



AUTOMAÇÃO E MONITORAMENTO INTELIGENTE DE UMA REDE DE PROCESSO AGRÍCOLA EM AMBIENTE PROTEGIDO

Lucas Anderson Ribeiro*¹ e Leonardo Alves Messias²

¹ UnB - Universidade de Brasília, ² IFG - Instituto Federal de Goiás

Resumo - O objetivo do presente trabalho visa a automação de um sistema de controle agrário, que proporcione monitoramento, controle e ajuste de parâmetros de uma cultura qualquer, apresentando uma solução simples e eficaz, que proporciona menor intervenção e maior segurança no controle de variáveis importantes para o crescimento saudável de hortaliças em ambiente fechado, tais como temperatura, umidade relativa do ar, umidade do solo e luminosidade. Dessa forma, implementou-se a prototipação de uma estufa em pequena escala para comportar a cultura tida como objeto de estudo, aliada ao sistema de monitoramento e controle desenvolvido em plataforma web, retornando ao operador, numérico e graficamente, os valores das variáveis sensoriadas, a fim de ajustar os atuadores aos parâmetros que melhor proporcionem um crescimento saudável da cultura.

Palavras-Chave - automação, controle, estufa, hortaliças, monitoramento.

AUTOMATION AND INTELLIGENT MONITORING OF AN AGRICULTURAL PROCESS NETWORK IN A PROTECTED ENVIRONMENT

Abstract - The objective of this work is to automate an agricultural control system, which provides monitoring, control and adjustment of parameters of any crop, presenting a simple and effective solution, which provides less intervention and greater security in the control of important variables for the healthy growth of vegetables indoors, such as temperature, relative humidity, soil moisture and light. In this way, the prototype of a small-scale greenhouse was implemented to accommodate the culture considered as the object of study, combined with the monitoring and control system developed on a web platform, returning to the operator, numerically and graphically, the values of the sensed variables, in order to adjust the actuators to the parameters that best provide a healthy growth of the crop.

Keywords - automation, control, greenhouse, monitoring, vegetables.

*lucas.and.rib@gmail.com

I. INTRODUÇÃO

O clima é um fator de grande influência na produção de hortaliças, no verão, as chuvas demasiadas danificam as hortaliças e criam condições favoráveis para o aparecimento de doenças. Ademais, o frio e os ventos do inverno acabam prolongando o ciclo dessas culturas. Com a necessidade de diminuir ou até mesmo erradicar tais contratempos, algumas estruturas e técnicas de proteção do ambiente agrícola foram criadas e desenvolvidas ao longo dos anos.

A grande diferença entre a produção agropecuária e as demais esferas da economia consiste, principalmente, na exposição às condições ambientais. Assim, é possível compreender que a agricultura é uma atividade de elevado risco, uma vez que não se têm controle de tais condições climáticas.

A fim de diminuir ou até mesmo erradicar os erros comuns no manejo das plantações, a utilização de sistemas de controle e automação em estufas vem se tornando cada vez mais comuns, na busca de tecnologias que facilitem o dia a dia do produtor, garantindo o total controle do ambiente, proporcionando eficiência na produtividade e economia de produtos.

Ante o exposto, o presente trabalho visa apresentar o desenvolvimento de um sistema de controle de processo agrícola em ambiente protegido, na qual utiliza-se estratégia de monitoramento avançado de dados, tais como temperatura, umidades do ar e do solo e luminosidade, que proporcionam a supervisão e comparação das variáveis de controle do microclima, auxiliando nas tomadas de decisão a respeito do cultivo.

II. FUNDAMENTAÇÃO TEÓRICA

A. Microcontrolador ESP32

Trata-se de um dispositivo IoT (*Internet of Things*) que emprega um microprocessador Tensilica Xtensa LX6 com suporte embutido para rede Wi-Fi, Bluetooth e memória flash integrada. Possui uma arquitetura que possibilita uma programação independente, sem a necessidade de outros microcontroladores, como é o exemplo do Arduino (Oliveira, 2017). Dentre as principais características que definem o microcontrolador ESP32, destacam-se o alto desempenho de potência aliado a um baixo consumo de energia, amplificador de baixo ruído, além da versatilidade e confiabilidade.

B. Sensores

Sensores são dispositivos eletroeletrônicos que possuem a capacidade de detectar e responder a estímulos químicos, físicos e/ou biológicos, produzindo a partir deles sinais que podem ser transformados em outra grandeza, com o objetivo de medir e observar os valores detectados.

- **Sensor de Temperatura e Umidade do Ar DHT22** - Este é um termômetro de boa precisão e sensibilidade para aplicação objetivada, com uma faixa de medição de -40 a +80 °C e precisão de mais ou menos 0,5 °C, para a temperatura, e uma faixa de umidade relativa de 0 a 100%, com precisão de 2%.
- **Sensor de Umidade do Solo FC-28** - O sensor de umidade do solo FC-28 é capaz de detectar as variações no ambiente em que é implantado, com a saída em estado alto quando o solo está seco, e estado baixo com o solo úmido. Possui duas sondas que transportam a corrente através do solo, realizando a leitura da resistência para analisar a umidade ideal. Possui tensão de operação de 3,3 a - 5 V.
- **Sensor de Luminosidade LDR** - O sensor de luminosidade LDR (*Light Dependent Resistor*) é um componente cuja resistência varia de acordo com a intensidade da luz. Em termos gerais, quanto maior a quantidade de luz incidindo sobre o dispositivo, menor será sua resistência, e quanto menor a luz, maior sua resistência.

C. Atuadores

Os atuadores são dispositivos que produzem movimento ou o controle de cargas e mecanismos, convertendo energia pneumática, elétrica ou hidráulica, em energia mecânica, que será enviada aos vários tipos de processos, por meio de sistemas de transmissão, para que haja um determinado trabalho.

- **Exaustor** - O exaustor é um *cooler* ou ventilador para a renovação do ar em ambientes com pouca ventilação, com a finalidade de manter o espaço climaticamente ideal.
- **Mini Bomba d'água Submersível** - A bomba d'água submersível é um motor de tamanho pequeno com a função de movimentar a água de um recipiente, permitindo uma maior troca de gases entre os ambientes. Sendo utilizada para promover a irrigação dentro da estufa.
- **Fita LED** - Em ambientes protegidos com pouca incidência de luz, como é o caso de uma estufa, por exemplo, é recomendada a utilização de algum tipo de iluminação artificial para contribuir no desenvolvimento da cultura, onde faz-se necessário o controle da incidência de luz.
- **Placa de Aquecimento de Cerâmica** - Em momentos frios, de menores temperaturas, faz-se necessária a utilização de uma placa de aquecimento de cerâmica de 12

V, que pode chegar a uma temperatura de 120 °C através do aquecimento de suas resistências internas, visando o controle de temperatura e umidade dentro da estufa com o objetivo de controlar condições climáticas adversas.

- **Piezoelétrico** - Para o controle de umidade relativa do ar, lidando especificamente com o aumento gradual desta, utilizou-se do efeito piezoelétrico, fundamental para o desenvolvimento das ondas ultrassônicas, isto é, através destas ondas, a placa piezoelétrica transformará a água que chega em estado líquido, aderida ao canudo de algodão, em água em estado gasoso, umidificando o interior do ambiente protegido.
- **Motor de Passo** - O motor de passo é um tipo de motor que tem a capacidade de converter um sinal de entrada em um ângulo de rotação, com extrema precisão de giro e posicionamento. No projeto, foi de suma importância para realizar a abertura e fechamento da janela da estufa, ligada às lógicas de umidificação e desumidificação do ambiente.

D. Linguagens de Programação e Ferramentas

- **Linguagem de Programação C** - A linguagem de programação principal escolhida para o desenvolvimento do código fonte foi a linguagem C pois, além da flexibilidade proporcionada pelo C, essa linguagem nos permite a programação de microcontroladores com menor tempo de compilação, ideal para a otimização do tempo de resposta do projeto. Sua maior vantagem é a fácil utilização em qualquer plataforma para gerar programas de alta eficiência.
- **Linguagem de Marcação HTML** - *HyperText Markup Language* é uma linguagem de marcação utilizada na construção de páginas *web*. O HTML é um modelo de linguagem mundial que consegue ser interpretado em qualquer navegador disponível, exibindo as páginas por trás dos códigos da mesma maneira.
- **Linguagem de Programação CSS** - A linguagem CSS (*Cascading Style Sheets*) é, de modo geral, um mecanismo para adicionar estilo a um documento *web*. O código CSS pode ser aplicado diretamente nas tags ou ficar contido dentro das tags *style*, visando boas práticas de código limpo, os scripts em CSS foram aplicados em arquivos distintos aos de linguagem HTML.
- **Linguagem de Programação JavaScript** - JavaScript é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma. Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da *World Wide Web*.
- **Google Firebase** - Para a criação e modelagem da aplicação *web* com foco no controle e supervisão das variáveis da estufa, tal ferramenta possibilitou vincular a aplicação *web* ao armazenamento em nuvem de back-end

em tempo real, armazenando os dados em documentos do tipo JSON.

III. METODOLOGIA

A abordagem introdutória ao tema decorreu-se através de pesquisas descritivas, explanando-se bibliografias, relatórios técnicos e artigos científicos sobre a cultura em ambiente protegido juntamente à tecnologias já existentes sobre o assunto, visando desenvolver um mecanismo de rede inteligente em ambiente protegido, com controle remoto preciso de uma plantação.

Através da análise das informações obtidas e estudo das possíveis formas de se manter uma estufa ou ambiente protegido com determinada aplicação tecnológica, partiu-se para implementação do projeto juntamente com a aquisição dos componentes eletrônicos e materiais necessários na aplicação, tais como microcontrolador, sensores de umidade e temperatura, atuadores, além de softwares e ferramentas para testes e simulações.

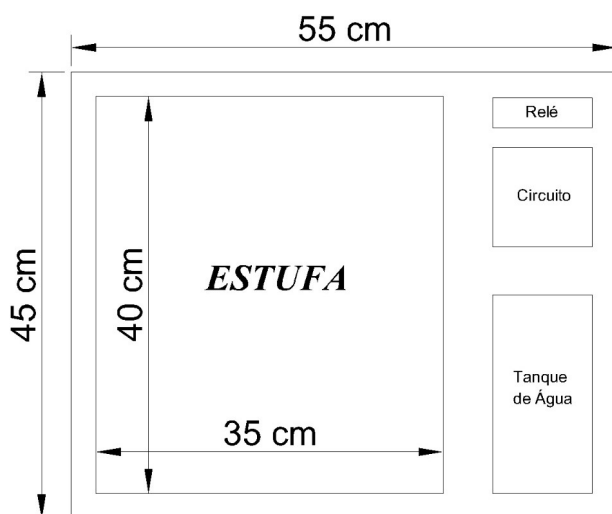
Após a definição dos materiais e componentes necessários para a realização do projeto, assim como a aquisição dos mesmos, foi realizada a programação do microcontrolador utilizado em linguagem C. Com o estudo anterior finalizado, foi realizada a construção de um protótipo do projeto de um ambiente escalável e possível de monitoramento remoto via web e ou smartphone, no qual poderemos realizar análise do solo, água, umidade, vento, luminosidade do ambiente.

IV. ESTRUTURA DO PROJETO

A. Arquitetura Funcional

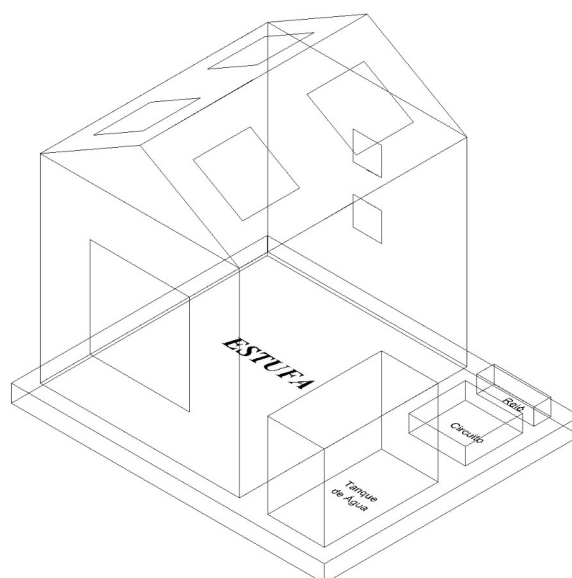
Na arquitetura funcional do sistema, buscou-se implementar uma estrutura, a fim de simular uma estufa em pequena escala que garantisse o desenvolvimento da solução criada. A ideia levou em conta a possibilidade de alocação de todos os componentes em um pequeno espaço, visando a organização e eficiência da estrutura, bem elucidada na figura [1].

Figura 1: Dimensões 2D da estrutura.



Além da função global do projeto, que é o controle automático em estufas, utilizando como referência soluções e trabalhos já implementados anteriormente, as análises realizadas para a construção da maquete basearam-se em um produto de utilização didática, que possibilita-se trabalhos futuros, portanto visando a montagem do sistema e o desenvolvimento da solução dimensionou-se a estufa, também, em modelagem tridimensional, ilustrado na figura [2].

Figura 2: Arquitetura conceitual 3D da estufa.



V. BIBLIOTECAS UTILIZADAS NA PLACA

Esta seção é dedicada às bibliotecas utilizadas e carregadas no script para viabilização e conclusão da programação envolvida no desenvolvimento do sistema, como ilustrado a seguir.

Bibliotecas.

```
// Importacao de bibliotecas:  
#include <Adafruit_Sensor.h>  
#include <FirebaseESP32.h>  
#include <ArduinoJson.h>  
#include <Stepper.h>  
#include <WiFi.h>  
#include <time.h>  
#include <DHT.h>
```

- **Biblioteca *Adafruit Sensor*** - Esta biblioteca atua reduzindo todos os dados a um único `sensors_event_type` e estabelecendo unidades SI específicas e padronizadas para cada família de sensores, os mesmos tipos de sensores retornam valores comparáveis a qualquer outro sensor similar.
- **Biblioteca *FirebaseESP32*** - Biblioteca de `client` do Firebase para ESP32. A biblioteca de `client` Firebase mais completa, rápida, segura e confiável que suporta

RTDB, Cloud Firestore, Firebase e Google Cloud Storage, Cloud Messaging e Cloud Functions para Firebase.

- **Biblioteca *ArduinoJson*** - *ArduinoJson* é uma biblioteca C++ JSON para Arduino, ESP32 e IoT (*Internet Of Things*) que possibilita a transação de informações entre a placa e o banco de dados em nuvem através de arquivos em formato JSON.
- **Biblioteca *Stepper*** - É a responsável por permitir que a placa ESP32 controle do motor de passo utilizado na maquete, ligado à lógica de desumidificação. Ela possibilita a manipulação dos motores unipolares ou bipolares, compatível com todas as arquiteturas.
- **Biblioteca *WiFi*** - Habilita a conexão de rede usando o *shield* WiFi da placa ESP32. Com esta biblioteca pode-se instanciar Servidores, Clients e enviar/receber pacotes UDP através de WiFi. O endereço IP pode ser atribuído estaticamente ou por meio de um protocolo DHCP. A biblioteca também pode gerenciar DNS.
- **Biblioteca *Time*** - Funcionalidade de cronometragem para o ESP32. Funções de data e hora, com provisões para sincronizar com fontes de tempo externas como GPS e protocolo NTP (Internet).
- **Biblioteca *DHT*** - A biblioteca DHT proporciona o correto funcionamento dos sensores DHT11 e DHT22, bem como de outros sensores de temperatura e umidade.

VI. RESULTADOS DO CÓDIGO FONTE DA PLACA

A. Função de Assimilação de Valores do Código da Placa ESP32

A função de assimilação do script fonte da placa ESP32 realiza a captura em tempo real dos valores booleanos atribuídos às variáveis de controle, o trecho de código abaixo exemplifica esta assimilação para os valores de iluminação. Ao capturar o valor 0, a iluminação local é desligada e de forma análoga, ao capturar o valor 1, a iluminação é ligada e em ambos os casos é logado no monitor serial, para fins de verificação.

Função de Assimilação dos Valores de Iluminação.

```
void assimilacao() {
    // Assimilacao dos valores de iluminacao - 0 ou 1:
    if(Firebase.getString(firebaseData,
        "/Atuadores/Iluminacao")) {
        String iluminacao_binario =
            firebaseData.stringData();

        if(iluminacao_binario.toInt() == 0) {
            digitalWrite(PINO_ILUMINACAO, LOW);
            Serial.println("- Iluminacao
                desligada.");
        } else {
            digitalWrite(PINO_ILUMINACAO, HIGH);
            Serial.println("- Iluminacao ligada.");
        }
    }
}
```

```
}
} else {
    Serial.print("[!] Erro ao capturar os dados
        em nuvem: [LATENCIA FIREBASE][ttl] -
        Iluminacao - ");
    Serial.print(firebaseData.errorReason());
    Serial.println(".");
}
}
```

B. Função de Atualização de Valores do Código da Placa ESP32

A função de atualização do script fonte da placa ESP32 realiza o envio em tempo real dos valores capturados pelos sensores conectados ao microcontrolador, o trecho de código abaixo exemplifica esta atualização para os valores de umidade do ar. Ao capturar um novo valor de medição, este será enviado ao banco de dados em nuvem Firebase, atualizando os valores da variável **UmidadeAr**.

Função de Atualização de Umidade do Ar.

```
void atualizacao() {
    // Atualizacao dos valores de umidade, DHT -->
    // banco de dados em nuvem:
    if(Firebase.get(firebaseData,
        "VariaveisMedidas/UmidadeAr")) {
        Firebase.set(firebaseData,
            "/VariaveisMedidas/UmidadeAr", umidade);
    } else {
        Serial.print("[!] Erro ao enviar os dados em
            nuvem: [LATNCIA FIREBASE][ttl] -
            Umidade do Ar - ");
        Serial.print(firebaseData.errorReason());
        Serial.println(".");
    }
}
```

C. Função de Atualização de Tempo de Protocolo NTP do Código da Placa ESP32

Para tornar possível a irrigação exclusivamente em tempos específicos, determinados e parametrizados pelo usuário para o crescimento mais saudável da cultura, fez-se necessário a utilização de um protocolo NTP que realizasse a captura em tempo real das horas, minutos, segundos e dias da semana e os armazenassem em variáveis utilizadas para estabelecer futuras condições de acionamento da bomba d'água.

Função de Atualização de Tempo.

```
void tempo() {
    // Variaveis de captura do servidor NTP:
    char hora[3];
    char minuto[3];
    char dia_da_semana[10];

    // Captura de data de hora:
    struct tm informacoes_do_tempo;

    if(!getLocalTime(&informacoes_do_tempo)) {
        Serial.println("[!] Erro ao capturar os
            dados do servidor NTP.");
    }
}
```

```

    return;
}

// Data e hora formatados:
strftime(hora, 3, "%H", &informacoes_do_tempo);
strftime(minuto, 3, "%M",
    &informacoes_do_tempo);
strftime(dia_da_semana, 10, "%A",
    &informacoes_do_tempo);

// Data e hora atuais:
Serial.print("[?] Estufa Automatica | Horario
    corrente: ");
Serial.print(hora);
Serial.print(":");
Serial.print(minuto);
Serial.print(" - ");
}

```

D. Função de Controle de Atuadores do Código da Placa ESP32

A função de controle geral estabelece dois tipos de funcionamento para o ambiente, sendo o primeiro deles quando a variável booleana **controle_binario** é igual a 0, assumindo assim apenas o controle manual, isto é, os atuadores funcionarão a partir do acionamento ON-OFF através da aplicação *web*, caso contrário, se o valor for igual a 1, o controle passa a ser automático, para o trecho de código apresentado abaixo, temos que este controle acionará a iluminação caso o valor medido pelo sensor esteja abaixo do mínimo configurado pelo usuário, de forma análoga, desacionará a iluminação caso o valor sensoriado esteja acima do valor mínimo.

Função de Controle Geral.

```

void controle() {
    // Assimilao dos valores de controle - 0
    [manual] ou 1 [automatico]:
    if(Firebase.getString(firebaseData,
        "/Variaveis/Controle")) {
        String controle_binario =
            firebaseData.stringData();

        if(controle_binario.toInt() == 0) {
            Serial.println(" Controle manual.");
        } else {
            // Logica de acionamento de iluminaao
            artificial:
            if(Firebase.getString(firebaseData,
                "/Variaveis/LuminosidadeMinima")) {
                String luminosidade_minima =
                    firebaseData.stringData();

                Serial.print(" Luminosidade minima:
                    ");
                Serial.print(luminosidade_minima);
                Serial.println(" Lux");

                // Controle do atuador - lampada

```

```

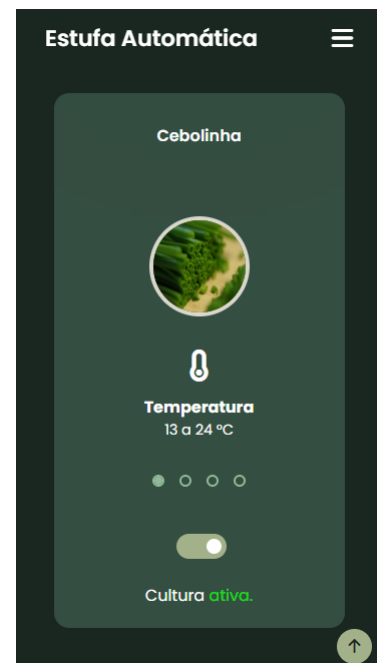
        incandescente:
        if(luminosidade <
            luminosidade_minima.toInt()) {
            Firebase.set(firebaseData,
                "/Atuadores/Iluminacao", "1");
        } else {
            Firebase.set(firebaseData,
                "/Atuadores/Iluminacao", "0");
        }
    } else {
        Serial.print("[!] Erro ao enviar os
            dados em nuvem: [LATENCIA
            FIREBASE][ttl] - Lampada
            Incandescente - ");
        Serial.print(firebaseData.
            errorReason());
        Serial.println(".");
    }
}

```

VII. RESULTADOS DA APLICAÇÃO WEB FINAL

A criação do site baseou-se em uma aplicação didática e de fácil acesso para qualquer usuário, seja agrônomo ou não, nas versões *Mobile* ou *Desktop*, a depender do *hardware* utilizado pelo usuário. Ao acessar o site, através do link <https://estufa-automatica.netlify.app/>, o operador é conduzido automaticamente para a tela principal da aplicação, que possui cinco ramificações, podendo ser escolhida a opção desejada: a tela Sobre, a tela Culturas, a tela Monitoramento, a tela Controle e a tela *Dashboard*.

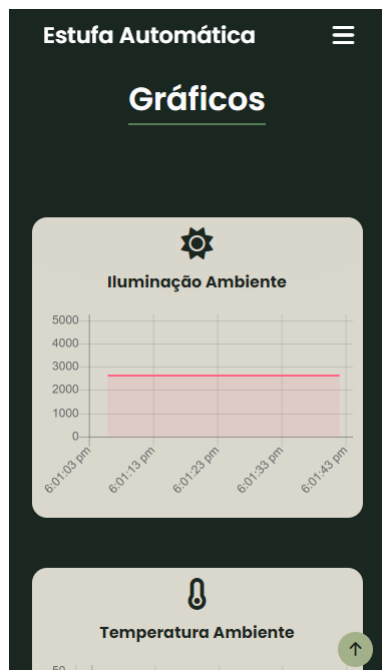
Figura 3: Card da Cultura Cebolinha.



A tela de maior importância para o projeto, como ilustra a figura [3] onde pode-se acompanhar os parâmetros a serem obedecidos pela placa para controle dos atuadores e detectar se esta está ativa ou não. Caso os parâmetros padrões não satisfaçam a cultura a ser cultivada, pode-se gerar uma nova

cultura através da aplicação web que irá reparametrizar toda a placa microcontrolada, gerando novos comportamentos para os atuadores.

Figura 4: Dashboard para Acompanhamento de Iluminação.



Para acompanhamento em tempo real, foi desenvolvido, também, uma página de *dashboards* que possui atualização de todas as variáveis sensorizadas a cada 1 segundo, permitindo estudos acerca das variações de medições e monitoramento contínuo.

VIII. RESULTADOS DA PLANTA BASE

Para os testes do sistema integrado e a possibilidade de colocá-lo em prática, foi necessária a realização de testes dos componentes de forma individual, juntamente com a conexão da aplicação web ao banco de dados Firebase, possibilitando a troca de informações em tempo real para o funcionamento integral do site.

Cada atuador foi testado separadamente, acompanhando pelo site a variação das condições climáticas, a partir dos parâmetros definidos para o desenvolvimento da cultura escolhida como objeto de estudo deste: o coentro.

Com a finalização dos testes e a aprovação da utilização dos sensores em conjunto com os atuadores para o desenvolvimento da cultura de coentro, e a integração de todas as partes do sistema (*software* e *hardware*), foi possível colocar em prática o sistema idealizado para conclusão deste trabalho e obter os resultados necessários para viabilização do produto desenvolvido, atingindo um crescimento mais saudável e mais verde da plantação em relação às culturas fora do ambiente protegido, como ilustra a figura [5].

Figura 5: Planta Base com Cultivo de Coentro.



IX. CONCLUSÕES

Com a implantação de um sistema integrado capaz de automatizar tais produções agrícolas, tal qual a Estufa Automática, alcançar-se-á um cultivo sujeito à minimização de interações humanas, tornando este mais preciso e menos suscetível a erros, além de, claro, maior performance e saúde no crescimento da cultura.

O sistema de controle desenvolvido para o trabalho em questão pode ser aplicado para qualquer cultura vegetal em pequena escala, em especial hortaliças, desde que sejam definidos previamente os parâmetros necessários para as variáveis de controle dentro do ambiente protegido, tidos como essenciais para o desenvolvimento sustentável da plantação.

Procurando adequar o projeto de forma a sanar uma necessidade real do mercado, com o estudo dos princípios de funcionamento e modelagem do projeto, conclui-se que o projeto é tecnicamente viável e eficaz, isto é, atinge-se a maturação da cultura final em um menor período de tempo, cerca de 3 a 5 dias, minimizando-se o gasto de recursos.

REFERÊNCIAS

- [1] FILHO, Aldir. C. M. **Sistema de Automação e Controle Inteligente para Cultivo Protegido - Tecnologia Acessível ao Pequeno Produtor**. 2017. 134 f. Monografia, IFSC, 2017.
- [2] MENDEZ, M. E. G. **Caracterização climática e manejo de ambientes protegidos: a experiência brasileira.**; Informe Agropecuário, Belo Horizonte, v.20, n.200/201, p. 15-23, 1999.
- [3] **Script completo desenvolvido no projeto. Disponível clicando aqui.**