

Ao analisar o presente projeto pode-se evidenciar que todos os objetivos propostos foram alcançados. Apenas o sensor de vazão apresentou variações na leitura da quantidade de água em diferentes instantes de tempo, que pode ocorrer em aspergir mais ou menos água do que deveria. Entretanto o resultando foi satisfatório, resultando em um protótipo que controla e supervisiona uma estufa de maneira adequada. Para o protótipo do projeto os sensores utilizados funcionaram bem, porém para um projeto maior, como em uma estufa em escala real, devem ser utilizados sensores mais robustos e confiáveis.

Para projetos futuros pode-se deixar a estufa mais interativa, deixando o usuário escolher a faixa de valores dos parâmetros, podendo a estufa ser usada para outras culturas, pode-se também apresentar os valores lidos dos sensores remotamente, como em uma página web ou aplicativo para celular.

REFERÊNCIAS

- ABRAF - Associação Brasileira De Produtores De Floresta Plantada. **Anuário estatístico da ABRAF 2010 ano base 2009**. Brasília: ABRAF 2010. ISSN 1980-8550. 140 p. Disponível em: <http://www.bibliotecaflorestal.ufv.br/bitstream/handle/123456789/3906/anuario-ABRAF-2010.pdf?sequence=1&isAllowed=y>. Acesso em: 15 mar. 2020.
- ABRAF- Associação Brasileira De Produtores De Floresta Plantada. **Anuário estatístico ABRAF 2013 ano base 2012**. São Paulo: ABRAF, 2013. 142 p. Disponível em: <http://www.bibliotecaflorestal.ufv.br/handle/123456789/3910>. Acesso em: 18 mar. 2020.
- ARDUINO**. Disponível em: <https://www.arduino.cc>. Acesso em: 25 fev. 2020.
- BEZERRA, F.C. **Produção de mudas de hortaliças em ambiente protegido**. Fortaleza: Embrapa, 2003. ISSN 1677-1915. 21 p. Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/CNPAT-2010/9014/1/Dc-072.pdf>. Acesso em: 03 mar.2020.
- BISON, O. **Melhoramento de eucalipto visando à obtenção de clones para a indústria de celulose**. 2004. Tese (Doutorado em Genética e melhoramento de plantas) - Universidade Federal de Lavras, Lavras, 2004. 182 p. Disponível em: http://repositorio.ufla.br/bitstream/1/3870/1/TESE_Melhoramento%20de%20eucalipto%20visando%20%C3%A0%20obten%C3%A7%C3%A3o%20de%20clones%20para%20a%20ind%C3%A9stria%20de%20celulose.pdf. Acesso em: 09 jun.2020.
- BUKMAN. G. **Desenvolvimento de um protótipo de baixo custo para segurança residencial**. 2016. Monografia (Graduação em Engenharia de controle e automação) - Universidade Federal de Ouro Preto, Ouro Preto, 2016. 47 p. Disponível em: https://www.monografias.ufop.br/bitstream/35400000/280/1/MONOGRAFIA_DesenvolvimentoDeProt%C3%B3tipo.pdf. Acesso em: 10 jun.2020.
- CERMEÑO, Z. S. **Construcción de invernaderos**. 3. ed. Madrid: Grupo MundiPrensa, 2005. Disponível em: https://books.google.com.br/books?id=Glip3Q7T9mEC&pg=PA117&hl=pt-BR&source=gbs_selected_pages&cad=2#v=onepage&q&f=false. Acesso em: 10 out. 2020.
- CGE - Centro de Gerenciamento de Emergências. **Umidade Relativa do Ar**, 2012. Cidade de São Paulo Infraestrutura urbanas e obras. Disponível em: <http://www.cgesp.org/v3/umidade-relativa-do-ar.jsp>. Acesso em: 01 dez. 2020.
- ESPRESSIF. **ESP32 Datasheet 2018b**. 64p. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Acesso em: 23 mar. 2020.

ESPRESSIF. **ESP-WROOM-32 Datasheet 2018c**. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datashet_en.pdf. Acesso em: 20 out. 2020.

ESPRESSIF. **Placas eletrônicas de desenvolvimento 2018a**. Disponível em: <https://www.espressif.com/en/products/hardware/development-boards>. Acesso em: 21 jun.2020.

FERNANDES, G. D.; PREUSS, E.; SILVA, T. L. **Sistema Automatizado de Controle de Estufas para Cultivo de Hortaliças**. Frederico Westphalen, 2017. 32 p. Disponível em: https://repositorio.ufsm.br/bitstream/handle/1/12958/TCCG_SIFW_2017_FERNANDES_DOUGLAS.pdf?sequence=1&isAllowed=y. Acesso em: 26 set. 2020.

FIGUEIREDO, G. **Casa da agricultura: produção em ambiente protegido**. Campinas, CATI, n. 2, jun. 2011. ISSN 0100- 6541. 22p. Disponível em: <http://docplayer.com.br/434877-Producao-em-ambiente-protgido.html>. Acesso em: 10 out. 2020.

GLÓRIA, L. P.; NAKA, R. D.; BELONI, G. *et al.* **Estufa climatizada: ter sua plantação sem preocupação**. São Caetano do Sul, 2015. 34 p. Disponível em: <https://livrozilla.com/doc/1437937/estufa-climatizada>. Acesso em: 20 out. 2020.

HIGASHI, E.N., FAGUNDES, N.B., SILVEIRA, R.L.V.A., *et al.* **Influência da temperatura e da umidade relativa no enraizamento de clones de eucalipto na região sul da Bahia**. 13 f. Disponível em: http://www.celsofoelkel.com.br/artigos/Poster_38.pdf. Acesso em: 15 mar. 2020.

HOPKINS, W. G. **Photosynthesis and respiration**. Infobase Publishing, 2006. 177 p. Disponível em: <https://www.scribd.com/document/298456255/Photosynthesis-and-Respiration-Hopkins-W-G-2006>. Acesso em: 17 mar. 2020.

JUNIOR, A. L. A. **Sistema de monitoramento e climatização de estufa de pequeno porte em um contexto doméstico**. 2016. 84 p. Trabalho de Conclusão de Curso (Engenharia em computação) - Centro Universitário de Brasília, Brasília, 2016. Disponível em: <https://repositorio.uniceub.br/jspui/bitstream/235/8677/1/20766579.pdf>. Acesso em: 29 abr.2020.

KOYANAGI, F. **ESP32 com protocolo ESP-NOW**. Março, 2018. Disponível em: <https://www.fernandok.com/2018/03/esp32-com-protocolo-esp-now.html>. Acesso em: 26 jun. 2020.

LITJENS, O. J. **Automação de estufas agrícolas utilizando sensoriamento remoto e o protocolo zigbee**. 2009. 72 p. Monografia (Graduação em Engenharia Elétrica com ênfase em eletrônica) - Universidade de São Paulo, São Carlos, 2009. Disponível em: <http://docplayer.com.br/49604155-Automacao-de-estufas-agricolas-utilizando-sensoriamento-remoto-e-o-protocolo-zigbee.html>. Acesso em: 26 jul. 2020.

LOPES, H. P. **Desenvolvimento de hardware para bancada remota de quatro tanques**. 2018. 72 p. Monografia (Graduação em Engenharia de controle e automação) - Universidade de Ouro Preto, Ouro Preto, 2018. Disponível em: https://monografias.ufop.br/bitstream/35400000/1349/1/MONOGRAFIA_DesenvolvimentoHardwareBancada.pdf. Acesso em: 08 nov. 2020.

MARANGONI, V, H; MOREIRA, H, R; SOUZA, P, S. **Automação de estufas agrícolas utilizando sensores e arduino**. 6º Jornada Científica e Tecnológica e 3º Simpósio de Pós-Graduação do IFSULDEMINAS, Pouso Alegre, 2014. Disponível em: <https://jornada.ifsuldeminas.edu.br/index.php/jcpoa/jcpoa/paper/viewFile/580/325>. Acesso em: 01 fev. 2020.

MARQUES, J. **O que é ESP-NOW – e como funciona? – Código exemplo explicado**. Janeiro 2020. Disponível em: <http://www.fvml.com.br/2020/01/o-que-e-esp-now-e-como-funciona-codigo.html>. Acesso em: 24 jun. 2020.

Murta, G. **Sensores DHT11 e DHT22: guia básico dos sensores de umidade e temperatura**. 2019. Disponível em: <https://blog.eletrogate.com/sensores-dht11-dht22/>. Acesso em: 10 nov. 2020.

NAHUZ, M. A. R., POFFO, O., BRAINER, M. S. C. P., et al. **Setores consumidores de madeira: aspectos do mercado atual e potencial do eucalipto**. 2012. Página 151. ISBN: 9788577911639.

PIPER. C. **A densidade ideal para a tela de sombrite cobrir uma estufa**. EHow Brasil. 2017. Disponível em: http://www.ehow.com.br/densidade-ideal-tela-sombrite-cobrir-estufa-fatos_39384/. Acesso em: 24 out. 2020.

PLOUGHMAN, T. **Understanding orchids**, 2007. 64 p. Disponível em: <http://www.the-wow-collection.com/software/orchids.pdf>. Acesso em: 23 maio 2018. Programar, revista portuguesa de programação, edição 58, outubro de 2017.

REIFSCHNEIDER, F. J. B., HENZ, G. P., RAGASSI, C. F., et al. **Novos ângulos da história da agricultura no Brasil**. Brasília, DF: Embrapa Informação Tecnológica, 2010, 112 p. Disponível em: <https://www.embrapa.br/busca-de-publicacoes/-/publicacao/868764/novos-angulos-da-historia-da-agricultura-no-brasil>. Acesso em: 26 nov. 2019.

REIS, N, V, B. **Construção de estufas para a produção de hortaliças, nas regiões norte, nordeste e centro-oeste**. Brasília: Embrapa hortaliças, 2005. 16p. ISSN 1415-3033. Disponível em: <https://www.embrapa.br/documents/1355126/9124396/Constru%C3%A7%C3%A3o+de+estufas.pdf/8bec74eb-2206-44ff-9aad-538141520c4a>. Acesso em: 28 jul. 2020.

SANTOS, A. B.; BARRETO, R. G. **Projeto e desenvolvimento de uma estufa automatizada para plantas**. 2012. 74 p. Trabalho de Conclusão de Curso (Bacharel em Engenharia industrial elétrica) - Universidade Tecnológica Federal do Paraná, Curitiba, 2012. Disponível em:

http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/1264/1/CT_ENGELN_2012_1_01.pdf. Acesso em: 28 out. 2020.

SANTOS, A.; XAVIES, A.; OLIVEIRA, M. L. *et al.* **Efeito da estaquia, miniestaquia, microestaquia e micropropagação no desempenho silvicultural de clones de *Eucalyptus grandis***. 2005. 10p. Disponível em: <http://www.ipef.br/PUBLICACOES/SCIENTIA/nr68/cap03.pdf>. Acesso em: 25 jan. 2020.

SEBRAE. **Eucaliptos clonados: uma alternativa sustentável e que gera renda**. Alagoas, SEBRAE, 2013. Disponível em: <http://www.al.agenciasebrae.com.br/sites/asn/uf/AL/eucaliptos-clonados-uma-alternativa-sustentavel-e-que-gera-renda,a8a23f316eb06410VgnVCM1000003b74010aRCRD>. Acesso em: 05 fev. 2020.

SILVA, A. P. G.; SALVADOR, M. **O que são sistemas supervisórios?** 2005. Disponível em: http://www.academia.edu/10054113/O_que_s%C3%A3o_sistemas_supervis%C3%B3rios. Acesso em: 29 jul. 2020.

SILVA, L.C.S. **Produção de mudas de *Eucalyptus urograndis* em substratos obtidos a partir de resíduos agroindustriais**. 2017. 49 f. Dissertação (Mestrado em Produção vegetal sustentável do Cerrado) - Instituto Federal de Educação, Ciência e Tecnologia Gioiano, Campus Rio Verde, Rio Verde, 2017. Disponível em: https://sistemas.ifgoiano.edu.br/sgcursos/uploads/anexos_5/2018-01-08-10-21-42Laisa%20Cristine%20dos%20Santos%20Silva.pdf. Acesso em: 15 abr. 2020.

SILVA, L. H. **Sensor de umidade e temperatura DHT22**. Disponível em: <https://arduinoeasy.wordpress.com/aplicacoes/sensor-de-umidade-e-temperatura-dht22/>. Acesso em: 01 dez. 2020.

SILVA, R. R. **Estufa automatizada para o cultivo de plantas – componentes atuadores do sistema**. In: CONGRESSO NACIONAL DE INICIAÇÃO CIENTÍFICA, 13, 2013, Jaguariúna. Anais do Conic-Semesp, v.1. Jaguariúna: Faculdade Anhanguera de Campinas, 2013. ISSN 2357-8904. Disponível em: <http://conic-semesp.org.br/anais/files/2013/trabalho-1000015762.pdf>. Acesso em: 28 abr. 2020.

SOUZA, G. T. **Controle e Automação Industrial**, Escola Técnica Estadual Pedro Ferreira Alves, julho 2004. Disponível em: <http://www.ebah.com.br/content/ABAAABgVQAD/automacao-industrial>. Acesso em: 26 abr. 2020.

STRAUB, M.G. **Sensor de refluxo de água para Arduino 1-30 l/min**, 2016. Disponível em: <https://www.usinainfo.com.br/blog/sensor-de-fluxo-de-agua-para-arduino-1-30-lmin/>. Acesso em 10 nov. 2020.

THOMSEN, A. **Controlando lâmpadas com Módulo Relé Arduino**. 2013. Disponível em: <https://www.filipeflop.com/blog/controla-modulo-rele-arduino/>. Acesso em: 06 nov. 2020.

WENDLING, M. **Sensores**. Guaratinguetá, 2010. Disponível em: <http://www2.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4---sensores-v2.0.pdf>. Acesso em: 28 mar. 2020.

APÊNDICE I - CÓDIGO MESTRE

```

#include <esp_now.h>
#include <WiFi.h>
#include <DHT.h>
#include <U8x8lib.h>

int sensorValue;
#define DHTPIN 15
#define DHTTYPE DHT22
#define CHANNEL 1
#define MIN_HUMIDITY 90
#define MAX_TEMP 29
#define MIN_TEMP 25
#define INTERVAL 1000
DHT dht(DHTPIN, DHTTYPE);
const int portaVazao = GPIO_NUM_5;
static void atualizaVazao();
volatile int pulsos_vazao = 0;
float vazao = 0;
float liters = 0;
float mililiters = 0;

void IRAM_ATTR gpio_isr_handler_up(void* arg)
{
  pulsos_vazao++;
  portYIELD_FROM_ISR();
}
U8X8_SSD1306_128X64_NONAME_SW_I2C          display(SCL,          SDA,
U8X8_PIN_NONE);

uint8_t gpios[] = {2, 18, 19, 23};

int gpioCount;

uint8_t macSlaves[][6] = {{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF}};
float humidity = 0;
float temp = 0;

void setup() {
  Serial.begin(115200);
  dht.begin();
  setupDisplay();
  iniciaVazao((gpio_num_t) portaVazao);

  gpioCount = sizeof(gpios)/sizeof(uint8_t);

  WiFi.disconnect();
  WiFi.mode(WIFI_STA);

  InitESPNow();

```



```

int slavesCount = sizeof(macSlaves)/6/sizeof(uint8_t);

for(int i=0; i<slavesCount; i++){
  esp_now_peer_info_t slave;
  slave.channel = CHANNEL;
  slave.encrypt = 0;
  memcpy(slave.peer_addr, macSlaves[i], sizeof(macSlaves[i]));
  esp_now_add_peer(&slave);
}

esp_now_register_send_cb(OnDataSent);

for(int i=0; i<gpioCount; i++){
  pinMode(gpios[i], OUTPUT);
}

send();
}

void InitESPNow() {
  if (esp_now_init() == ESP_OK) {
    Serial.println("ESPNow Init Success");
  }
  else {
    Serial.println("ESPNow Init Failed");
    ESP.restart();
  }
}

void setupDisplay(){
  display.begin();
  display.setPowerSave(0); //modo powerSave (0-Off ? 1-On)
  display.setFont(u8x8_font_torussansbold8_u); //fonte utilizada
}

void iniciaVazao(gpio_num_t Port){
  gpio_set_direction(Port, GPIO_MODE_INPUT);
  gpio_set_intr_type(Port, GPIO_INTR_NEGEDGE);
  gpio_set_pull_mode(Port, GPIO_PULLUP_ONLY);
  gpio_intr_enable(Port);
  gpio_install_isr_service(0);
  gpio_isr_handler_add(Port, gpio_isr_handler_up, (void*) Port);
}

void send(){

  uint8_t values[gpioCount];

  humidity = dht.readHumidity();
  temp = dht.readTemperature();

```

```

    Serial.println(humidity);
    Serial.println(temp);

vazao = pulsos_vazao/5.5;
    mililiters = vazao / 60;    // Calculo de mililitros
    liters = liters + mililiters;
    pulsos_vazao = 0;
    Serial.println("Leitura do Sensor de Vazao:");
    Serial.println(liters);

sensorValue = analogRead(A0); // read analog input pin 0
Serial.print(sensorValue, DEC); // prints the value read
Serial.print(" \n"); // prints a space between the numbers
if(analogRead(A0) > 3300){ //SE LEITURA DO PINO FOR IGUAL A 1 (HIGH), FAZ
    digitalWrite(gpios[3], HIGH); //ACENDE O LED
}
else if(analogRead(A0) < 2800){ //SENÃO, FAZ
    digitalWrite(gpios[3], LOW); //DESLIGA O LED
}

if(analogRead(A0) < 1200){ //SE LEITURA DO PINO FOR IGUAL A 1 (HIGH), FAZ
    digitalWrite(gpios[2], HIGH); //ACENDE A LUZ
}
else if (analogRead(A0) > 1800){ //SENÃO, FAZ
    digitalWrite(gpios[2], LOW); //DESLIGA A LUZ
}

if(humidity < MIN_HUMIDITY)
{
    Serial.println("Umidade mínima atingida, enviando sinal para ligar o relê");
    digitalWrite(gpios[0], HIGH);
    if(liters >5){
        digitalWrite(gpios[0], LOW);
        liters = 0;
        Serial.println(liters);
    }
}
if(humidity > MIN_HUMIDITY)
{
    Serial.println("Umidade máxima atingida, enviando sinal para desligar o relê");
    if(liters >5){
        digitalWrite(gpios[0], LOW);
        liters = 0;
    }
}
if(temp > MAX_TEMP)
{
    Serial.println("TEMPERATURA máxima atingida, enviando sinal para ligar o relê");
    digitalWrite(gpios[1], HIGH);
}

```

```

}
else if(temp < MIN_TEMP)
{
  Serial.println("TEMPERATURA mínima atingida, enviando sinal para desligar o
relê");
  digitalWrite(gpios[1], LOW);
}

for(int i=0; i<gpioCount; i++){
  values[i] = digitalRead(gpios[i]);
  Serial.println(values[i]);
}

uint8_t broadcast[] = {0xFF, 0xFF,0xFF,0xFF,0xFF,0xFF};
esp_err_t result = esp_now_send(broadcast, (uint8_t*) &values, sizeof(values));
Serial.print("Send Status: ");
if (result == ESP_OK) {
  Serial.println("Success");
}
else {
  Serial.println("Error");
}
}

void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
  char macStr[18];
  snprintf(macStr, sizeof(macStr), "%02x:%02x:%02x:%02x:%02x:%02x",
  mac_addr[0], mac_addr[1], mac_addr[2], mac_addr[3], mac_addr[4], mac_addr[5]);
  Serial.print("Sent to: ");
  Serial.println(macStr);
  Serial.print("Status: ");
  Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Success" : "Fail");
  send();
}

void loop() {

  showOnDisplay();

}

void showOnDisplay(){
  String strname = "EUCALIPTU'S";
  String strname1 = "GREENHOUSE";
  String strHumdity = "UMIDADE: " + String(humidity);
  String strTemp = "TEMP: " + String(temp);
  String strVazao = "VAZAO: " + String(liters);
  String strLdr = "LDR: " + String(sensorValue);

```

```
display.clearLine(0);
display.drawString(2, 0, strname.c_str());
display.drawString(2, 1, strname1.c_str());
display.drawString(0, 3, strHumdity.c_str());
display.drawString(0, 4, strTemp.c_str());
display.drawString(0, 5, strVazao.c_str());
display.drawString(0, 6, strLdr.c_str());
}
```

APÊNDICE II - CÓDIGO ESCRAVO

```

#include <esp_now.h>
#include <WiFi.h>

uint8_t gpios[] = {2, 18, 19, 23};

int gpioCount;

void setup() {
  Serial.begin(115200);

  gpioCount = sizeof(gpios)/sizeof(uint8_t);

  WiFi.mode(WIFI_STA);
  InitESPNow();
  esp_now_register_recv_cb(OnDataRecv);

  for(int i=0; i<gpioCount; i++){
    pinMode(gpios[i], OUTPUT);
  }
}

void InitESPNow() {
  if (esp_now_init() == ESP_OK) {
    Serial.println("ESPNow Init Success");
  }
  else {
    Serial.println("ESPNow Init Failed");
    ESP.restart();
  }
}

void OnDataRecv(const uint8_t *mac_addr, const uint8_t *data, int data_len) {
  char macStr[18];
  //Cópia o Mac Address origem para uma string
  snprintf(macStr, sizeof(macStr), "%02x:%02x:%02x:%02x:%02x:%02x",
           mac_addr[0], mac_addr[1], mac_addr[2], mac_addr[3], mac_addr[4],
           mac_addr[5]);
  //Exibição do Mac Address que foi a origem da mensagem
  Serial.print("Received from: ");
  Serial.println(macStr);
  Serial.println("");

  for(int i=0; i<gpioCount; i++){
    digitalWrite(gpios[i], data[i]);
  }
}

void loop() {
}

```