

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

TELMO BINS VOLKMANN

**IMPLANTAÇÃO DE SISTEMA IOT PARA
AUTOMAÇÃO RESIDENCIAL E
IRRIGAÇÃO DE JARDINS**

Porto Alegre
2022

TELMO BINS VOLKMANN

**IMPLANTAÇÃO DE SISTEMA IOT PARA
AUTOMAÇÃO RESIDENCIAL E
IRRIGAÇÃO DE JARDINS**

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

ORIENTADOR:

Prof. Dr. Marcelo Götz

Porto Alegre
2022

TELMO BINS VOLKMANN

**IMPLANTAÇÃO DE SISTEMA IOT PARA
AUTOMAÇÃO RESIDENCIAL E
IRRIGAÇÃO DE JARDINS**

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

Orientador: _____
Prof. Dr. Marcelo Götz, UFRGS
Doutor pela Universität Paderborn – Paderborn, Alemanha)

Banca Examinadora:

Prof. Dr. Marcelo Götz, UFRGS
Doutor pela Universität Paderborn – Paderborn, Alemanha)

Prof. Dr. Renato Ventura Bayan Henriques, UFRGS
Doutor pela Universidade de Minas Gerais - Brasil)

Prof. Dr. Fabiano Disconzi Wildner, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul - Brasil

Mário Roland Sobczyk Sobrinho
Coordenador de Curso
Eng. de Controle e Automação

Porto Alegre, maio de 2022.

AGRADECIMENTOS

Gostaria de agradecer primeiramente à minha família por todo o apoio e auxílio durante o período de faculdade e de execução deste trabalho. Em especial aos meus pais por cederem a casa para a instalação do projeto e pelo auxílio no monitoramento do sistema enquanto eu não estava presente. Também gostaria de agradecer aos meus colegas de graduação, que participaram deste processo comigo auxiliando nos estudos e compartilhando histórias e opiniões. Agradeço também à universidade e aos professores por todo o conhecimento adquirido e pela preparação para o mercado de trabalho.

Segue aqui um agradecimento especial ao colega Leonardo Zambotto por contribuir com este trabalho através da impressão 3D de um dos painéis utilizados e das caixas utilizadas para proteção dos sensores. Além disso, agradeço ao meu orientador pelo apoio, dicas e orientação na execução e no planejamento do projeto.

RESUMO

A procura por soluções de automação residencial e sistemas de irrigação de jardins está cada vez maior, porém o custo desse tipo de produto é alto e, até o momento, não existe uma solução comercial que integre a automação residencial com o controle de irrigação. Diante desse cenário, construiu-se a ideia de implementar um projeto que integre ambos os sistemas, seja controlado via celular e possua um baixo custo. Inicialmente, foi realizada uma pesquisa que comprova que existe um nicho de mercado para este produto. Então, foi definida a arquitetura do sistema como um todo, que se baseia no conceito de IoT, tendo como objetivo implementar um produto que seja capaz de controlar a irrigação e a iluminação do pátio da casa de praia do aluno. O sistema desenvolvido possui dois microcontroladores, sensores de luminosidade, temperatura e umidade, módulos com relés, válvulas solenoides, aspersores e uma bomba hidráulica, componentes que foram selecionados durante a execução do projeto. Além disso, foram implementados *softwares* para os microcontroladores, uma interface para celular, uma para computador, um banco de dados para armazenamento das informações e foi selecionado o protocolo MQTT e um *broker* para a comunicação entre eles. Então, os equipamentos foram instalados no pátio e os parâmetros para os controladores foram obtidos através de testes empíricos. Por fim, foi realizada uma pesquisa com o usuário final do produto, o gramado foi fotografado antes e depois do uso do sistema por 60 dias, foram gerados gráficos com as variáveis relevantes para análise do comportamento dos controladores e foi realizado o cálculo do custo total de implantação e sua comparação com o custo de produtos de irrigação de jardins e de automação residencial existentes no mercado. Uma vez realizadas essas análises, é possível verificar a qualidade do produto obtido, o alto valor agregado da integração de sistemas de irrigação e de automação residencial e o correto comportamento do sistema.

Palavras-chave: Automação Residencial, Controle de Irrigação, Controle de Iluminação, Engenharia de Controle e Automação, Eletrônica, Instrumentação, Internet das Coisas, Microcontroladores, Protocolo MQTT.

ABSTRACT

The demand for home automation and garden irrigation solutions is increasing rapidly, however, the price to acquire this kind of product is still high and there is no product on the market that integrates home automation and irrigation systems. Given this scenario, the concept of this project was conceived, which consists of the implementation of a system that integrates both the demanded solutions, allow their control via cellphone and can be built with a low cost. The project started with a market survey, which ensured the existence of a market niche for this kind of solution. Then, the systems architecture was defined based on the IoT concept and its main capabilities were specified. It shall be able to control the external lights and the garden irrigation of the students beach house. The developed system is composed by two microcontrollers, a brightness sensor, two umidity sensors, a temperture sensor, relay modules, solenoid valves, sprinklers and a hydraulic pump. During the project execution, each component supplier and model were selected to meet the application needs. Furthermore, firmwares for the microcontrollers, a cellphone app, a computer app and a database were implemented. For the communication between these softwares, the MQTT protocol and the adafruit broker were chosen. After that, the equipments were installed on the yard and the controllers parameters were obtained through empirical tests. Finally, the following procedures were performed to evaluate the product: a survey with the products operators, a comparisson between photos of the garden taken before and after running the systems irrigation control for 60 days, generation and analysis of graphs with the relevant data to asses the controllers behavior, and calculation of the total cost for the product deployment and its comparisson with the price of comercial products for garden irrigation and home automation. After the execution of the previous procedures, it was possible to prove the products quality, the high added value related to the integration between irrigation and home automation systems, and the proper functioning of the solution.

Keywords: Home Automation, Irrigation Control, Lights Control, Control and Automation Engineering, Electronics, Instrumentation, Internet of Things, Microcontrollers, MQTT Protocol.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	8
LISTA DE TABELAS	9
LISTA DE ABREVIATURAS	10
LISTA DE SÍMBOLOS	11
1 INTRODUÇÃO	12
1.1 Objetivos do Trabalho	14
1.2 Organização	14
2 REVISÃO DA LITERATURA	15
2.1 Sistemas IoT e o Protocolo MQTT	15
2.2 Constituição do Solo e sua Umidade Ideal	16
3 METODOLOGIA	17
3.1 Definição do Protocolo de Comunicação e do <i>Broker</i>	19
3.2 Implementação do Servidor e do Banco de Dados	20
3.3 Especificação dos Componentes	20
3.3.1 Componentes Hidráulicos	20
3.3.2 Componentes Elétricos	22
3.3.3 Componentes de <i>Hardware</i>	22
3.3.3.1 Especificação do Microcontrolador	22
3.3.3.2 Especificação do Conversor CA/CC	23
3.3.3.3 Especificação do Módulo de Relés	23
3.3.3.4 Especificação do Transformador	23
3.3.3.5 Especificação dos Sensores	24
3.4 Diagramas Elétricos e Montagem das Placas dos Controladores	24
3.5 <i>Softwares</i> dos Microcontroladores	26
3.5.1 <i>Software</i> do Controlador Principal	26
3.5.2 <i>Software</i> do Controlador da Bomba	28
3.6 Localização dos Equipamentos no Pátio	28
3.7 Instalação dos Equipamentos no Pátio	29
3.7.1 Painel Principal e seus Sensores	29
3.7.2 Sistema de Iluminação	31
3.7.3 Painel do Controlador da Bomba e Sistema de Acionamento	32
3.7.4 Sistema Hidráulico	33
3.8 Calibração do Sensor de Umidade	34
3.9 Parâmetros para Controle dos Modos Automáticos	35

3.10	Interfaces para Computador e Celular	36
4	RESULTADOS E DISCUSSÕES	38
5	CONCLUSÃO	42
	APÊNDICE A - DIAGRAMAS E FIGURAS	43
A.1	Figuras das Etapas do Ensaio de Irrigação do Solo	43
A.2	Figura das Etapas do Ensaio para Obter o Tempo de Absorção de Água pelo Solo	44
A.3	Figura Ensaio Definição <i>Setpoints</i> Luminosidade	45
A.4	Diagrama Elétrico da Placa do Controlador Principal	45
	APÊNDICE B - CÓDIGOS FONTE DOS <i>SOFTWARES</i> DO PROJETO	46
B.1	Arquivo com Configurações de Rede e do Servidor MQTT para os dois Controladores	46
B.2	Código Fonte <i>Software</i> Controlador Principal	46
B.3	Código Fonte <i>Software</i> Controlador Bomba	51
	REFERÊNCIAS	52

LISTA DE ILUSTRAÇÕES

1	Resultado Pesquisa Mercado: Possui casa com jardim e acredita que sensor de umidade agrega valor	13
2	Resultado Pesquisa Mercado: se interessaria pelo produto e pagaria pelo produto	13
3	Arquitetura do Sistema	17
4	Arquitetura do Controlador Principal	18
5	Sistema de Comunicação e Interface com o Usuário	19
6	Localização dos Aspersores no Pátio	21
7	Projeto da Placa do Controlador Principal	25
8	Placa do Controlador Principal Montada	26
9	Modo Automático do Controle de Irrigação	27
10	Modo Automático do Controle de Luminosidade	28
11	Mapa com Localização dos Equipamentos	29
12	Instalação Painel Principal	30
13	Instalação Sensor Luminosidade	30
14	Instalação Sensor Umidade	31
15	Instalação Iluminação vista 1	31
16	Instalação Iluminação vista 2	32
17	Instalação Sistema e Painel de Acionamento da Bomba	32
18	Instalação Aspersores	33
19	Instalação Válvulas Solenoides	34
20	Ensaio Calibração Sensor de Umidade: Dados Obtidos	35
21	Interfaces do Sistema com o Usuário	37
22	Resultado Pesquisa Com Usuário	38
23	Comparação Gramado Antes e Após Ligar Modo Automático de Irrigação	39
24	Comportamento Modo Automático Irrigação	40
25	Comportamento Modo Automático Iluminação	40
26	Comportamento Leitura Temperatura	41
27	Ensaio Calibração Sensor de Umidade: Etapas e Materiais Utilizados	43
28	Ensaio Calibração Sensor de Umidade: Solo Seco e Saturado	44
29	Ensaio Tempo de Absorção de Umidade pelo Solo	44
30	Ensaio <i>Setpoints</i> Controle de Luminosidade	45
31	Diagrama Elétrico da Placa do Controlador Principal	45

LISTA DE TABELAS

1	Comparação Entre Modelos de Microcontroladores	22
2	Custos do Projeto	41

LISTA DE ABREVIATURAS

CA	Corrente Alternada
CC	Corrente Contínua
GND	<i>Ground</i> (Terra)
IoT	<i>Internet of Things</i> (Internet das Coisas)
MVP	<i>Minimum Viable Product</i> (Mínimo Produto Viável)
MQTT	<i>Message Queuing Telemetry Transport</i>
QoS	<i>Quality of Service</i> (qualidade de serviço)
TCP	<i>Transmission Control Protocol</i>

LISTA DE SÍMBOLOS

% Percentagem

A Ampere

CV Cavalo vapor

cm Centímetros

cm^3 Centímetros cúbicos

KB Kilobytes

m Metro

m^2 Metro quadrado

m^3/h Metro cúbico por hora

mA MilliAmperes

Mb Megabits

ml Mililitros

R\$ Reais

® Marca registrada

V Volts

1 INTRODUÇÃO

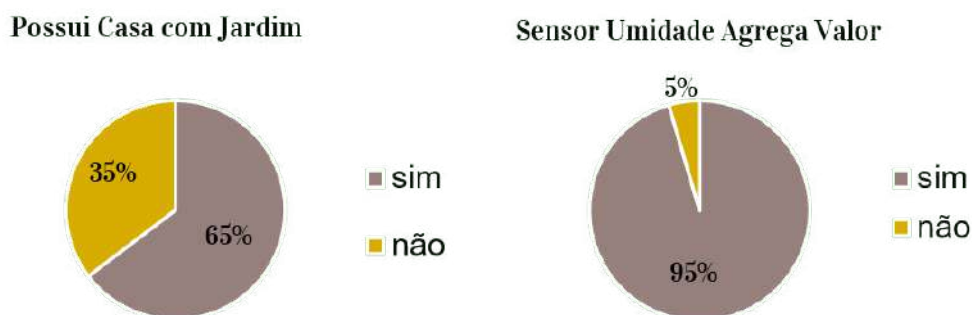
A procura por soluções de automação residencial está cada vez maior, o que permitiu que este setor crescesse 35% no Brasil no período entre 2007 e 2009 (PILOTI, 2014). Esse crescimento possibilitou o surgimento de diversas companhias no ramo, entre elas as empresas Homesystems (2021), Quantasmarthome (2021) e Foltech (2021), as quais oferecem produtos que integram sistemas de segurança, iluminação, climatização, entre outros. Apesar de sua expansão, o setor continua associado a um alto custo, o qual pode ser observado na pesquisa realizada por Piloti (2014), cujo resultado mostra que o investimento necessário para automatizar uma residência varia entre R\$ 2.000,00 e R\$ 50.000,00. Além disso, a análise dos serviços oferecidos por essas e outras organizações que atuam na área mostra que elas não têm expertise em irrigação de jardins, criando a necessidade de contratar empresas especializadas para oferecer este serviço sem incorporá-lo ao projeto de automação residencial.

Para análise do estado da arte do mercado de irrigação automatizada, pode-se tomar como base a realidade da Rainbird (2021), a qual é considerada um dos principais representantes desse setor. Dentre os produtos presentes em seu portfólio, observa-se a existência de controladores que podem receber comando via celular, os quais são comercializados por R\$ 1.600,00, e sensores de chuva, vendidos por R\$ 300,00, o que mostra o alto custo associado a este tipo de aplicação. Ainda, observa-se que os serviços oferecidos por esta companhia utilizam apenas temporização e/ou sensores de chuva para controlar a irrigação, não permitindo que o sinal de controle seja diretamente relacionado com a umidade presente no solo, o que seria o ideal para este caso.

Tendo em vista os dados apresentados, observa-se que o usuário que deseja automatizar sua residência e a irrigação de seu jardim precisa adquirir duas soluções independentes, o que além de tornar necessária a interação com duas interfaces de controle diferentes, também resulta no aumento significativo do valor que precisaria ser dispendido por ele.

Diante desse cenário, surge a ideia de implementar um sistema que permita a integração entre a automação residencial e o controle de irrigação, possua um baixo custo e possa ser controlado via celular. Junto a ela surge a proposta de inovar nos processos de controle existentes na área de irrigação, através do uso de sensores de umidade. Para este objetivo ser alcançado, torna-se relevante a realização de uma pesquisa para avaliar o interesse do público por este novo conceito, a qual foi realizada através da plataforma *Google Forms* e tem suas respostas apresentadas nas figuras 1 e 2.

Figura 1: Resultado Pesquisa Mercado: Possui casa com jardim e acredita que sensor de umidade agrega valor.



Fonte: Elaborado pelo autor.

Figura 2: Resultado pesquisa mercado: se interessaria pelo produto e pagaria pelo produto.



Fonte: Elaborado pelo autor.

A pesquisa contou com um total de 133 respondentes, sendo que 65% informaram que possuem casa com pátio, 95% se interessariam por esta solução caso tivessem uma casa com pátio, 93% pagariam por este produto e 95% consideram que o uso de sensores de umidade agregaria valor ao sistema de irrigação. Analisando os resultados encontrados pode-se comprovar a existência de um nicho de mercado para a ideia.

Uma vez comprovada a relevância do tema proposto, o presente trabalho se dispõe a validar este conceito através da implementação de um MVP (*Minimum Viable Product*), o qual consiste em um sistema que permite ao cliente realizar, através do celular, a leitura da temperatura ambiente e o controle da irrigação do gramado e da iluminação de seu pátio. O objetivo deste projeto é analisar a aplicabilidade dessa solução, apresentar resultados referentes ao seu custo-benefício e propor melhorias para transformá-la em um produto.

1.1 Objetivos do Trabalho

Os objetivos do trabalho são:

- Implementar um sistema de controle de luminosidade;
- Implementar um sistema de controle de irrigação que utilize sensores de umidade;
- Integrar ambos os sistemas em uma única solução que possa ser controlada via celular;
- Levantar dados referentes à aplicabilidade e à qualidade da solução apresentada e seu valor agregado;
- Levantar dados referentes ao custo para implementação do MVP e compará-los aos custos de soluções de mercado apresentados na introdução do presente trabalho;
- Propor melhorias para transformar a ideia em um produto.

1.2 Organização

A estrutura do presente trabalho está dividida em quatro capítulos, além da Introdução: Revisão da Literatura, Metodologia, Resultados e Discussões e Conclusão. O primeiro apresenta referências de tecnologias e trabalhos acadêmicos relacionados ao tema já desenvolvidos por outros autores e conceitos necessários para o entendimento desta pesquisa. O segundo apresenta todas as etapas executadas no projeto, incluindo a definição da arquitetura do sistema, a escolha do protocolo de comunicação, a escolha dos componentes, a montagem e instalação do produto e a implementação dos *softwares* necessários. O terceiro apresenta uma avaliação do produto obtido, incluindo uma pesquisa realizada com o usuário final, a análise de gráficos gerados a partir de dados coletados das variáveis de interesse para confirmar o correto funcionamento do sistema e a comparação do custo total para a implementação do sistema com os valores de produtos de irrigação de jardins e automação residencial existentes no mercado. O último apresenta considerações finais relativas à qualidade do produto e sua aplicabilidade, possíveis melhorias a serem aplicadas e trabalhos futuros relacionados ao projeto.

2 REVISÃO DA LITERATURA

Como ponto de partida para a construção deste projeto são analisados trabalhos de conclusão de curso, publicados por alunos do curso de Engenharia de Controle e Automação, referentes ao uso de sistemas IoT (Internet das Coisas, do inglês *Internet of Things*) para aplicações semelhantes. O primeiro trabalho, de autoria do colega Richard Adolphs (Adolphs (2020)), aborda o uso de sistemas IoT para irrigação de hortaliças domésticas de pequeno porte. O segundo, de autoria do colega Ethar Teixeira (Teixeira (2019)), aborda o uso de sistemas IoT para monitorar a temperatura, a umidade do ar e a luminosidade de um ambiente residencial.

O presente projeto pode ser considerado como uma integração das abordagens de ambos os trabalhos, implementando o controle de irrigação de jardins de maior porte integrado ao monitoramento de temperatura ambiente e luminosidade e adicionando o controle de iluminação externa. Além disso, outra característica observada que difere este sistema dos demais, consiste na adição da interação com o usuário através de comandos para ativar e desativar o modo automático ou controlar manualmente os sistemas de iluminação e de irrigação.

Uma vez citados os trabalhos que servem de referência para o presente projeto, esta seção segue com a apresentação dos principais conceitos necessários para o entendimento dos passos executados na sua elaboração.

2.1 Sistemas IoT e o Protocolo MQTT

Estes conceitos são de suma importância para o entendimento do projeto como um todo, pois o sistema a ser implementado trata-se de um sistema IoT e utiliza o protocolo MQTT (do inglês *Message Queuing Telemetry Transport*) para sua comunicação. A Internet das Coisas é um paradigma que vem ganhando espaço no cenário moderno das telecomunicações *wireless*. A ideia consiste em poder conectar uma variedade de objetos, como sensores, tags RFID, smartphones, computadores, e até objetos de uso mais comum entre si. Estes objetos geram um fluxo de dados e a conexão permite que eles transmitam estes dados para outros objetos no meio, formando assim uma internet de coisas (WANZELER; FULBER; MERLIN, 2016).

O MQTT é um protocolo de camada de aplicação que opera sobre o protocolo TCP (do inglês *Transmission Control Protocol*) na camada de transporte (ZOLETT; RAMIREZ, 2020), tendo se tornado um protocolo padrão para soluções IoT (YUAN, 2017). Este protocolo foi desenvolvido pela empresa IBM nos anos 1990 e têm as vantagens de ser um protocolo flexível e leve, permitindo sua aplicação em *hardwares* com baixa capacidade de processamento ou redes com alta latência ou limitação de banda (YUAN, 2017). Além disso, ele é um protocolo de mensagem com suporte para a comunicação assíncrona, ou

seja, que desacopla o emissor e o receptor da mensagem tanto no espaço quanto no tempo, sendo assim aplicável em ambientes com rede de conexão não confiável (YUAN, 2017).

Seu funcionamento é baseado em uma arquitetura de publicação e assinatura, na qual existem dois elementos principais: o servidor (que é chamado de *broker*) e o cliente. O *broker* é responsável por coordenar o fluxo de mensagem através de tópicos (chamados de *feeds*), que podem ser vistos como endereços de memória onde os clientes escrevem ou leem informações através do padrão de publicação e assinatura (ZOLETT; RAMIREZ, 2020). Ao iniciar a conexão com o *broker*, o cliente pode enviar uma ou mais mensagens chamadas de *subscribe*, para indicar sobre quais tópicos ele deseja receber informações (CONCEIÇÃO; RESENDE COSTA, 2019).

Outra característica importante dos protocolos MQTT é a possibilidade de trabalhar com três níveis de qualidade de serviço (QoS), os quais garantem diferentes confiabilidades para o envio e o recebimento de mensagens. O nível zero garante apenas o envio, não se preocupando com o recebimento. O nível um exige que o recebimento seja confirmado pelo menos uma vez, não impedindo múltiplos recebimentos. O nível dois garante que a mensagem seja recebida exatamente uma vez, impedindo múltiplos recebimentos (ZOLETT; RAMIREZ, 2020).

2.2 Constituição do Solo e sua Umidade Ideal

O solo é um material poroso constituído de três fases que interagem entre si: sólida, líquida e gasosa. Esse material poroso serve de apoio físico (sustentação), químico e biológico (nutrição) para o desenvolvimento da maioria dos vegetais (FRIZZONE, 2017). Uma de suas importantes funções é de atuar como um reservatório de água e nutrientes (solução do solo) para utilização pelas plantas.

A fração líquida do solo é representada pela solução do solo e a gasosa é composta pelo ar do solo que contém vapor da água e certa quantidade de O₂, CO₂, N₂, NH₃ e outros gases. A fase sólida dele é constituída de matéria mineral e orgânica de diferentes formas, tamanhos, constituição e orientação, constituindo a matriz do solo. A distribuição dos componentes sólidos quanto à forma e arranjo no perfil do solo é denominada de estrutura e a parte do solo não ocupada pela matriz é denominada de espaço poroso. O arranjo, a organização e a orientação das partículas determinam o espaço poroso disponível para a penetração de líquidos e de gases (FRIZZONE, 2017).

Quando o espaço poroso do solo está totalmente ocupado pela água diz-se que o solo está saturado. À medida que ocorre a drenagem da água o ar entra no espaço poroso e, nestas condições, diz-se que o solo está não saturado (espaço poroso com água e ar) (FRIZZONE, 2017).

Na prática da irrigação é importante manter certo balanço entre a porção dos poros ocupada pela água e ocupada pelo ar, evitando que o solo fique seco ou saturado. A composição, em volume, de um solo ideal tem as seguintes proporções: 45% de fração mineral, 5% de fração orgânica, 30% de fração de água e 20% fração de ar (FRIZZONE, 2017). Esta proporção será utilizada como referência para os ensaios que determinam o *setpoint* de umidade para iniciar a irrigação do solo.

3 METODOLOGIA

O presente projeto tem como objetivo implementar um sistema de controle de iluminação e irrigação de baixo custo que possa ser controlado via celular. O primeiro passo em seu desenvolvimento é a definição do local onde este produto será instalado, no caso, a casa de praia do aluno. Na sequência, são especificados os requisitos a serem atendidos por ele. Um deles consiste na existência de dois conjuntos de luminárias a serem controladas de forma independente, um referente ao paisagismo do pátio e outro à iluminação do ambiente. O segundo requisito consiste na definição de um setor específico onde devem ser instaladas as luminárias e onde deve ser realizada a irrigação, localizado na lateral da casa e com uma área de $127m^2$. O terceiro consiste na existência de dois modos de operação para os sistemas de controle, um referente ao acionamento manual e outro ao modo automático. O último consiste na necessidade de integrar o sistema de irrigação com uma bomba e um poço artesiano já existentes na casa, associando o conceito de sustentabilidade ao projeto, evitando o desperdício de água tratada e, conseqüentemente, gerando uma economia ao usuário final.

Uma vez especificados os requisitos do projeto, o próximo passo é a definição da arquitetura do mesmo, apresentada na Figura 3. Nota-se que o sistema possui dois controladores, o que é necessário devido a bomba estar instalada distante do setor onde é implementada a solução. Além disso, observa-se que sua arquitetura pode ser dividida em cinco partes: a primeira referente ao sistema hidráulico, a segunda à iluminação, a terceira ao controlador principal, a quarta ao controlador da bomba e a última à comunicação e interface com o usuário.

Figura 3: Arquitetura do Sistema.



Fonte: Elaborado pelo autor.

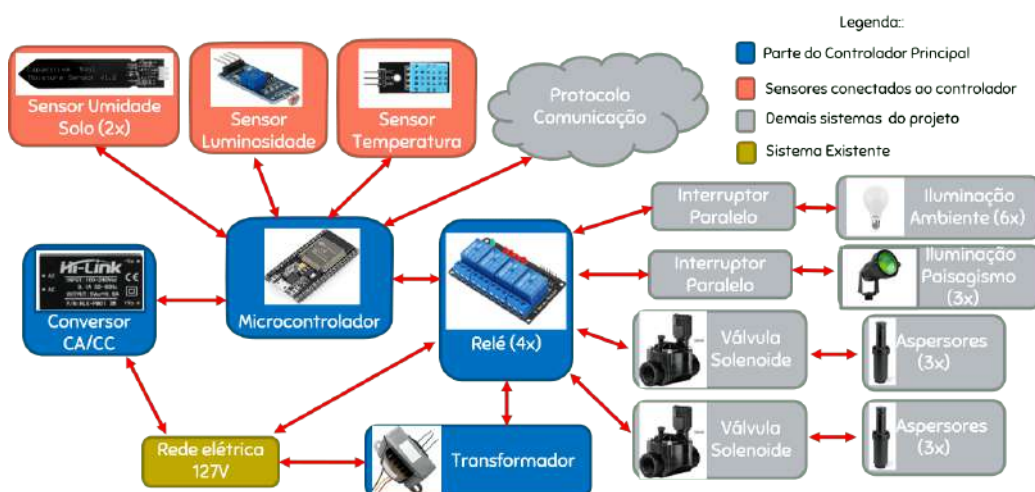
O sistema hidráulico é abastecido com água por um poço artesiano já existente, conectado a uma bomba hidráulica também já instalada no pátio. O fluxo é direcionado para um conjunto de seis aspersores distribuídos em duas linhas a serem acionadas de forma complementar, cuja necessidade será justificada na Seção 3.3.1, e duas válvulas

solenoides normalmente fechadas usadas para permitir o controle dessas linhas de forma independente.

O sistema de iluminação é composto por três luminárias utilizadas para o paisagismo, seis luminárias para a iluminação ambiente, e dois interruptores paralelos para permitir que o usuário controle a iluminação pelo celular ou diretamente através do painel instalado no pátio de acordo com sua necessidade.

O controlador principal é responsável por adquirir os dados referentes à umidade, temperatura e luminosidade, assim como realizar o controle dos sistemas de iluminação, de irrigação e do controlador da bomba. Sua arquitetura é apresentada na Figura 4. Ela é composta por um microcontrolador, um conversor de corrente alternada (CA) em corrente contínua (CC) usado para permitir sua alimentação com a tensão da rede, um conjunto de relés que tem como finalidade permitir o controle da iluminação e da irrigação, um transformador que converte a tensão de entrada do sistema em valores capazes de ativar válvulas de irrigação, dois sensores de umidade, um sensor de luminosidade e um sensor de temperatura.

Figura 4: Arquitetura do Controlador Principal.

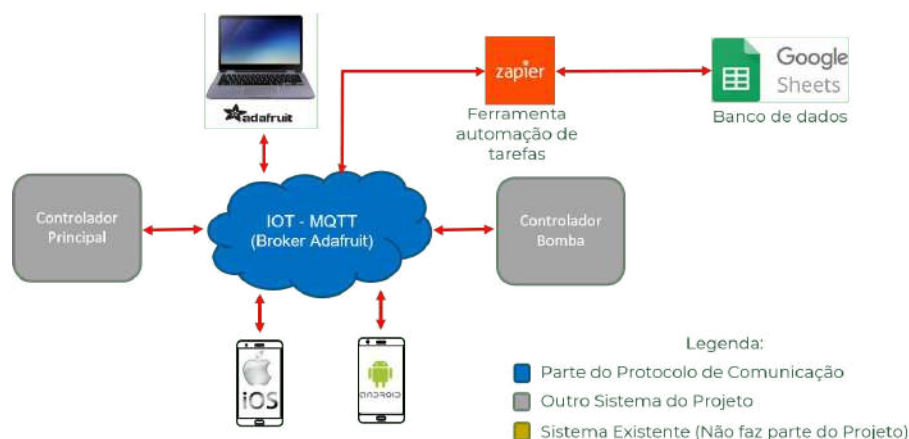


Fonte: Elaborado pelo autor.

O controlador da bomba funciona como um escravo do controlador principal, sendo responsável por acioná-la de acordo com os comandos recebidos. Ele é composto por um microcontrolador, um conversor CA/CC usado para permitir sua alimentação com a tensão da rede, e um conjunto de relés que tem como finalidade permitir o acionamento da bomba.

O sistema de comunicação e interface com o usuário consiste em protocolo para a comunicação entre um aplicativo celular, um aplicativo para computador e os dois microcontroladores. Ele também inclui os aplicativos para celular e para computador e um banco de dados para armazenamento das informações coletadas. Sua estrutura segue na Figura 5.

Figura 5: Sistema de Comunicação e Interface com o Usuário.



Fonte: Elaborado pelo autor.

Uma vez definida a arquitetura da solução, os próximos passos para a implementação desse projeto são: a escolha do protocolo de comunicação, a implementação do servidor e do banco de dados, a definição dos componentes utilizados, o desenho dos diagramas elétricos e a montagem das placas para os componentes eletrônicos, a implementação dos *Softwares* dos microcontroladores, a definição da localização dos equipamentos no pátio, a montagem do painel do controlador principal e seus sensores, a instalação do sistema de acionamento da bomba, a montagem do sistema hidráulico, a instalação do painel e do sistema de acionamento da bomba, a calibração dos sensores de umidade, a obtenção dos parâmetros para o controle dos modos automáticos, e a implementação dos aplicativos para computador e celular. Esses passos são descritos na sequência.

3.1 Definição do Protocolo de Comunicação e do *Broker*

Seguindo o padrão do mercado para sistemas IoT e levando em conta as características referentes ao MQTT levantadas na Seção 2.1, esse protocolo foi escolhido para realizar comunicação entre os controladores e os aplicativos do projeto. A escolha se deu pelo fato de ele suportar comunicação de forma assíncrona, permitindo seu uso em ambientes com conexão de rede não confiável, o que é muito comum em pátios e jardins. Além disso, sua flexibilidade facilita a implementação e a integração com diversos aplicativos e dispositivos.

Uma vez selecionado o protocolo de comunicação, é necessária a definição de um servidor para o gerenciamento da comunicação, chamado na literatura de *broker*. Esta seleção se deu através da pesquisa de diversos *brokers* e na comparação das seguintes características: possuir versão gratuita, número de tópicos disponíveis (*feeds*), taxa máxima de transações por minuto e facilidade de implementação. O *broker* escolhido é o Adafruit-IO (2021), o qual se destaca por possuir uma versão gratuita com 10 *feeds*, uma taxa máxima de 30 transações por minuto e ampla documentação, atendendo as necessidades do projeto.

3.2 Implementação do Servidor e do Banco de Dados

Conforme citado na Seção 3.1, o *broker* selecionado para este projeto é o AdafruitIO. Para a implementação de um servidor utilizando este *broker* é necessária a criação de uma conta no *site* da empresa, associando um *username* e uma chave para permitir a conexão dos clientes ao servidor. Então são definidos e configurados os *feeds* a serem utilizados pela aplicação, que consistem em: um *feed* para valores de temperatura, dois para valores de umidade do solo, um para valores de luminosidade, um para o estado das válvulas solenoides, um para o estado das luminárias, um para o estado da bomba, dois para o controle das luminárias e um para o controle da irrigação.

Uma vez configurado o servidor, a próxima etapa consiste na implementação da comunicação dos microcontroladores com o mesmo. É importante ressaltar que o *broker* escolhido pelo aluno e suas bibliotecas abstraem a implementação via software dos *headers* das mensagens do protocolo. Por este motivo, nessa etapa apenas são definidos em quais *feeds* cada controlador se inscreve para receber seus valores e em quais *feeds* ele escreve. O controlador principal se inscreve nos *feeds* de controle das luminárias e de irrigação para receber seus valores, escreve os valores medidos pelos sensores em seus respectivos *feeds*, e define os estados (ligado ou desligado) da bomba, das válvulas e das luminárias e enviá-los em seus respectivos *feeds*. O controlador da bomba, por sua vez, se inscreve no *feed* referente ao estado da bomba para receber seu valor atual e poder acioná-la de acordo com o ele.

Além de permitir a comunicação entre os diversos dispositivos do projeto, também é necessário que a solução seja capaz de armazenar os dados relevantes do sistema para posterior análise. Por este motivo, foi implementado um banco de dados através do uso de um automatizador de tarefas, que captura as informações enviadas para o servidor e as insere em uma planilha do *Google Sheets*. O automatizador escolhido para realizar esta atividade é o Zapier (2021), o qual possui um plano inicial que possibilita a criação de 20 automatizações com um tempo de atualização de 15 minutos para cada uma delas, atendendo as necessidades do projeto. Uma vantagem no uso deste automatizador é o fato de ele já possuir uma solução integrada à plataforma *Adafruit IO* e ao *Google Sheets*, tornando o processo de transferir as informações para o banco de dados transparente para os demais componentes de *software* do produto.

3.3 Especificação dos Componentes

Após a definição da arquitetura e do protocolo de comunicação do sistema, o próximo passo para a implementação do projeto consiste na definição dos componentes de *hardware* e hidráulicos a serem utilizados. Este processo pode ser dividido em três etapas: a primeira referente à definição dos componentes do sistema hidráulico, a segunda à parte elétrica e a terceira à eletrônica.

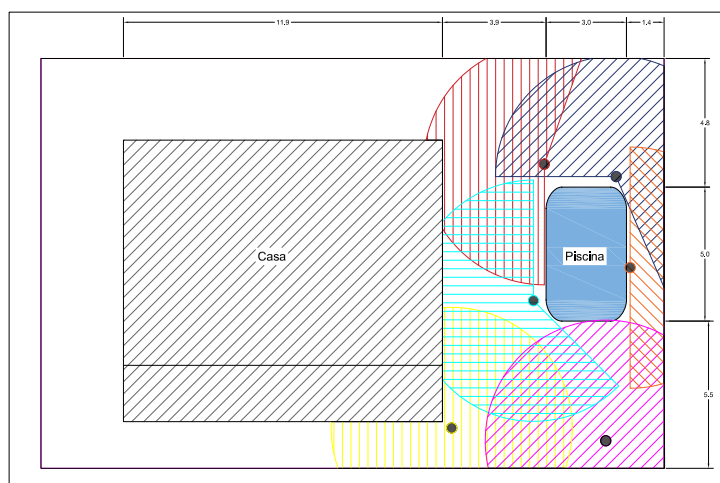
3.3.1 Componentes Hidráulicos

Ao iniciar a especificação desses componentes, é importante ressaltar que a bomba hidráulica utilizada, por já estar instalada, é tratada como uma restrição de projeto. Essa bomba possui as seguintes características: vazão mínima de $0,74 \text{ m}^3/\text{h}$, vazão máxima de $5,9 \text{ m}^3/\text{h}$, pressão máxima sem vazão de 2,9 bar, potência de 1 CV, tensão de operação de 220 V e corrente máxima de 5 A. Além disso, ela já está previamente conectada ao poço artesiano da casa e a um ponto de fornecimento de água que fica localizado próximo ao

setor onde será instalado o sistema.

O primeiro passo para a especificação dos componentes hidráulicos é a definição do modelo e da quantidade de aspersores a serem utilizados. O modelo escolhido para este projeto é o aspersor PSU-06 da marca *Hunter*, que possui um raio de alcance máximo de 5,8 m, uma vazão de 0,04 m³/h à 1,3 m³/h e trabalha com valores de pressão entre 1,4 bar e 4,8 bar. Com base nas especificações do modelo, a definição da quantidade de aspersores a ser utilizada e suas posições são definidas tendo como finalidade garantir que todo o solo seja molhado de forma homogênea. Ela é realizada através da análise e desenho de uma planta no *software* Autocad®, a qual é apresentada na Figura 6, resultando na necessidade de instalar seis aspersores distribuídos de acordo com as posições definidas na figura. É importante ressaltar que a análise levou em conta o fato de a água do poço artesiano utilizado não ser considerada apropriada para consumo humano e, por este motivo, as posições dos aspersores foram definidas de forma a evitar que eles molhem diretamente a piscina.

Figura 6: Localização dos Aspersores no Pátio.



Fonte: Elaborado pelo autor.

A próxima etapa consiste em definir quantos aspersores podem ser acionados ao mesmo tempo de acordo com a capacidade da bomba. Para esta definição, foram realizados testes em loco, os quais levaram a conclusão de que a melhor solução para o projeto é a distribuição dos seis aspersores em duas linhas a serem acionadas isoladamente, cada uma fornecendo água para três deles.

Devido a existência de duas linhas que devem ser controladas de forma independente pelo controlador, torna-se necessário o uso de válvulas hidráulicas com acionamento através de solenoide. Com o objetivo de diminuir o custo do projeto, foram realizados testes com válvulas de máquina de lavar para implementar este controle, porém as mesmas não foram capazes de suprir a vazão necessária para o acionamento dos aspersores. Por este motivo, foi selecionado outro modelo de válvulas que é capaz de trabalhar nas faixas de pressão e vazão suportadas pela bomba. O modelo escolhido consiste nas válvulas solenoides normalmente fechadas modelo 100-HV da empresa *Rainbird*, as quais possuem uma pressão de operação que varia de 1 bar à 10,3 bar, uma vazão de operação que varia entre 0,05 m³/h e 9,1 m³/h, e solenoide acionado com 24 V AC e com uma corrente de ativação de 0,25 A.

3.3.2 Componentes Elétricos

Os componentes elétricos do sistema consistem em dois interruptores paralelos que tem a finalidade de permitir o uso das luminárias quando o sistema estiver desligado ou sem acesso a internet, três lâmpadas LED verdes utilizadas para o paisagismo, e seis luminárias do tipo tartaruga com lâmpadas amarelas usadas para a iluminação ambiente do pátio. A solução também conta com três mini disjuntores de 10 A, sendo um deles utilizado na entrada de energia do controlador geral, que é alimentado com 127 V (uma fase conectada ao disjuntor e um neutro), e os outros dois na entrada de energia do controlador da bomba, que é alimentado com 220 V (duas fases, cada uma conectada a um disjuntor).

3.3.3 Componentes de *Hardware*

Os componentes de *hardware* presentes na solução consistem nos eletrônicos a serem inseridos nas placas dos controladores, em um transformador e nos sensores a serem utilizados. O primeiro item a ser especificado é o microcontrolador, então é escolhido o conversor CC/CA para permitir a alimentação deste microcontrolador. Em seguida são selecionados os relés responsáveis por permitir que o microcontrolador realize os controles das válvulas e lâmpadas. Então é especificado o transformador que converte a tensão da rede no nível necessário para o acionamento das válvulas, e, por fim, os sensores.

3.3.3.1 Especificação do Microcontrolador

A definição do modelo de microcontrolador a ser utilizado no projeto é realizada através da comparação entre três modelos de baixo custo que são comumente utilizados em soluções IoT, comparação a qual é apresentada na Tabela 1. O primeiro parâmetro a ser levado em consideração neste processo de decisão é a necessidade de conexão com a internet via Wi-Fi, o que torna desejável que o microcontrolador já possua um módulo Wi-Fi incluso em sua solução, evitando a necessidade de comprar e integrar mais um componente. Outro fator importante é o número de portas IO, que está diretamente relacionado à expansividade do sistema. Para atender as necessidades do projeto, o microcontrolador deve possuir, pelo menos, 8 portas IO (4 para leitura dos sensores e 4 para acionamento dos relés), sendo possível utilizar 3 delas para os sensores analógicos selecionados na Seção 3.3.3.5. Além disso, também devem ser levados em conta o custo associado a este microcontrolador e a facilidade de integrá-lo ao servidor MQTT selecionado na Seção 3.1.

Tabela 1: Comparação Entre Modelos de Microcontroladores.

Módulo	Controlador	Wi-Fi	Portas I/O	Portas Analógicas	Biblioteca Adafruit IO	Preço (R\$)
Arduino Uno	ATMega328P	Não	14	6	Não	40,00
NodeMCU 8266	Esp8266	Sim	13	1	Não	47,40
NodeMCU 32	Esp32	Sim	36	18	Sim	54,15

Fonte: Arduino-Uno (2021), NodeMCU-8266 (2021) e NodeMCU-32S (2021).

De acordo com os dados levantados pode-se observar que o Microcontrolador Esp8266 deve ser descartado por não possuir o número de portas analógicas necessárias. Além disso, conclui-se que a melhor alternativa é o Microcontrolador Esp32, o qual atende todos os requisitos do projeto, possuindo módulo Wi-Fi, um maior número de portas e uma biblioteca padrão fornecida pela empresa Adafruit-IO (2021) para a comunicação com seu servidor.

O modelo escolhido conta também com interfaces SPI, UART e I2C, 2 portas com conversor digital analógico, memória flash de 16 Mb, memória RAM de 512 KB e conexão *bluetooth*, o que torna possível a expansão do sistema e sua integração com outros equipamentos em projetos futuros. Sua tensão de alimentação é de 5 V CC, com um consumo de corrente de 260 mA, portas de IO de 3,3 V CC e é compatível com o ambiente de desenvolvimento do Arduino®.

3.3.3.2 Especificação do Conversor CA/CC

O processo de escolha deste componente tem como principal objetivo utilizar o mesmo modelo de conversor para o controlador principal e o da bomba, o que resulta na necessidade de ele suportar tensões de alimentação de 127 V AC (controlador principal) e 220 V AC (controlador da bomba). Outro requisito relevante para o componente é a necessidade de ele possuir uma tensão de saída de 5 V CC, permitindo a alimentação do modelo de microcontrolador utilizado, cujas características são apresentadas na Seção 3.3.3.1. Também é importante que o conversor seja capaz de fornecer a corrente necessária para alimentar o microcontrolador (260 mA), os relés (20 mA), os sensores de umidade (2 sensores consumindo 5 mA cada), o sensor de temperatura (2,5 mA) e o sensor de luminosidade (15 mA) utilizados na solução, tornando necessária uma capacidade de corrente de, pelo menos, 307,5 mA. Uma vez definidos os requisitos a serem atendidos pelo componente, o próximo fator de escolha é o menor custo.

Após uma pesquisa de mercado por componentes de baixo custo que atendam os requisitos levantados, o modelo de conversor selecionado consiste no HLK-PM01 (2021), que suporta uma tensão de alimentação de 100 V AC à 240 V AC, possui uma tensão de saída de $5 \pm 0,2$ V CC e uma capacidade de corrente de 600 mA.

3.3.3.3 Especificação do Módulo de Relés

A seleção dos relés tem como principal fator de decisão a modularidade, o que facilita a montagem do sistema e permite o uso de um mesmo módulo para acionar diversos atuadores diferentes, tornando a solução expansível. Levando em conta esta necessidade e o fato de o controlador principal ser responsável pelo acionamento de quatro atuadores diferentes, é utilizado um módulo de quatro relés para ambos os controladores.

Levando em conta o fator mencionado e as características das cargas a serem acionadas, os relés presentes neste módulo devem ser capazes de fornecer tensões de 127 V AC (luminárias), 220 V AC (bomba) e 24 V CC (solenoides das válvulas hidráulicas). Eles também precisam ter uma capacidade de corrente de, pelo menos, 5 A (corrente máxima consumida pela bomba) e uma tensão de alimentação e de ativação de 3,3 V CC para torná-los compatíveis com o microcontrolador escolhido. Além disso, para garantir o isolamento entre o circuito de controle e o de potência, o módulo de relés deve contar com opto-acopladores.

O modelo escolhido é o MOD.RELE.3V.4CH. (2021), o qual atende os requisitos de projeto, possui uma capacidade de corrente de 10 A e um consumo de 20 mA do microcontrolador durante sua operação.

3.3.3.4 Especificação do Transformador

O transformador tem como objetivo converter a tensão de entrada do controlador principal (127 V AC) para a tensão de acionamento das válvulas hidráulicas (24 V AC). Por este motivo, além de ser capaz de realizar a conversão citada, o modelo precisa possuir

uma capacidade de corrente suficiente para alimentar as duas válvulas, ou seja, 0,5A. O transformador com melhor custo-benefício encontrado no mercado possui uma capacidade de corrente de 2 A e atende os demais requisitos.

3.3.3.5 Especificação dos Sensores

Conforme observado na Seção 3, o presente projeto conta com dois sensores de umidade, um sensor de temperatura e um sensor de luminosidade. A primeira característica a ser observada na escolha destes componentes é que, em virtude do modelo de microcontrolador utilizado, todos os sensores devem ser capazes de operar com uma tensão de alimentação de 3,3 V CC. Além disso, devido ao fato da temperatura não ser utilizada por nenhum sistema de controle e a luminosidade ser utilizada apenas para determinar se é dia ou noite, precisão não é um fator relevante para a escolha destes sensores, tornando o custo o principal fator a ser levado em consideração para sua seleção. Por este motivo, o processo de seleção desses dois sensores foi realizado através da comparação dos custos de diversos sensores existentes no mercado que atendessem os requisitos mínimos citados.

O modelo escolhido para o sensor de temperatura consiste no DHT11 (2021), o qual atende os requisitos de projeto, possuindo uma faixa de medição de temperatura de 0° C à 50° C, uma repetibilidade de $\pm 0,2^\circ$ C, precisão de $\pm 2^\circ$ C, um consumo de corrente de 2,5 mA e comunicação serial. O sensor de luminosidade selecionado é o módulo LDR (2021), que atende as necessidades do projeto, possuindo um consumo de corrente de 15 mA e uma interface analógica para envio de suas leituras para o microcontrolador.

Para a especificação dos sensores de umidade é necessária a comparação entre os dois tipos de sensores que atendem as necessidades de projeto e possuem baixo custo, os quais consistem nos sensores capacitivos e resistivos. A principal diferença entre eles é a forma como eles realizam a medição da umidade: o resistivo possui dois eletrodos expostos e realiza a medição através da relação da resistência do solo entre estes eletrodos com a umidade presente no solo, enquanto o capacitivo utiliza a relação entre a capacitância do solo e sua umidade. Conforme observado por diversos fabricantes, como a empresa Nicawe (2021), sensores de umidade capacitivos estão menos sujeitos a corrosão e, conseqüentemente, têm uma vida útil maior, motivo que leva a escolha por este tipo de sensor para o presente projeto. Assim como para a seleção dos demais sensores, a escolha do modelo do sensor de umidade capacitivo se deu através da comparação dos custos de sensores existentes no mercado que atendessem os requisitos do projeto citados nesta seção. O modelo selecionado consiste no sensor SEN0193 (2021), o qual possui um consumo de corrente de 5 mA e uma interface analógica para envio de suas leituras para o microcontrolador, retornando valores proporcionais a percentagem de volume de água presente no solo.

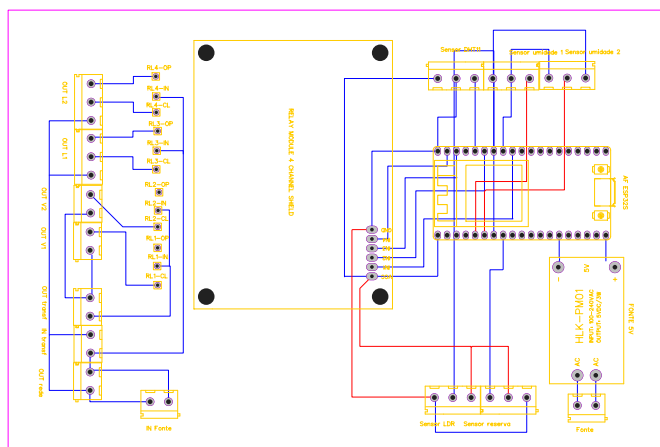
3.4 Diagramas Elétricos e Montagem das Placas dos Controladores

Após a seleção dos componentes a serem utilizados, a próxima etapa do projeto consiste na montagem das placas do microcontroladores, tendo como objetivo tornar o *hardware* do sistema de controle robusto para permitir a instalação do mesmo no pátio. Este processo inicia com a implementação do diagrama de ligação entre os componentes, seguido da definição de sua localização na placa e das trilhas de ligação entre eles através de seu desenho na ferramenta EasyEDA (2021), e terminando com a montagem da placa em si.

Apesar de o sistema possuir dois controladores (controlador principal e controlador da bomba), o projeto das placas é realizado de forma modular, tornando necessária apenas a

remoção dos bornes de conexão dos sensores ao montar a placa do controlador da bomba. Por este motivo será apresentado apenas o projeto e a montagem da placa do controlador principal, cujo diagrama elétrico segue no Apêndice A.4 e desenho de sua placa e as trilhas que ligam seus componentes segue na Figura 7. Pode-se observar no diagrama e no desenho a presença do microcontrolador, do conversor CA/CC e do módulo de relés, e de bornes para as seguintes conexões: alimentação da placa, sensores de umidade, sensor de luminosidade, sensor de temperatura e acionamento das duas válvulas e das duas linhas de luminárias.

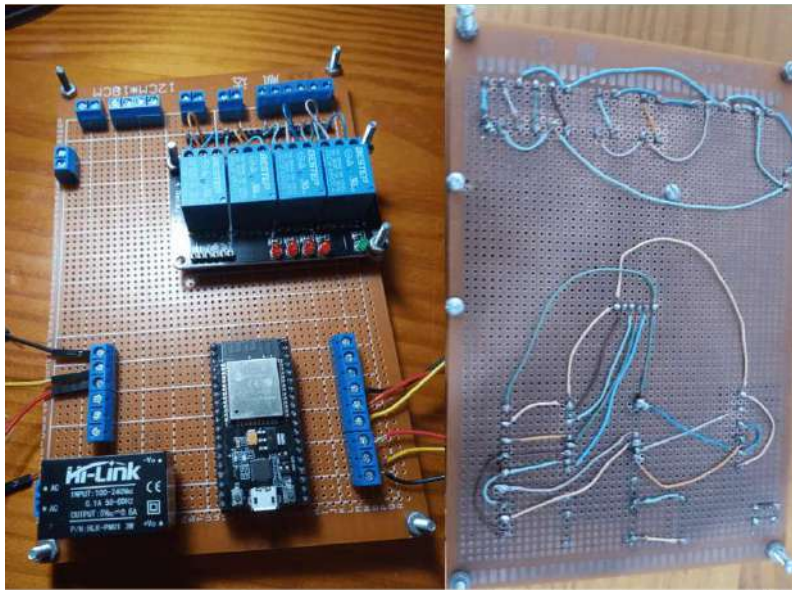
Figura 7: Projeto da Placa do Controlador Principal.



Fonte: Elaborado pelo autor.

Uma vez finalizado o projeto da placa, a próxima etapa consiste na montagem da mesma. Ela é montada tendo como base uma placa fenolite perfurada de 12 cm x 18 cm, onde são soldados os componentes e as trilhas de conexão entre eles são realizadas através de cabos sólidos soldados manualmente. Uma vez inseridos os componentes e soldadas as trilhas, a proteção da placa contra intempéries é realizada através da aplicação de silicone em sua superfície inferior. A Figura 8, apresenta a placa do controlador principal montada e o resultado do processo de soldagem das trilhas da placa. Após sua montagem, são inseridas etiquetas na placa para identificar quais equipamentos devem ser conectados aos bornes e em que ponto do borne deve ser conectado o terra, a alimentação e o sinal de leitura dos sensores e demais equipamentos.

Figura 8: Placa do Controlador Principal Montada.



Fonte: Elaborado pelo autor.

3.5 Softwares dos Microcontroladores

Após a montagem das placas do controlador principal e do controlador da bomba, a próxima etapa do projeto consiste na implementação dos *softwares* para os microcontroladores dessas placas. Neste caso é implementado um *software* para cada microcontrolador. Os códigos fontes do projeto são apresentados no Apêndice B.

3.5.1 Software do Controlador Principal

Este *software* é composto basicamente por quatro funções. A primeira consiste na troca de informações com o servidor, cuja implementação segue na Seção 3.1. A segunda consiste na leitura dos sensores presentes no sistema, a terceira no controle de irrigação e a quarta no controle dos dois setores de iluminação.

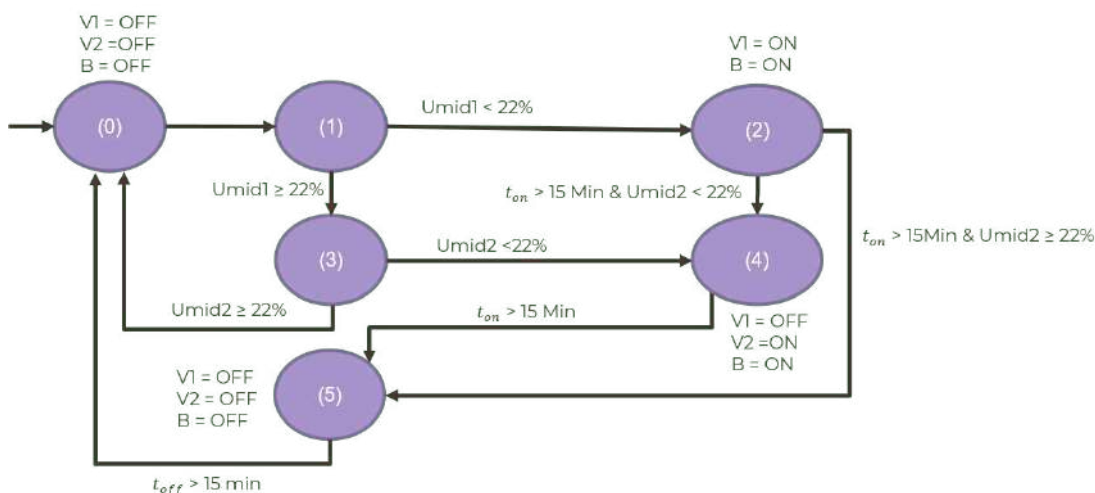
A função referente a leitura dos sensores consiste no uso de um conversor analógico digital presente no microcontrolador para a leitura dos sensores de umidade e luminosidade, no uso de uma biblioteca padrão para a comunicação via RS232 com o sensor de temperatura e no envio dos valores lidos pelos sensores para o servidor. Devido as limitações do *broker* escolhido já apresentadas na Seção 3.1, é necessária a escolha de maiores tempos de amostragem para os sensores. Levando em conta essas limitações e as características do sistema de interesse foi definido que a leitura dos valores de umidade seja realizada a cada 3 minutos, dos de luminosidade a cada 4 minutos e dos de temperatura a cada 5 minutos.

A funcionalidade do controle de irrigação é implementada em duas camadas. A primeira consiste em uma função responsável por selecionar o modo de operação do controlador de acordo com a entrada setada pelo usuário, possuindo as seguintes possibilidades: desligado, onde as válvulas e a bomba ficam desligadas; irrigar o setor 1, onde a bomba e a válvula do setor 1 são acionadas; irrigar o setor 2, onde a bomba e a válvula do setor 2 são acionadas; modo automático ligado apenas de dia, onde o modo automático do sistema de irrigação só é ativado quando o sensor de luminosidade indicar que é dia; modo automático ligado apenas a noite, onde o modo automático do sistema de irrigação só é ativado quando o sensor de luminosidade indicar que é noite; e modo automático, onde o

modo automático do sistema de irrigação é ativado sem levar em conta se é dia ou noite.

A segunda camada do controle de irrigação consiste na função que controla seu modo automático, a qual é ativada quando a primeira camada desta funcionalidade liga o modo automático do controle de irrigação. Esta função consiste em uma máquina de estados, que possui 6 estados e é apresentada na Figura 9, onde Umid1 e Umid2 são os valores de umidade lidos pelos sensores, V1 e V2 são os valores aplicados nos relés que controlam as válvulas solenoides da bomba de irrigação e B é o valor do comando a ser enviado para o controlador da bomba.

Figura 9: Modo Automático do Controle de Irrigação.



Fonte: Elaborado pelo autor.

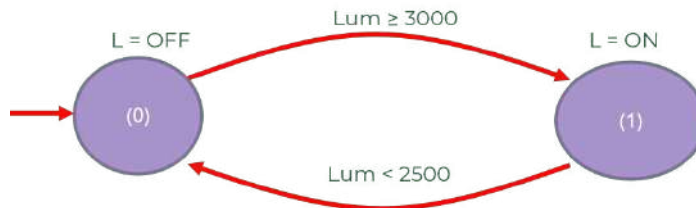
Quando o sistema de irrigação entra em modo automático ele inicia no estado 0, onde ambas as válvulas e a bomba estão desligadas. Então ele segue para o estado 1, onde as válvulas e a bomba continuam desligadas e o valor lido pelo sensor de umidade do primeiro setor é comparado com um *setpoint* de 22%, caso o valor lido seja maior ou igual a ele o sistema segue para o estado 3, caso contrário ele segue para o estado 2. Uma vez no estado 2, a válvula do setor 1 e a bomba são acionadas e o sistema aguarda 15 minutos, então o valor de umidade lido pelo sensor do setor 2 é comparado com um *setpoint* de 22%, caso o valor lido seja maior ou igual a ele o sistema segue para o estado 5, caso contrário ele segue para o estado 4. Uma vez no estado 3, o valor de umidade lido pelo sensor do setor 2 é comparado com um *setpoint* de 22%, caso o valor lido seja maior ou igual a ele o sistema retorna para o estado 0, caso contrário ele segue para o estado 4. Uma vez no estado 4, a válvula do setor 2 e a bomba são acionadas e a válvula do setor 1 é desligada, então o sistema aguarda 15 minutos e segue para o estado 5. Uma vez no estado 5, ambas as válvulas e a bomba são desligadas, o sistema aguarda 15 minutos para permitir que o solo absorva a água fornecida e retorna para o estado zero.

É importante ressaltar que a transição entre os estados da máquina na Figura 9 ocorre a cada 3 minutos, tempo equivalente a uma leitura dos sensores de umidade. Além disso, os valores utilizados no *setpoint* de umidade e nos tempos t_{on} e t_{off} são justificados na Seção 3.9.

O controle dos dois setores de iluminação ocorre de forma independente, porém seu comportamento e suas funções são as mesmas, tornando possível a implementação de duas funcionalidades idênticas para realizar este controle. Assim como o controle de irrigação, essa funcionalidade também pode ser dividida em duas camadas. A primeira delas é

responsável por seleccionar o modo de operação do controlador de acordo com a entrada seleccionada pelo usuário, possuindo as seguintes possibilidades: iluminação desligada, iluminação ligada e modo automático. A segunda camada é responsável pelo controle do modo automático e é ativada quando a camada 1 de seu respectivo setor de iluminação está neste modo. Essa função consiste em uma máquina de estados, que possui 2 estados e é apresentada na Figura 10, onde Luminosidade são os valores de luminosidade lidos pelo sensor e L é o estado do relé que aciona as luminárias do setor.

Figura 10: Modo Automático do Controle de Luminosidade.



Fonte: Elaborado pelo autor.

Quando o sistema de controle de luminosidade de uma das linhas de luminárias entra em modo automático ele inicia no estado 0, onde o relé que aciona as luminárias é desligado e o valor lido pelo sensor de luminosidade é comparado com um *setpoint*, caso ele seja maior que este valor o sistema segue para o estado 1, caso contrário ele se mantém no mesmo estado. Uma vez no estado 1, o relé que aciona as luminárias é acionado e o valor lido pelo sensor de luminosidade é comparado com outro *setpoint*, caso ele seja menor que este valor o sistema retorna para o estado 0, caso contrário se mantém no estado atual.

É importante ressaltar que a transição entre os estados da máquina presente na Figura 10 ocorre a cada 5 minutos, tempo equivalente a uma leitura dos sensores de luminosidade. Além disso, os *setpoints* de luminosidade são justificados na Seção 3.9.

3.5.2 Software do Controlador da Bomba

Conforme observado na Seção 3, o controlador da bomba atua como um escravo do controlador principal, recebendo o comando de acionamento da bomba produzido pelo controlador principal e ativando seu respectivo relé de acordo com o valor recebido. A implementação de seu *software* consiste na comunicação do mesmo com o servidor conforme descrito na Seção 3.1, na inscrição para a leitura do *feed* referente ao controle da bomba e no acionamento de seu respectivo relé de acordo com o valor atual deste *feed*. Devido ao fato de a versão gratuita do *broker* escolhido não possuir garantia de entrega dos *feeds*, além do *subscribe* no *feed* do valor do acionamento da bomba, é inserida uma redundância neste *software* através da leitura deste campo a cada 1 minuto.

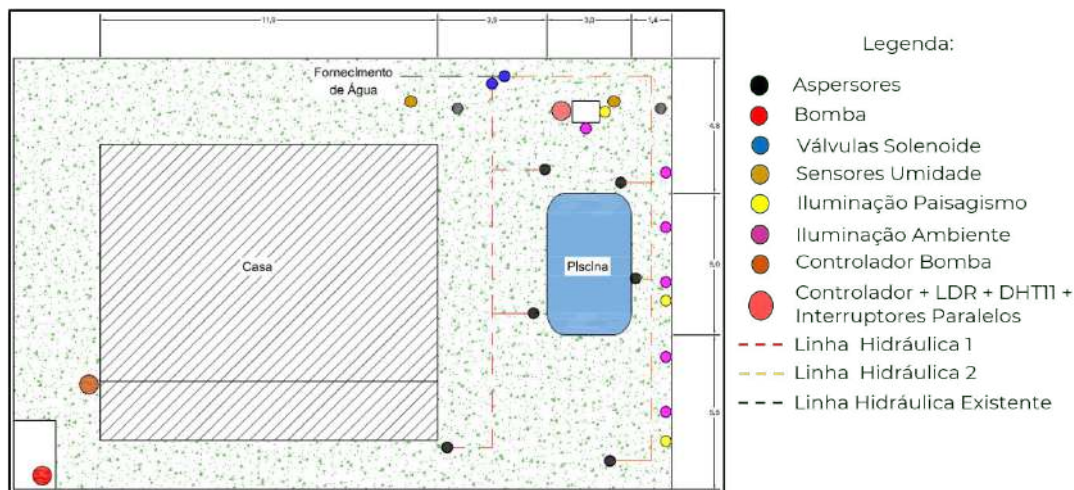
É importante ressaltar que a bomba já instalada no pátio não está preparada para operar a vazio por longos períodos de tempo. Por este motivo, seu *software* de controle possui um mecanismo de segurança que desliga a bomba caso seu microcontrolador perca conexão com a internet.

3.6 Localização dos Equipamentos no Pátio

Uma vez implementados os *softwares* e montadas as placas dos microcontroladores, a próxima etapa do projeto consiste na definição do local onde eles e os demais equipamentos usados na solução devem ser instalados, a qual é apresentada na Figura 11. Esta definição

é realizada tomando como base a posição da bomba já instalada no pátio, o setor escolhido para a implementação e a posição dos aspersores definida na Seção 3.3.1. É importante ressaltar que o painel de controle da bomba fica localizado mais próximo a casa e distante da bomba devido a necessidade de acesso ao Wi-Fi para sua conexão ao servidor.

Figura 11: Mapa com Localização dos Equipamentos.



Fonte: Elaborado pelo autor.

3.7 Instalação dos Equipamentos no Pátio

Uma vez realizada a programação e montagem dos controladores e definida a localização dos equipamentos no pátio, a próxima etapa do projeto consiste na instalação deles. Ela pode ser dividida na instalação do painel do controlador principal e seus sensores, na instalação do sistema de iluminação, na instalação do painel do controlador da bomba e seu sistema de acionamento e na instalação do sistema hidráulico.

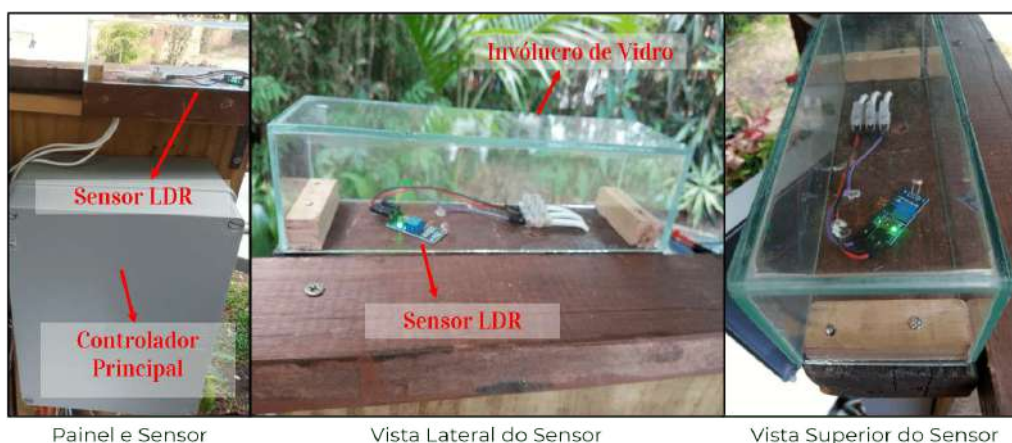
3.7.1 Painel Principal e seus Sensores

A instalação dos equipamentos tratados nesta seção inicia com a fixação de um painel de 30 cm x 22 cm x 12 cm ao lado de um painel para controle de piscina já existente no pátio da casa. Após a fixação do painel, é instalada em seu interior a placa do controlador principal, etapa que é seguida pela conexão do transformador e do sensor de temperatura em seus respectivos bornes e sua fixação no interior do painel. Então é realizada a instalação de um disjuntor de 10 A no painel já existente, o qual é conectado à rede e ao painel do controlador principal para alimentá-lo com 127 V. A Figura 12 apresenta uma vista interna e uma vista externa do painel instalado, onde o disjuntor de entrada instalado no painel já existente está destacado por um círculo vermelho.

Figura 12: Instalação Painel Principal.

Fonte: Elaborado pelo autor.

Após a montagem do painel, o próximo componente a ser instalado consiste no sensor de luminosidade, o qual é fixado sobre uma plataforma de madeira montada acima do painel do controlador principal. Uma vez fixado, o sensor é conectado ao seu respectivo borne no painel do controlador principal e é envolvido por um invólucro de vidro transparente para garantir sua proteção contra intempéries sem prejudicar as leituras de luminosidade. A Figura 13 apresenta a localização do sensor em relação ao painel, uma vista lateral e outra superior do sensor instalado.

Figura 13: Instalação Sensor Luminosidade.

Fonte: Elaborado pelo autor.

Por fim, é realizada a instalação dos sensores de umidade. Devido a estes sensores estarem localizados a uma maior distância do painel, são utilizados cabos blindados tipo manga de 3 vias AWG 26 para minimizar o efeito do ruído. É importante ressaltar que, neste caso, a blindagem do cabo deve ser aterrada junto ao terra (GND) do controlador e não deve ser conectada ao GND do sensor. Além disso, para garantir a proteção destes cabos contra a umidade, os mesmos são envolvidos por uma mangueira antes de serem enterrados no pátio.

Uma vez conectados os cabos aos sensores de umidade e aos seus respectivos bornes no painel do controlador principal, o próximo passo é a proteção dos sensores para a sua

instalação no pátio. Nesta etapa são fixadas caixas de plástico e é inserido silicone sobre a região onde estão localizados os eletrônicos do sensor. Então, é montada uma caixa cilíndrica de proteção na parte superior do sensor para evitar o contato com umidade em sua conexão com os cabos. A Figura 14 apresenta a forma como este sensor é protegido e sua instalação no pátio.

Figura 14: *Instalação Sensor Umidade.*



Fonte: Elaborado pelo autor.

3.7.2 Sistema de Iluminação

Após a montagem do painel do controlador principal, pode-se prosseguir para a instalação do sistema de iluminação, a qual inicia com a fixação dos interruptores paralelos e sua conexão com os respectivos bornes do painel. Então são montadas as luminárias do sistema de iluminação ambiente e de paisagismo e as mesmas são conectadas aos seus respectivos interruptores utilizando cabos plastichumbo de 1,5 mm enterrados no jardim. As figuras 15 e 16 mostram o sistema instalado.

Figura 15: *Instalação Iluminação vista 1.*



Fonte: Elaborado pelo autor.

Figura 16: Instalação Iluminação vista 2.

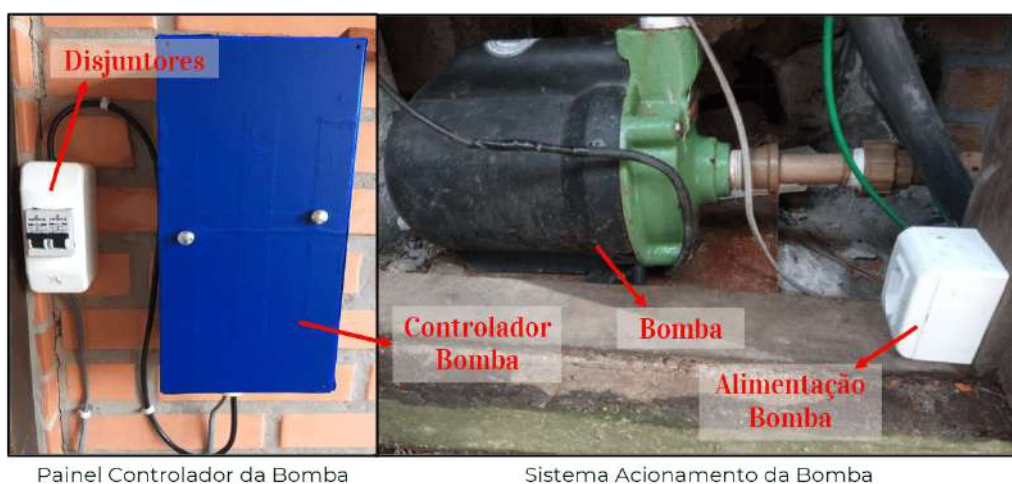


Fonte: Elaborado pelo autor.

3.7.3 Painel do Controlador da Bomba e Sistema de Acionamento

Outra etapa importante na implantação do sistema é a montagem do painel de controle da bomba e do sistema de acionamento a ser utilizado. Este processo inicia com a fixação de um painel de 30cm x 18cm x 15cm na parede da casa de acordo com a localização estipulada na Seção 3.6 e na instalação da placa do controlador dentro do painel. Então dois disjuntores de 10 A são inseridos dentro de uma caixa para disjuntores, fixados na parede da casa e conectados as duas fases da rede e aos bornes de alimentação do painel para fornecer uma tensão de 220 V a ele. Por fim, é instalada uma tomada na estrutura onde a bomba está localizada e a mesma é conectada através de cabo plasticumbo de 2,5 mm aos respectivos bornes do painel. A Figura 17 mostra a instalação do painel, dos disjuntores e da tomada da bomba.

Figura 17: Instalação Sistema e Painel de Acionamento da Bomba



Painel Controlador da Bomba

Sistema Acionamento da Bomba

Fonte: Elaborado pelo autor.

3.7.4 Sistema Hidráulico

A última parte do produto a ser montada é o sistema hidráulico, o qual consiste no conjunto de 6 aspersores, divididos em duas linhas, cada uma com uma válvula solenoide. O sistema foi instalado utilizando tubos PVC de 3/4 de polegada para as linhas e mangueiras com o mesmo diâmetro para realizar as conexões com as válvulas e aspersores devido à necessidade de maior flexibilidade.

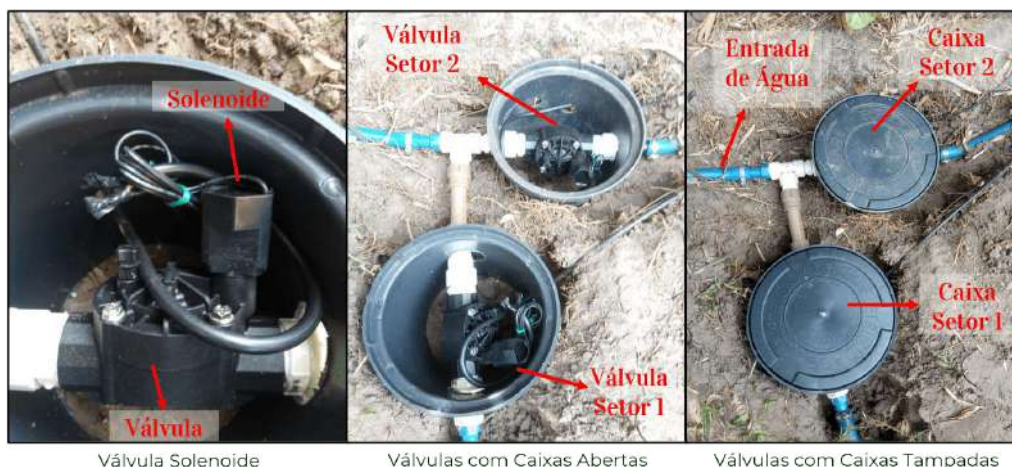
Sua instalação inicia com a montagem das duas linhas principais e suas terminações utilizando os tubos PVC. Ela segue com montagem dos aspersores e sua conexão com as linhas principais, a qual se dá através de linhas secundárias, com diâmetro de 1/2 polegada conectadas a 90°, através da introdução de um “Tê” ao longo da linha principal. No final de cada linha secundária é inserida uma conexão flexível que faz sua ligação com a base do aspersor. Estes aspersores, por sua vez, são fixados ao solo verticalmente com a parte móvel, escamoteável, nivelada com a superfície do terreno. A Figura 18 apresenta a instalação de um aspersor no solo.

Figura 18: *Instalação Aspersores*



Fonte: Elaborado pelo autor.

O processo termina com a instalação das válvulas, a qual é apresentada na Figura 19, sua ligação com as linhas principais e com o ponto de fornecimento de água através de mangueiras flexíveis, e a conexão do acionamento de seus solenoides aos respectivos bornes de saída do painel do controlador principal através de cabo plastichumbo 1,5 mm enterrado.

Figura 19: Instalação Válvulas Solenoides

Fonte: Elaborado pelo autor.

3.8 Calibração do Sensor de Umidade

Após a implantação do sistema, o próximo passo do projeto consiste na calibração de seus sensores e a conversão dos valores lidos por eles para unidades com sentido físico, etapas que são necessárias apenas para os sensores de umidade, uma vez que os sensores de temperatura já retornam valores medidos em graus Célsius e não são utilizados por nenhum sistema de controle e os sensores de luminosidade são usados apenas para avaliar se é dia ou noite.

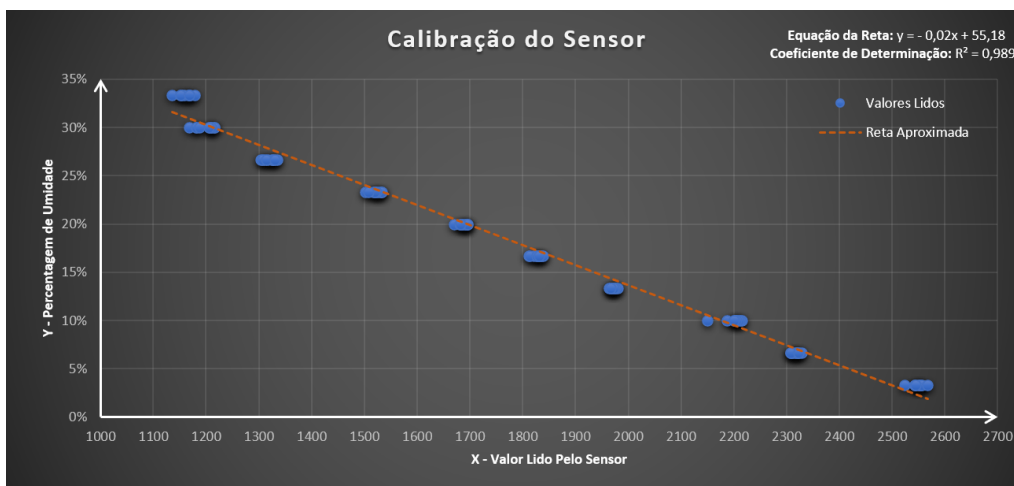
Conforme mencionado na Seção 3.3.3.5, os sensores de umidade retornam valores proporcionais a percentagem de água presente no solo. Para permitir o uso destas leituras pelo sistema de controle e a interpretação dos mesmos pelo usuário, é necessária a determinação de uma equação que permita a conversão dos valores lidos pelo conversor analógico digital do controlador para percentagem de umidade e a definição das faixas de valores que correspondem ao solo seco, solo com umidade ideal e solo saturado. Esta determinação consiste no processo de calibração do sensor e é realizada através de um ensaio com os sensores de umidade.

O ensaio inicia com a extração de 750 cm³ de solo do pátio, o qual é aquecido até estar seco, inserido em um recipiente e depois resfriado até atingir a temperatura ambiente. Na sequência são instalados os sensores de umidade neste solo e conectados ao microcontrolador, para realizar a medição da umidade de solo nesta condição. O resultado obtido neste ponto é considerado como o valor de 0% de umidade no solo. A partir desse estado inicial, são inseridos volumes fixos de 25 ml de água e são coletados 10 valores medidos pelos sensores a cada inserção até o momento em que o solo passa pelo valor de umidade ideal e atinge a saturação. O anexo A.1 apresenta figuras com as etapas do ensaio e os materiais utilizados.

Em posse do volume de solo utilizado no ensaio, do volume de água inserido em cada iteração e dos valores medidos pelos sensores, é possível traçar um gráfico que relacione a percentagem de umidade no solo e os valores obtidos pelos sensores, o qual é apresentado na Figura 20. A partir da figura observa-se que, apesar de o comportamento do sensor não ser linear, o mesmo pode ser aproximado por uma reta para soluções que não exijam alta precisão, solução utilizada no presente trabalho por tratar-se de um projeto de irrigação residencial. A equação desta reta é determinada através de uma regressão linear obtida no

software Microsoft Excel®, resultando na relação entre a medida do sensor e a percentagem de umidade $Umid = -0,02 \cdot ValLido + 55,18$ [%] e apresentando um coeficiente de determinação de $R^2 = 0,9894$, o que comprova que a equação obtida representa de forma satisfatória o comportamento do sensor.

Figura 20: Ensaio Calibração Sensor de Umidade: Dados Obtidos



Fonte: Elaborado pelo autor.

Durante a realização do ensaio, é utilizada a experiência do aluno para determinar empiricamente quais faixas de valores de percentagem de umidade correspondem aos valores ideal, seco e saturado. As faixas encontradas são: solo seco para percentagens de umidade abaixo de 20%, umidade ideal para percentagens entre 20% e 27% e solo saturado para percentagens de umidade acima de 27%. Após a obtenção dos valores, os mesmos são comparados com os encontrados durante a pesquisa teórica apresentada na Seção 2.2, concluindo-se que as faixas utilizadas pelo aluno estão próximas do valor teórico de umidade ideal.

3.9 Parâmetros para Controle dos Modos Automáticos

Para realizar a implementação das máquinas de estados que controlam os modos automáticos dos controles de irrigação e de luminosidade, é necessária a definição de alguns parâmetros, sendo eles: os *setpoints* de luminosidade, o *setpoint* de umidade, o tempo que o sistema de irrigação deve ficar ligado (t_{on}) e o tempo que o sistema de irrigação aguarda após molhar o solo antes de retornar ao estado inicial (t_{off}).

O *setpoint* de luminosidade que determina quando o sistema é ligado é definido tendo como base os valores lidos pelos sensores durante um período de 24 horas, onde é selecionado o valor para o qual é considerado noite. Os dados obtidos neste ensaio são apresentados no Apêndice A.3. Para evitar que pequenas oscilações nos dados obtidos pelo sensor afetem o comportamento do sistema, é inserida uma histerese no mesmo, resultando nas seguintes faixas de operação: para valores de luminosidade abaixo de 2500 as luminárias são desligadas, para valores entre 2500 e 3000 as luminárias se mantêm em seu estado anterior e para valores acima de 3000 as luminárias são ligadas. É importante ressaltar que os valores de luminosidade citados são adimensionais, pois os mesmos se tratam dos valores obtidos a partir do sensor de luminosidade sem nenhuma conversão e são utilizados apenas para determinar as faixas de operação do sistema, não sendo necessária

sua interpretação por parte do usuário.

O *setpoint* de umidade tem como objetivo garantir que o solo não alcance valores de umidade abaixo da faixa de umidade ideal estabelecida na Seção 3.8. Por este motivo o valor selecionado para este parâmetro é uma umidade de 22%. Para manter os valores de umidade dentro dessa faixa ideal e evitar que o sistema acione diversas vezes ao dia, é necessária a determinação da variação de umidade que os aspersores devem aplicar a cada acionamento do sistema, valor que é definido como 3%. A partir desta variação desejada é possível obter a quantidade de água que deve ser fornecida pelos sensores a cada acionamento e, conseqüentemente, o tempo que os mesmos devem ficar ligados para causar essa variação. A quantidade de água é obtida através da equação $V_{nec} = var \cdot h \cdot \pi \cdot r^2 = 14,9 [m^3]$, que relaciona esta quantidade com o raio de alcance do aspersor ($r = 5$ m), a profundidade de instalação do sensor ($h = 0,1$ m) e a variação de umidade desejada ($var = 3\% = 0,03$). O tempo (t_{on}) é obtido através da relação entre esta quantidade de água (V_{nec}) e a quantidade de água fornecida pelo aspersor a cada minuto (V_{asp}), cujo resultado é apresentado na equação $t_{on} = V_{nec}/V_{asp} = 14,9 [min]$. Com base no valor obtido para t_{on} , o tempo que os aspersores ficam ligados é definido como $t_{on} = 15$ minutos.

O último parâmetro a ser definido é o tempo que a máquina de estados aguarda após irrigar algum setor para retornar ao estado inicial, cuja finalidade é garantir que a água fornecida pelos aspersores seja absorvida pelo solo e a dinâmica dos sensores perceba esta variação de umidade antes de o sistema voltar a avaliar os valores lidos pelo sensor. Este tempo é definido a partir de um ensaio de infiltração realizado no pátio, o qual inicia com a execução de uma cava de dimensões conhecidas no solo e com a introdução de um anel metálico para garantir que a absorção de água ocorra apenas pelo fundo da cava. Então é fixada uma régua com marcações para medir o nível de água neste buraco e é inserido um volume de 21 litros de água no mesmo, o que é equivalente a um nível de 30 cm. O Apêndice A.2 apresenta os passos para realização desta etapa inicial do ensaio.

Após a inserção de água, é utilizado um cronômetro para observar quanto tempo o solo leva para absorver um volume de água suficiente para seu nível baixar um centímetro, processo que é repetido até não haver mais água no buraco. A partir da média dos tempos obtidos neste ensaio, é auferido o tempo que o solo leva para absorver água o suficiente para baixar o nível em 1 cm, o qual corresponde à 1 minuto. Em posse desse valor é possível obter, através da equação $t_{off} = t_{abs} \cdot h = 15 [min]$, o tempo necessário para o solo absorver água o suficiente para descer seu nível em um valor equivalente a altura de instalação do sensor ($h = 15$ cm). Conforme observado na equação, o valor encontrado para este parâmetro consiste em $t_{off} = 15$ minutos.

Por fim, é importante ressaltar que os parâmetros obtidos nesta seção são estimativas iniciais e seus valores serão validados e ajustados de acordo com os resultados levantados na Seção 4.

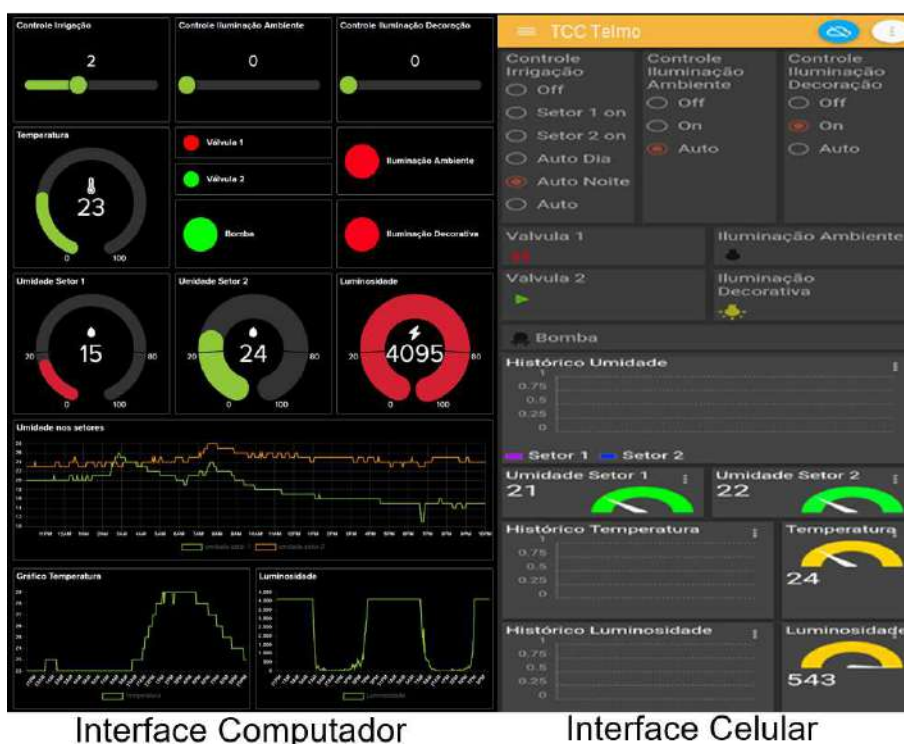
3.10 Interfaces para Computador e Celular

Após a instalação e integração do sistema e seus componentes, a última etapa no desenvolvimento deste projeto consiste na implementação das interfaces para computador e celular através das quais o usuário irá interagir com o produto. Ambas as interfaces são construídas por intermédio do uso de *dashboards* fornecidos por soluções de mercado. Por este motivo, a implementação delas consiste na configuração dessas ferramentas para sua comunicação com o servidor e a definição e configuração de cada *dashboard* a ser inserido nelas.

A configuração da comunicação das ferramentas com o servidor é realizada através do fornecimento do *site* onde o mesmo é hospedado e do *username* e sua chave de acesso. A configuração dos *dashboards* pode ser dividida em duas etapas, a primeira consiste na escolha de seu formato dentre as opções fornecidas pela ferramenta e a segunda na configuração de seus parâmetros. Os formatos utilizados na solução consistem basicamente em: botões de comando que permitem enviar valores para um *feed* do servidor, LEDs que alteram seu valor de acordo com o estado atual de um *feed*, gráficos que mostram os valores lidos de um *feed* ao decorrer do tempo e *gauges* que mostram o valor atual de um *feed* e seu nível de acordo com uma faixa de valores pré-configurada. A segunda etapa consiste na configuração do *dashboard* em si, alterando seus parâmetros, ícones e o associando a um *feed* do protocolo criado na Seção 3.1.

Para a implementação da interface para computador é utilizada a ferramenta provida pela mesma empresa que fornece o servidor MQTT, a Adafruit-IO (2021), e a implementação da interface para celular utiliza o aplicativo IoT MQTT Panel disponível na Google Play Store. A Figura 21 mostra as interfaces para computador e celular implementadas.

Figura 21: Interfaces do Sistema com o Usuário



Interface Computador

Interface Celular

Fonte: Elaborado pelo autor.

Pode-se observar que ambas as interfaces possuem botões para seleção entre os modos de operação dos sistemas de controle de irrigação e de luminosidade, possuem indicadores do estado atual das válvulas, da bomba e das luminárias, possuem gráficos que mostram os históricos dos valores de temperatura, luminosidade e umidade, e possuem *gauges* que mostram o valor atual da temperatura, luminosidade e umidade.

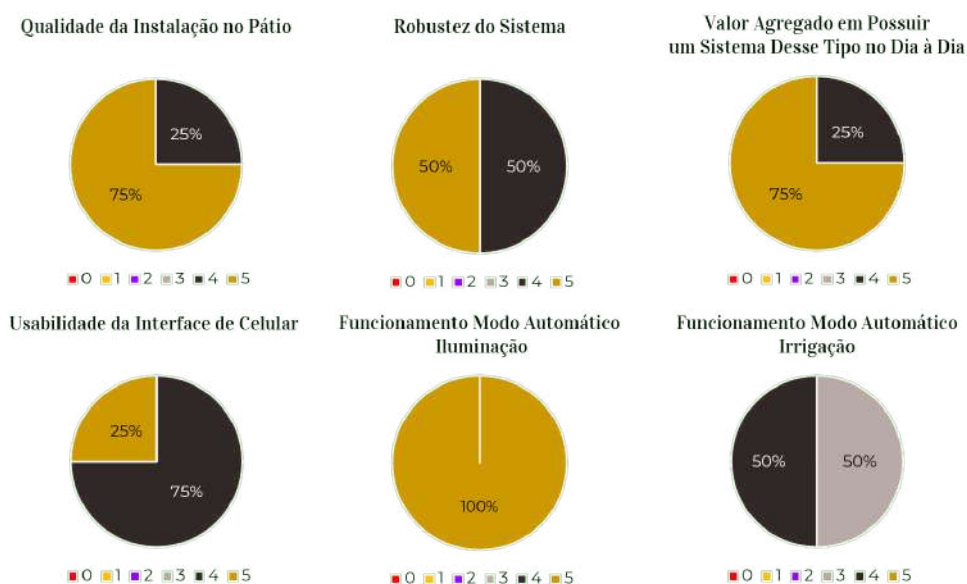
É importante ressaltar que os valores de luminosidade são mostrados ao usuário apenas para o mesmo poder confirmar o correto funcionamento do sistema, porém em um produto final a forma como esses valores são apresentados ao operador deveria ser modificada para o uso de um *dashboard* que apenas indicasse se é dia ou noite.

4 RESULTADOS E DISCUSSÕES

Uma vez concluída a implementação e instalação do sistema de interesse, o presente projeto segue com a avaliação da qualidade do produto obtido e sua relevância para o usuário final. Esta avaliação tem início com uma pesquisa realizada com os operadores, seguindo com uma comparação do estado do gramado do pátio antes e depois do uso do sistema por sessenta dias e finalizando com a coleta de dados para comprovar o correto funcionamento dos modos automáticos de luminosidade e irrigação e da leitura da temperatura ambiente.

A pesquisa com o operador consiste em um conjunto de perguntas realizadas aos quatro moradores da casa que utilizam o produto, para avaliar aspectos relevantes referentes à solução apresentada no presente estudo. Para cada item o usuário atribui uma nota de 0 à 5 e insere comentários para justificar sua resposta. Os resultados da consulta seguem na Figura 22.

Figura 22: Resultado Pesquisa Com Usuário.



Fonte: Elaborado pelo autor.

A partir da Figura 22 pode-se concluir que a usabilidade da interface implementada é avaliada pela maioria dos usuários como nota quatro devido ao fato de o aplicativo escolhido consumir muita bateria quando roda em segundo plano, tornando necessária a configuração do aplicativo para rodar apenas em primeiro plano, o que resulta na não atualização das informações enquanto ele não estiver aberto. Observa-se também que a

qualidade da instalação realizada pelo aluno e o valor agregado em possuir uma solução desse tipo em seu jardim receberam nota máxima da maioria dos usuários, comprovando a qualidade do projeto e sua relevância. Referente aos modos automáticos, o controle de iluminação possui nota máxima atribuída por todos os usuários, devido ao fato de ele funcionar conforme o esperado e a iluminação ambiente e de paisagismo tornarem o pátio mais bonito à noite. Porém o controle de irrigação recebeu as piores notas da pesquisa, o que é justificado pelo fato de o *setpoint* de umidade selecionado possuir um valor maior do que o necessário e a algumas oscilações que ocorrem nos valores de umidade lidos pelos sensores, levando o sistema a irrigar demais o solo e ligar diversas vezes ao dia.

Com os resultados da pesquisa em mãos, alterou-se o *setpoint* de umidade ajustado e inseriu-se um filtro média móvel nas funções de leitura de umidade, melhorando o desempenho do sistema e resultando em sua aprovação por parte dos usuários. Após esse ajuste, o modo automático do controle de irrigação foi ativado para a realização de outro teste para verificar seu funcionamento. No início do teste fotografou-se o gramado do pátio antes de ativar o modo automático. Após 60 dias de sua ativação o jardim foi fotografado novamente para, através da comparação entre as fotos, observar o efeito do sistema de controle no gramado, conforme apresentado na Figura 23. Observa-se que o produto contribui de forma significativa para a melhora da aparência do gramado, fornecendo umidade suficiente para seu desenvolvimento.

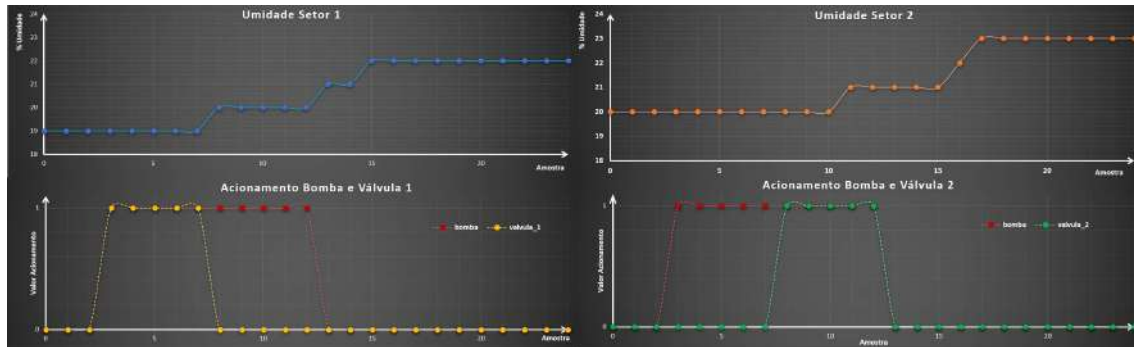
Figura 23: Comparação Gramado Antes e Após Ligar Modo Automático de Irrigação por 60 dias.



Fonte: Elaborado pelo autor.

O último procedimento adotado para analisar o comportamento do modo automático de irrigação é realizado através da coleta de dados referentes a umidade de ambos os setores a serem irrigados e ao acionamento das duas válvulas e da bomba. A Figura 24 apresenta a comparação da umidade lida pelo sensor com o acionamento da bomba e da válvula para cada setor. A figura é obtida através de um ensaio, onde o modo automático de irrigação é ativado na amostra $n = 1$. Ela mostra que o sistema funciona conforme o esperado, ativando a bomba e os setores de irrigação quando a umidade está abaixo do *setpoint*, seguindo a máquina de estados implementada e aumentando o valor de umidade lido pelos sensores após a ativação de seus respectivos setores. Outro fato que se pode observar a partir da figura é que existe um atraso entre o acionamento da irrigação dos setores e o aumento da umidade lida pelos respectivos sensores, o qual se dá pelo tempo de absorção de água pelo solo.

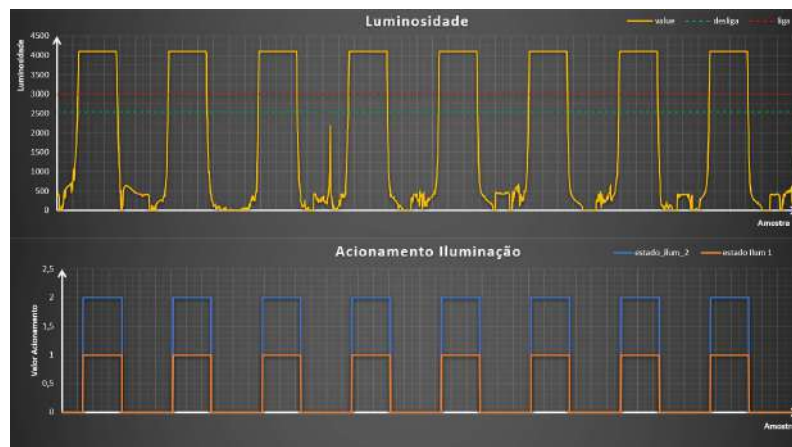
Figura 24: Comportamento Modo Automático Irrigação.



Fonte: Elaborado pelo autor.

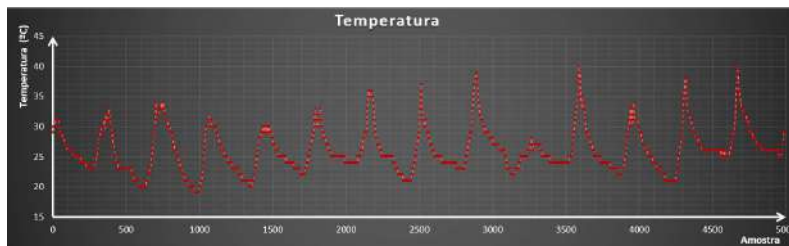
Outra parte do sistema que deve ter seu comportamento analisado consiste no controle do modo automático de iluminação, que é testada através da coleta dos valores lidos pelo sensor de luminosidade e dos relés de acionamento de ambos os setores de iluminação, cujos resultados são apresentados na Figura 25. A partir da figura observa-se que o sistema se comporta conforme o esperado, ligando e desligando as luminárias de acordo com os *setpoints* definidos.

Figura 25: Comportamento Modo Automático Iluminação.



Fonte: Elaborado pelo autor.

Para finalizar a verificação do sistema, são coletados os dados medidos pelo sensor de temperatura, os quais são apresentados na Figura 26. A partir da figura observa-se que o sistema é capaz de ler a temperatura de acordo com o esperado. Além disso, são realizados testes da ativação manual dos setores de irrigação e de iluminação, os quais também atingem os resultados esperados, comprovando o correto funcionamento do produto como um todo.

Figura 26: Comportamento Leitura Temperatura.

Fonte: Elaborado pelo autor.

Após verificar que o sistema se comporta conforme o esperado e atende às expectativas do usuário, o próximo ponto a ser avaliado é se o mesmo cumpre com o objetivo de possuir um baixo custo quando comparado aos valores para sistemas de automação residencial e sistemas de irrigação presentes no mercado (apresentados na Seção 1). Para analisar este quesito, os preços de cada componente e seu somatório são apresentados na Tabela 2, resultando em um valor total de R\$ 1781,63. Pode-se observar que o valor obtido é menor que o valor mínimo para sistemas de automação residencial (R\$ 2000,00) e é próximo do custo para compra de apenas um controlador para sistemas de irrigação (R\$ 1600,00), comprovando que a solução é mais barata que as existentes no mercado.

Tabela 2: Custos do Projeto.

Item	Unidade	Quantidade	Preço Unitário (R\$)	Preço Total (R\$)
Aspersores	Peça	6	27,00	162,00
Válvulas	Peça	2	140,00	280,00
Interruptor Paralelo	Peça	2	6,00	12,00
Tomada 2P+T 20A	Peça	1	9,16	9,16
Lâmpada LED Verde	Peça	3	30,00	90,00
Luminária Tartaruga	Peça	6	12,90	77,40
Mini disjuntor 10 A	Peça	3	8,50	25,50
Esp32	Peça	2	54,15	108,30
Conversor CC/CA	Peça	2	39,90	79,80
Módulo Relés	Peça	2	63,90	127,80
Transformador	Peça	1	60,00	60,00
Sensor Umidade	Peça	2	26,90	53,80
Sensor Luminosidade	Peça	1	7,90	7,90
Sensor Temperatura	Peça	1	22,90	22,90
Placa Fenolite 12x18	Peça	2	11,90	23,80
Bornes	Peça	25	1,90	47,50
Painel 30x22x12	Peça	1	68,90	68,90
Painel 30x18x15	Peça	1	49,90	49,90
Cabo Manga 3 vias AWG 26	m	5	3,16	15,80
Cabo Plasticumbo 1,5mm	m	30	3,45	138,00
Cabo Plasticumbo 2,5mm	m	10	5,27	138,00
Tubo PVC 3/4"	m	45	5,40	243,00
Tubo PVC 1/2"	m	10	5,00	50,00
Mangueira	m	5	2,00	10,00
Total				1781,63

Fonte: Elaborado pelo autor

É importante ressaltar que o uso de módulos é um fator que encareceu o MVP, que pode ser evitado através da implementação de uma placa de circuito impresso onde todos os componentes são soldados, tornando o sistema mais robusto, compacto e barato. Por este motivo, esta modificação deve ser implementada caso o MVP venha a ser comercializado como um produto. Além disso, para não comprometer o andamento do projeto, os cabos e canos PVC utilizados foram adquiridos em ferragens próximas a casa do aluno, onde o custo é mais elevado.

5 CONCLUSÃO

Com base nos procedimentos e testes realizados, foi possível implementar um sistema para o controle, via celular, da irrigação e iluminação de jardins, atendendo os objetivos propostos. Além disso, pode-se concluir que a finalidade do MVP foi atingida, pois ele permitiu o levantamento de dados que demonstram que é possível implementar um sistema que integre o controle de irrigação com a automação residencial (representada pelo controle de iluminação). Através da análise dos custos associados ao desenvolvimento do projeto e sua comparação com os custos de produtos de irrigação de jardins e automação residencial existentes, verifica-se que o conceito abordado neste projeto permite a implementação de uma solução com menor custo e que possui a vantagem de ser expansível.

Durante as etapas finais do projeto foi realizada uma pesquisa com o usuário final da solução proposta, a qual teve como resultado a comprovação do valor agregado deste tipo de solução e o levantamento de alguns pontos de melhoria. Pode-se dizer que, apesar de o projeto proposto se tratar de um MVP e não de um produto, ele já atende as necessidades dos usuários que residem no local onde foi instalado, sendo robusto o suficiente e permitindo uma futura expansão para controlar outros equipamentos, como o portão da garagem, outros pontos de iluminação e a irrigação de outros setores do pátio.

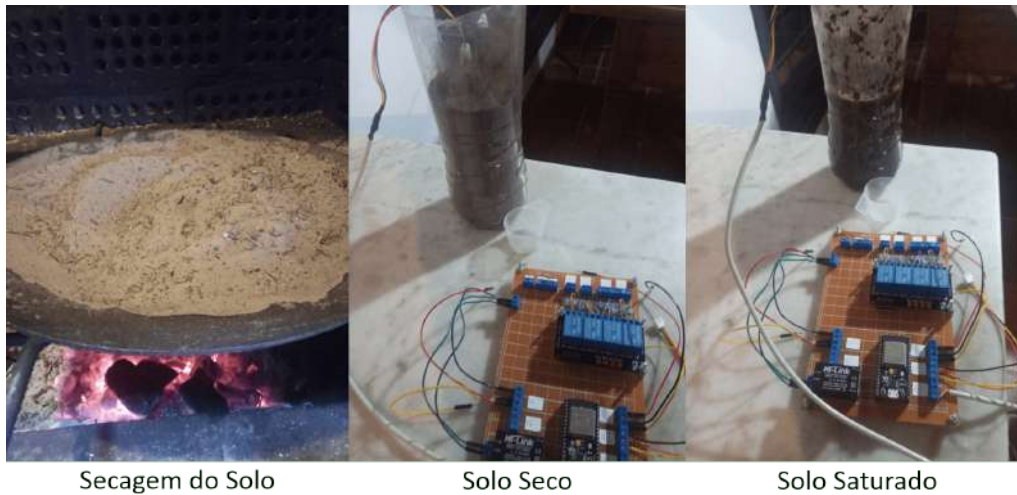
Outro dado relevante levantado durante a execução do projeto consiste nas melhorias necessárias para transformar o MVP executado em um produto. Uma delas é a troca do aplicativo para celular, solucionando o problema do alto consumo de bateria relatado na pesquisa com o usuário. Outra é o projeto de uma placa de circuito impresso onde todos os componentes possam ser soldados, substituindo os módulos utilizados para reduzir custos, aumentar a robustez do sistema e torná-lo mais compacto. Além disso, existem outras melhorias que podem ser temas de trabalhos futuros. Entre elas citamos o uso de inteligência artificial para determinar os parâmetros do modo automático de irrigação, a partir dos dados de temperatura, luminosidade, umidade, características do solo e de dados climáticos. Também citamos a implementação de um servidor MQTT em um microcontrolador para permitir o uso de mais *feeds* e de qualidade de serviço diferentes da zero (assegurando o envio e recebimento dos dados), por exemplo, a empresa adafruit fornece o código fonte de seu servidor e alega que ele pode ser portado para um Raspberry Pi. Outros temas citados são a integração do servidor MQTT com um assistente virtual e sua configuração para tornar sua comunicação criptografada. Também pode-se trabalhar no uso dos dados coletados para avaliar a precisão e a exatidão das medidas realizadas pelos sensores e na implementação de autotestes para verificar o correto funcionamento dos equipamentos do produto, atividades não realizadas no presente projeto.

Por fim, considerando o conteúdo apresentado neste trabalho, acredita-se ter conseguido demonstrar a viabilidade conceitual e técnica do sistema proposto possibilitando, inclusive, o desenvolvimento de um produto robusto e com potencial de comercialização no mercado.

APÊNDICE A - DIAGRAMAS E FIGURAS

A.1 Figuras das Etapas do Ensaio de Irrigação do Solo

Figura 27: *Ensaio Calibração Sensor de Umidade: Etapas e Materiais Utilizados*



Secagem do Solo

Solo Seco

Solo Saturado

Fonte: Elaborado pelo autor.

Figura 28: *Ensaio Calibração Sensor de Umidade: Solo Seco e Saturado*



Fonte: Elaborado pelo autor.

A.2 **Figura das Etapas do Ensaio para Obter o Tempo de Absorção de Água pelo Solo**

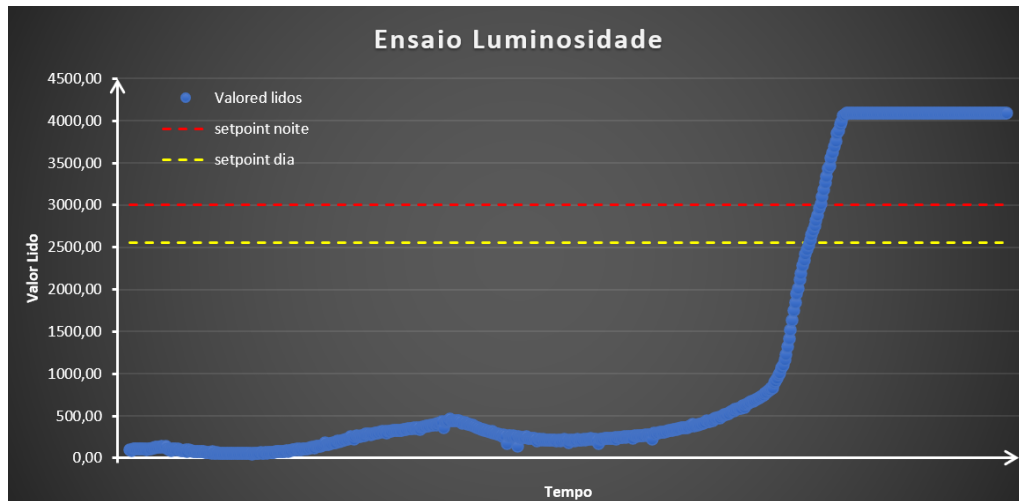
Figura 29: *Ensaio Tempo de Absorção de Umidade pelo Solo*



Fonte: Elaborado pelo autor.

A.3 Figura Ensaio Definição *Setpoints* Luminosidade

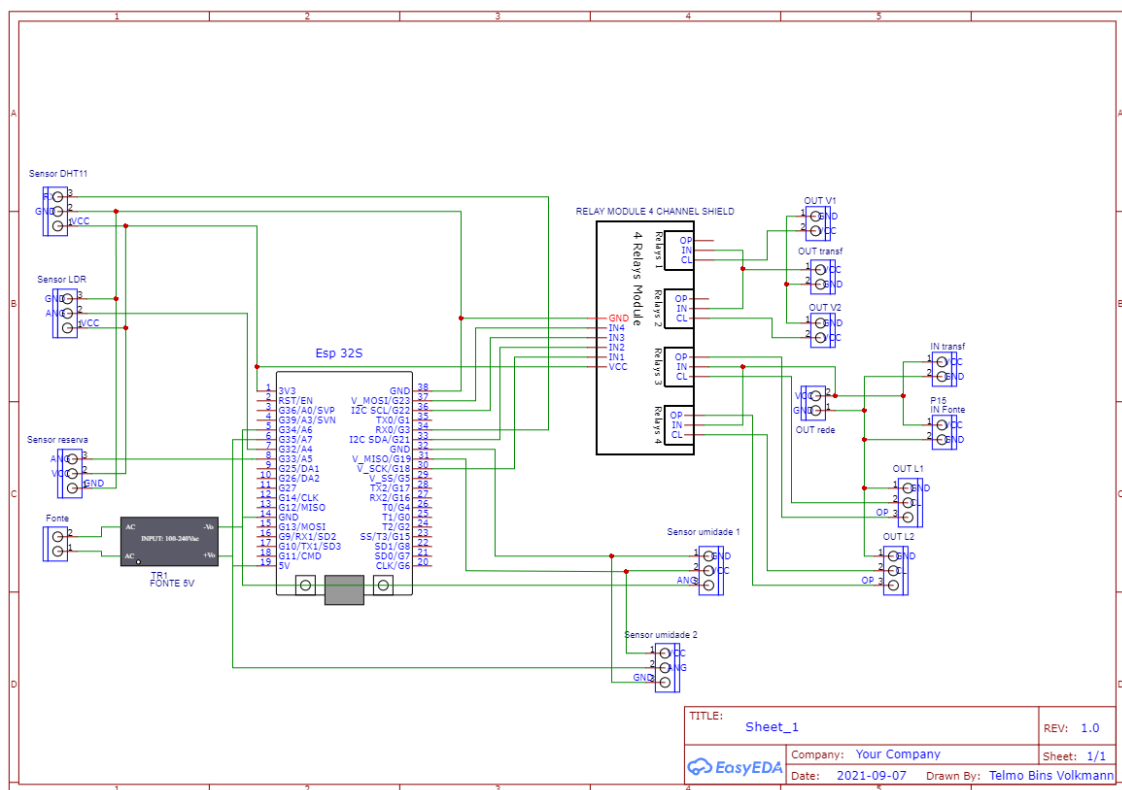
Figura 30: Ensaio *Setpoints* Controle de Luminosidade



Fonte: Elaborado pelo autor.

A.4 Diagrama Elétrico da Placa do Controlador Principal

Figura 31: Diagrama Elétrico da Placa do Controlador Principal.



Fonte: Elaborado pelo autor.

APÊNDICE B - CÓDIGOS FONTE DOS *SOFTWARES* DO PROJETO

B.1 Arquivo com Configurações de Rede e do Servidor MQTT para os dois Controladores

```

1  /***** Adafruit IO Config *****/
2  #define IO_USERNAME "tbvolkmann"
3  #define IO_KEY "aio_DABo940KSN0kh75FuclVA4BHZ7J"
4  /***** WiFi *****/
5  #define WIFI_SSID "Volkman"
6  #define WIFI_PASS "av225084"
7
8  #include "AdafruitIO_WiFi.h"
9
10 #if defined(USE_AIRLIFT) || defined(ADAFRUIT_METRO_M4_AIRLIFT_LITE) || \
11     defined(ADAFRUIT_PYPORTAL)
12 // Configure the pins used for the ESP32 connection
13 #if !defined(SPIWIFI_SS) // if the wifi definition isnt in the board variant
14 // Don't change the names of these #define's! they match the variant ones
15 #define SPIWIFI_SPI
16 #define SPIWIFI_SS 10 // Chip select pin
17 #define NINA_ACK 9 // a.k.a BUSY or READY pin
18 #define NINA_RESETN 6 // Reset pin
19 #define NINA_GPIO0 -1 // Not connected
20 #endif
21 AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS, SPIWIFI_SS,
22     NINA_ACK, NINA_RESETN, NINA_GPIO0, &SPIWIFI);
23 #else
24 AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);
25 #endif

```

B.2 Código Fonte *Software* Controlador Principal

```

1  #include "config.h" //Wifi e MQTT Server Configuration File
2  #include "DHT.h"
3  #define DHTTYPE DHT11
4  #define Port_Ilum_Dec 23
5  #define Port_Ilum_Amb 22
6  #define Port_Valvula_1 21
7  #define Port_Valvula_2 18
8  #define Port_power_LDR 12
9  #define Port_power_Umid 19
10 #define Port_Anlog_LDR 32
11 #define Port_Anlog_Reserva 33
12 #define Port_Anlog_Umid_Set_1 34
13 #define Port_Anlog_Umid_Set_2 35
14 #define Port_Rx_DHT11 3
15 DHT dht(Port_Rx_DHT11, DHTTYPE);
16 //Delays
17 #define Ilum_DELAY 300000 //Delay para ler valores LDR - 5min
18 #define Umid_DELAY 180000 //Delay para ler valores Umidade - 3min
19 #define Temp_DELAY 240000 //Delay para ler valores Temperatura - 4min
20 #define Modo_Auto_Maq_Estados_Delay 180000 //Delay para ler atualizar valores da maquina de estados do modo automatico de irrigacao - 3min
21 #define Valvula_Aberta_Delay 900000 //Tempo que a valvula vai ficar aberta - 15min
22 #define Cont_Irrig_Parado_Delay 900000 //Tempo que o controle de irrigacao fica parado para esperar absorcao do solo - 15min
23 #define Lum_Histerese 0.85 //Histerese para sensor de luminosidade
24 #define a_umid -0.02 //Valores Ajuste curva sensores umidade
25 #define b_umid 55.18 //Valores Ajuste curva sensores umidade
26 unsigned long Ilum_Last_Update = 0;
27 //variaveis usadas para os timers
28 unsigned long Umid_Last_Update = 0;
29 //variaveis usadas para os timers
30 unsigned long Temp_Last_Update = 0;
31 //variaveis usadas para os timers
32 unsigned long Modo_Auto_Maq_Estados_Last_Update = 0;
33 unsigned long Valvula_Aberta_Last_Update = 0;

```

```

34 unsigned long Cont_Irrig_Parado_Last_Update = 0;
35 unsigned long last_millis = 0;
36 int Estado_Irrig_Modo_Auto = 0; //variavel de estado da maquina de estados que controla o modo automatico de irrigacao
37 int Estado_Ilum_Amb = 0; //Estados Reles
38 int Estado_Ilum_Dec = 0; //Estados Reles
39 int Estado_Irrig_Setor_1 = 0; //Estados Reles
40 int Estado_Irrig_Setor_2 = 0; //Estados Reles
41 int Irrig_Modo_Auto = 0; //Flag para ativar modo automatico irrigacao
42 long SetPoint_Luminosidade = 3000; //Setpoints
43 long SetPoint_Umididade = 20; //Setpoints
44 int Estado_Ilum_MQTT = 0; //Valores a serem Escritos nos Feeds MQTT
45 int Estado_Irrig_MQTT = 0; //Valores a serem Escritos nos Feeds MQTT
46 int Estado_Bomba_MQTT = 0; //Valores a serem Escritos nos Feeds MQTT
47 int Controle_Irrigacao_MQTT = 0; //Valores a serem Lidos dos Feeds MQTT
48 int Controle_Iluminacao_Ambiente_MQTT = 0; //Valores a serem Lidos dos Feeds MQTT
49 int Controle_Iluminacao_Decoracao_MQTT = 0; //Valores a serem Lidos dos Feeds MQTT
50 long Umidade_Set_1 = 0; //Valores medidos pelos sensores
51 long Umidade_Set_1_Vec[4]; //Valores medidos pelos sensores
52 long Umidade_Set_2 = 0; //Valores medidos pelos sensores
53 long Umidade_Set_2_Vec[4]; //Valores medidos pelos sensores
54 long Luminosidade = 0; //Valores medidos pelos sensores
55 long Temperature = 0; //Valores medidos pelos sensores
56 AdafruitIO_Feed *Umid_set_1_Feed = io.feed("umidade-setor-1"); //Feed sensor umidade setor 1
57 AdafruitIO_Feed *Umid_set_2_Feed = io.feed("umidade-setor-2"); //Feed sensor umidade setor 2
58 AdafruitIO_Feed *Temp_Feed = io.feed("temperatura"); //Feed sensor temperatura
59 AdafruitIO_Feed *Lumi_Feed = io.feed("luminosidade"); //Feed sensor temperatura
60 AdafruitIO_Feed *Estado_Valvulas_Feed = io.feed("estado-valvulas"); //Feed estado valvulas
61 AdafruitIO_Feed *Estado_Iluminacao_Feed = io.feed("estado-iluminacao"); //Feed estado iluminacao
62 AdafruitIO_Feed *Estado_Bomba_Feed = io.feed("estado-bomba"); //Feed estado bomba
63 AdafruitIO_Feed *Controle_Irrig_Feed = io.feed("controle-irrigacao"); //Feed controle irrigacao
64 AdafruitIO_Feed *Controle_Ilum_Amb_Feed = io.feed("controle-iluminacao-ambiente"); //Feed controle iluminacao do ambiente
65 AdafruitIO_Feed *Controle_Ilum_Dec_Feed = io.feed("controle-iluminacao-decoracao"); //Feed controle iluminacao decoracao
66 void setup()
67 {
68   Adafruit_Connect(); //Conexao ao servidor adafruit e a rede wifi
69   send_int_to_mqtt(Estado_Irrig_MQTT, Estado_Valvulas_Feed, "Estado_Valvulas_Feed"); //Reseta valor Feed do servidor MQTT
70   send_int_to_mqtt(Estado_Bomba_MQTT, Estado_Bomba_Feed, "Estado_Bomba_Feed"); //Reseta valor Feed do servidor MQTT
71   send_int_to_mqtt(Estado_Ilum_MQTT, Estado_Iluminacao_Feed, "Estado_Iluminacao_Feed"); //Envia valor Feed ao servidor MQTT
72   pinMode(Port_Ilum_Dec, OUTPUT);
73   pinMode(Port_Ilum_Amb, OUTPUT);
74   pinMode(Port_Valvula_2, OUTPUT);
75   pinMode(Port_Valvula_1, OUTPUT);
76   pinMode(Port_power_LDR, OUTPUT);
77   pinMode(Port_power_Umid, OUTPUT);
78   pinMode(Port_Anlog_LDR, INPUT);
79   pinMode(Port_Anlog_Reserva, INPUT);
80   pinMode(Port_Rx_DHT11, INPUT);
81   pinMode(Port_Anlog_Umid_Set_2, INPUT);
82   pinMode(Port_Anlog_Umid_Set_1, INPUT);
83   digitalWrite(Port_Ilum_Dec, HIGH);
84   digitalWrite(Port_Ilum_Amb, HIGH);
85   digitalWrite(Port_Valvula_2, HIGH);
86   digitalWrite(Port_Valvula_1, HIGH);
87   digitalWrite(Port_power_LDR, HIGH);
88   digitalWrite(Port_power_Umid, HIGH);
89   umid_sensors_begin();
90   dht.begin();
91 }
92 void loop()
93 {
94   if (io.status() < AIO_CONNECTED) //Verifica conexao ao servidor adafruit e a rede wifi -> se nao conectado tenta reconectar
95     Adafruit_Connect();
96   io.run(); //Mantem a comunicacao com o servidor
97   if (millis() > Temp_Last_Update + Temp_DELAY){ //Temporizacao da leitura da temperatura
98     Read_Temperature();
99     Temp_Last_Update = millis();
100 }
101 if (millis() > Umid_Last_Update + Umid_DELAY){ //Temporizacao da leitura da luminosidade
102   Read_Umid_Sensor1(Port_Anlog_Umid_Set_1, &Umidade_Set_1, Umid_set_1_Feed, "Umid_set_1_Feed");
103   Read_Umid_Sensor2(Port_Anlog_Umid_Set_2, &Umidade_Set_2, Umid_set_2_Feed, "Umid_set_2_Feed");
104   Umid_Last_Update = millis();
105 }
106 if (millis() > Ilum_Last_Update + Ilum_DELAY){ //Temporizacao da leitura da luminosidade
107   Read_LDR_Sensor(Port_Anlog_LDR, &Luminosidade, Lumi_Feed, "Lumi_Feed");
108   Controle_Ilum(); //Controle de Iluminacao ambiente e decorativa
109   Ilum_Last_Update = millis();
110 }
111 if (millis() > Modo_Auto_Maq_Estados_Last_Update + Modo_Auto_Maq_Estados_Delay){ //Temporizacao da leitura da luminosidade
112   Controle_Irrig(); //Controle de Irrigacao -> ativa manualmente setor ou ativa modo automatico de acordo com periodo do dia escolhido
113   Modo_Auto_Irrig();
114   Modo_Auto_Maq_Estados_Last_Update = millis();
115 }
116 if(last_millis > millis()){ //Ajuste quando millis estourar seu valor maximo (aproximadamente 50 dias)
117   Ilum_Last_Update = millis();
118   Umid_Last_Update = millis();
119   Temp_Last_Update = millis();
120   Modo_Auto_Maq_Estados_Last_Update = millis();
121   Valvula_Aberta_Last_Update = millis();
122   Cont_Irrig_Parado_Last_Update = millis();
123 }
124 last_millis = millis();
125 }
126 void umid_sensors_begin()
127 {
128   long val;
129   val = round(a_umid * analogRead(Port_Anlog_Umid_Set_1) + b_umid);
130   Umidade_Set_1_Vec[0]=val;
131   val = round(a_umid * analogRead(Port_Anlog_Umid_Set_1) + b_umid);
132   Umidade_Set_1_Vec[1]=val;

```



```

133 val = round(a_umid * analogRead(Port_Anlog_Umid_Set_1) + b_umid);
134 Umidade_Set_1_Vec[2]=val;
135 val = round(a_umid * analogRead(Port_Anlog_Umid_Set_1) + b_umid);
136 Umidade_Set_1_Vec[3]=val;
137 val = round(a_umid * analogRead(Port_Anlog_Umid_Set_2) + b_umid);
138 Umidade_Set_2_Vec[0]=val;
139 val = round(a_umid * analogRead(Port_Anlog_Umid_Set_2) + b_umid);
140 Umidade_Set_2_Vec[1]=val;
141 val = round(a_umid * analogRead(Port_Anlog_Umid_Set_2) + b_umid);
142 Umidade_Set_2_Vec[2]=val;
143 val = round(a_umid * analogRead(Port_Anlog_Umid_Set_2) + b_umid);
144 Umidade_Set_2_Vec[3]=val;
145 }
146 void Adafruit_Connect() //conexao com servidor Adafruit e Wifi
147 {
148   Serial.begin(115200);
149   Serial.println("Connecting_to_Adafruit_IO");
150   io.connect();
151   Controle_Irrig_Feed->onMessage(Read_Control_Irrig);
152   Controle_Ilum_Amb_Feed->onMessage(Read_Control_Ilum_Amb);
153   Controle_Ilum_Dec_Feed->onMessage(Read_Control_Ilum_Dec);
154   while (io.status() < AIO_CONNECTED) {
155     Serial.println("Connecting_Status:_");
156     Serial.print(io.statusText());
157   }
158   Controle_Irrig_Feed->get();
159   Controle_Ilum_Amb_Feed->get();
160   Controle_Ilum_Dec_Feed->get();
161 }
162 void send_int_to_mqtt(int value, AdafruitIO_Feed *Feed, String Feed_Name) //Funcao para enviar valor ao servidor MQTT
163 {
164   Feed->save(value);
165 }
166 void Read_Control_Ilum_Amb(AdafruitIO_Data *data) //Leitura feed controle iluminacao ambiente
167 {
168   Controle_Iluminacao_Ambiente_MQTT = data->toInt();
169   Controle_Ilum();
170 }
171 void Read_Control_Ilum_Dec(AdafruitIO_Data *data) //Leitura feed controle iluminacao decorativa
172 {
173   Controle_Iluminacao_Decoracao_MQTT = data->toInt();
174   Controle_Ilum();
175 }
176 void Read_Control_Irrig(AdafruitIO_Data *data) //Leitura feed controle irrigacao
177 {
178   Controle_Irrigacao_MQTT = data->toInt();
179   Controle_Irrig();
180 }
181 void Read_Umid_Sensor1(int SensorPort, long *valor_lido, AdafruitIO_Feed *Feed_Onde_Escrever, String Nome_Feed)
182 {
183   long val;
184   val = round(a_umid * analogRead(SensorPort) + b_umid); //leitura do valor de umidade do sensor e conversao para % de umidade
185   if(abs(val-*valor_lido)<10){
186     Umidade_Set_1_Vec[0] = Umidade_Set_1_Vec[1];
187     Umidade_Set_1_Vec[1] = Umidade_Set_1_Vec[2];
188     Umidade_Set_1_Vec[2] = Umidade_Set_1_Vec[3];
189     Umidade_Set_1_Vec[3] = val;
190   }
191   *valor_lido = (Umidade_Set_1_Vec[0]+Umidade_Set_1_Vec[1]+Umidade_Set_1_Vec[2]+Umidade_Set_1_Vec[3])/4 //leitura sensor e conversao para % de umidade
192   send_int_to_mqtt(*valor_lido, Feed_Onde_Escrever, Nome_Feed);
193 }
194 void Read_Umid_Sensor2(int SensorPort, long *valor_lido, AdafruitIO_Feed *Feed_Onde_Escrever, String Nome_Feed)
195 {
196   long val;
197   val = round(a_umid * analogRead(SensorPort) + b_umid); //leitura do valor de umidade do sensor e conversao para % de umidade
198   if(abs(val-*valor_lido)<10){
199     Umidade_Set_2_Vec[0] = Umidade_Set_2_Vec[1];
200     Umidade_Set_2_Vec[1] = Umidade_Set_2_Vec[2];
201     Umidade_Set_2_Vec[2] = Umidade_Set_2_Vec[3];
202     Umidade_Set_2_Vec[3] = val;
203   }
204   *valor_lido = (Umidade_Set_2_Vec[0]+Umidade_Set_2_Vec[1]+Umidade_Set_2_Vec[2]+Umidade_Set_2_Vec[3])/4 //leitura sensor e conversao para % de umidade
205   send_int_to_mqtt(*valor_lido, Feed_Onde_Escrever, Nome_Feed);
206 }
207 void Read_LDR_Sensor(int SensorPort, long *valor_lido, AdafruitIO_Feed *Feed_Onde_Escrever, String Nome_Feed)
208 {
209   *valor_lido = analogRead(SensorPort);
210   send_int_to_mqtt(*valor_lido, Feed_Onde_Escrever, Nome_Feed);
211 }
212 void Read_Temperature()
213 {
214   float temp;
215   temp = dht.readTemperature();
216   if(isnan(temp))
217     Serial.println("Erro_Leitura_Temperatura");
218   else{
219     Temperature = round(temp);
220     send_int_to_mqtt(Temperature, Temp_Feed, "Temp_Feed");
221   }
222 }
223 void Controle_Ilum() //Funcao Controle de Iluminacao ambiente e decorativa
224 {
225   switch (Controle_Iluminacao_Decoracao_MQTT){
226     case 0:
227       Estado_Ilum_Dec = 0; //Iluminacao desligada
228       digitalWrite(Port_Ilum_Dec, HIGH);
229       break;
230     case 1:
231       Estado_Ilum_Dec = 1; //Iluminacao ligada manualmente

```

```

232 digitalWrite(Port_Ilum_Dec, LOW);
233 break;
234 case 2:
235 if (Luminosidade >= SetPoint_Luminosidade){
236     Estado_Ilum_Dec = 1; //Iluminacao liga
237     digitalWrite(Port_Ilum_Dec, LOW);
238 }
239 else{
240     if(Luminosidade < (SetPoint_Luminosidade*Lum_Histerese)){
241         Estado_Ilum_Dec = 0; //Iluminacao desliga
242         digitalWrite(Port_Ilum_Dec, HIGH);
243     }
244 }
245 break;
246 default:
247     break;
248 }
249 switch (Controle_Iluminacao_Ambiente_MQTT) //Controle Iluminacao Ambiente{
250 case 0:
251     Estado_Ilum_Amb = 0; //Iluminacao desligada
252     digitalWrite(Port_Ilum_Amb, HIGH);
253     break;
254 case 1:
255     Estado_Ilum_Amb = 1; //Iluminacao ligada manualmente
256     digitalWrite(Port_Ilum_Amb, LOW);
257     break;
258 case 2:
259     if (Luminosidade >= SetPoint_Luminosidade){
260         Estado_Ilum_Amb = 1; //Iluminacao liga
261         digitalWrite(Port_Ilum_Amb, LOW);
262     }
263     else{
264         if(Luminosidade < (SetPoint_Luminosidade*Lum_Histerese)){
265             Estado_Ilum_Amb = 0; //Iluminacao desliga
266             digitalWrite(Port_Ilum_Amb, HIGH);
267         }
268     }
269     break;
270 default:
271     break;
272 }
273 Estado_Ilum_MQTT = Estado_Ilum_Amb + (Estado_Ilum_Dec << 1);
274 send_int_to_mqtt(Estado_Ilum_MQTT, Estado_Iluminacao_Feed, "Estado_Iluminacao_Feed"); //Envia valor Feed ao servidor MQTT
275 }
276 void Controle_Irrig() //Funcao Controle de Irrigacao
277 {
278     switch (Controle_Irrigacao_MQTT){
279     case 0:
280         Estado_Irrig_Setor_1 = 0;
281         Estado_Irrig_Setor_2 = 0;
282         Estado_Bomba_MQTT = 0;
283         Irrig_Modo_Auto = 0;
284         Estado_Irrig_Modo_Auto = 0;
285         break;
286     case 1:
287         Estado_Irrig_Setor_1 = 1;
288         Estado_Irrig_Setor_2 = 0;
289         Estado_Bomba_MQTT = 1;
290         Irrig_Modo_Auto = 0;
291         Estado_Irrig_Modo_Auto = 0;
292         break;
293     case 2:
294         Estado_Irrig_Setor_1 = 0;
295         Estado_Irrig_Setor_2 = 1;
296         Estado_Bomba_MQTT = 1;
297         Irrig_Modo_Auto = 0;
298         Estado_Irrig_Modo_Auto = 0;
299         break;
300     case 3:
301         if (Luminosidade < SetPoint_Luminosidade)
302             Irrig_Modo_Auto = 1;
303         else{
304             Estado_Irrig_Setor_1 = 0;
305             Estado_Irrig_Setor_2 = 0;
306             Estado_Bomba_MQTT = 0;
307             Irrig_Modo_Auto = 0;
308             Estado_Irrig_Modo_Auto = 0;
309         }
310         break;
311     case 4:
312         if (Luminosidade >= SetPoint_Luminosidade)
313             Irrig_Modo_Auto = 1;
314         else{
315             Estado_Irrig_Setor_1 = 0;
316             Estado_Irrig_Setor_2 = 0;
317             Estado_Bomba_MQTT = 0;
318             Irrig_Modo_Auto = 0;
319             Estado_Irrig_Modo_Auto = 0;
320         }
321         break;
322     case 5:
323         Irrig_Modo_Auto = 1;
324         break;
325     default:
326         break;
327 }
328 if (Irrig_Modo_Auto == 0){
329     Estado_Irrig_MQTT = Estado_Irrig_Setor_1 + (Estado_Irrig_Setor_2 << 1);
330     send_int_to_mqtt(Estado_Irrig_MQTT, Estado_Valvulas_Feed, "Estado_Valvulas_Feed"); //Envia valor Feed ao servidor MQTT

```

```

331 send_int_to_mqtt(Estado_Bomba_MQTT, Estado_Bomba_Feed, "Estado_Bomba_Feed"); //Envia valor Feed ao servidor MQTT
332 if (Estado_Irrig_Setor_1 == 1)
333     digitalWrite(Port_Valvula_1, LOW);
334     else
335     digitalWrite(Port_Valvula_1, HIGH);
336 if (Estado_Irrig_Setor_2 == 1)
337     digitalWrite(Port_Valvula_2, LOW);
338     else
339     digitalWrite(Port_Valvula_2, HIGH);
340 }
341 }
342
343 void Modo_Auto_Irrig()
344 {
345     if (Irrig_Modo_Auto == 1){
346         switch (Estado_Irrig_Modo_Auto){
347             case 0:
348                 Estado_Irrig_Setor_1 = 0;
349                 Estado_Irrig_Setor_2 = 0;
350                 Estado_Bomba_MQTT = 0;
351                 Estado_Irrig_Modo_Auto = 1; //Proximo estado da maquina de estados
352                 break;
353             case 1:
354                 if (Umidade_Set_1 < SetPoint_Umidade){
355                     Estado_Irrig_Modo_Auto = 2; //Proximo estado da maquina de estados
356                     Valvula_Aberta_Last_Update = millis();
357                 }
358                 else
359                     Estado_Irrig_Modo_Auto = 3; //Proximo estado da maquina de estados
360                 break;
361             case 2:
362                 Estado_Irrig_Setor_1 = 1;
363                 Estado_Bomba_MQTT = 1;
364                 if (millis() > Valvula_Aberta_Last_Update + Valvula_Aberta_Delay){
365                     if (Umidade_Set_2 < SetPoint_Umidade){
366                         Estado_Irrig_Modo_Auto = 4; //Proximo estado da maquina de estados
367                         Valvula_Aberta_Last_Update = millis();
368                     }
369                     else{
370                         Estado_Irrig_Modo_Auto = 5; //Proximo estado da maquina de estados
371                         Cont_Irrig_Parado_Last_Update = millis();
372                     }
373                 }
374                 break;
375             case 3:
376                 if (Umidade_Set_2 < SetPoint_Umidade){
377                     Estado_Irrig_Modo_Auto = 4; //Proximo estado da maquina de estados
378                     Valvula_Aberta_Last_Update = millis();
379                 }
380                 else{
381                     Estado_Irrig_Modo_Auto = 0; //Proximo estado da maquina de estados
382                     Cont_Irrig_Parado_Last_Update = millis();
383                 }
384                 break;
385             case 4:
386                 Estado_Irrig_Setor_2 = 1;
387                 Estado_Irrig_Setor_1 = 0;
388                 Estado_Bomba_MQTT = 1;
389                 if (millis() > Valvula_Aberta_Last_Update + Valvula_Aberta_Delay){
390                     Estado_Irrig_Modo_Auto = 5; //Proximo estado da maquina de estados
391                     Cont_Irrig_Parado_Last_Update = millis();
392                 }
393                 break;
394             case 5:
395                 Estado_Irrig_Setor_2 = 0;
396                 Estado_Irrig_Setor_1 = 0;
397                 Estado_Bomba_MQTT = 0;
398                 if (millis() > Cont_Irrig_Parado_Last_Update + Cont_Irrig_Parado_Delay)
399                     Estado_Irrig_Modo_Auto = 0; //Proximo estado da maquina de estados
400                 break;
401             default:
402                 break;
403         }
404         Estado_Irrig_MQTT = Estado_Irrig_Setor_1 + (Estado_Irrig_Setor_2 << 1);
405         send_int_to_mqtt(Estado_Irrig_MQTT, Estado_Valvulas_Feed, "Estado_Valvulas_Feed"); //Envia valor Feed ao servidor MQTT
406         send_int_to_mqtt(Estado_Bomba_MQTT, Estado_Bomba_Feed, "Estado_Bomba_Feed"); //Envia valor Feed ao servidor MQTT
407         if (Estado_Irrig_Setor_1 == 1)
408             digitalWrite(Port_Valvula_1, LOW);
409             else
410             digitalWrite(Port_Valvula_1, HIGH);
411         if (Estado_Irrig_Setor_2 == 1)
412             digitalWrite(Port_Valvula_2, LOW);
413             else
414             digitalWrite(Port_Valvula_2, HIGH);
415     }
416     else{
417         Estado_Irrig_Modo_Auto = 0;
418     }
419 }

```

B.3 Código Fonte *Software* Controlador Bomba

```

1 #include "config.h" //Wifi e MQTT Server Configuration File
2 #define Port_Reserv_220V_1 23
3 #define Port_Reserv_220V_2 22
4 #define Port_Bomba 21
5 #define Port_Valvula_Irrig 18
6 #define Bomba_DELAY 60000 //Delay para ler valores Bomba = 1min
7 unsigned long Bomba_Last_Update = 0; //variaveis usadas para os timers
8 unsigned long Last_millis = 0; //variaveis usadas para os timers
9 int Controle_Bomba_MQTT = 0; //Valores a serem Lidos dos Feeds MQTT
10 AdafruitIO_Feed *Estado_Bomba_Feed = io.feed("estado-bomba"); //Feed estado bomba
11 void setup()
12 {
13   Adafruit_Connect(); //Conexao ao servidor adafruit e a rede wifi
14   pinMode(Port_Reserv_220V_1, OUTPUT);
15   pinMode(Port_Reserv_220V_2, OUTPUT);
16   pinMode(Port_Valvula_Irrig, OUTPUT);
17   pinMode(Port_Bomba, OUTPUT);
18   digitalWrite(Port_Reserv_220V_1, HIGH);
19   digitalWrite(Port_Reserv_220V_2, HIGH);
20   digitalWrite(Port_Valvula_Irrig, HIGH);
21   digitalWrite(Port_Bomba, HIGH);
22 }
23 void loop()
24 {
25   if (io.status() < AIO_CONNECTED){ //Verifica conexao ao servidor adafruit e a rede wifi -> se nao conectado tenta reconectar
26     digitalWrite(Port_Bomba, HIGH); //Desliga a bomba em caso de desconexao - bomba ficar ligada a vazio pode causar defeito na mesma
27     Adafruit_Connect();
28   }
29   io.run(); //Mantã o m a comunicaã§ãõ com o servidor
30   if (millis() > Bomba_Last_Update + Bomba_DELAY){ //Temporizacao da leitura da bomba - usado apenas para garantir leitura do valor da bomba
31     Estado_Bomba_Feed->get();
32     Bomba_Last_Update = millis();
33   }
34   Controle_Bomba();
35   if(Last_millis > millis()){ //Ajuste para evitar erro quando millis estourar seu valor maximo (aproximadamente 50 dias)
36     Bomba_Last_Update = millis();
37   }
38   Last_millis = millis();
39 }
40 void Adafruit_Connect() //conexao com servidor Adafruit e Wifi
41 {
42   Serial.begin(115200);
43   Serial.print("Connecting_to_Adafruit_IO");
44   io.connect();
45   Estado_Bomba_Feed->onMessage(Read_Control_Bomba);
46   while (io.status() < AIO_CONNECTED) {
47     Serial.print("\n");
48     Serial.print("Connecting_Status:");
49     Serial.print(io.statusText());
50   }
51   Estado_Bomba_Feed->get();
52 }
53 void Read_Control_Bomba(AdafruitIO_Data *data) //Leitura feed controle bomba
54 {
55   Controle_Bomba_MQTT = data->toInt();
56 }
57 void Controle_Bomba()
58 {
59   if(Controle_Bomba_MQTT==1){
60     digitalWrite(Port_Bomba, LOW);
61     Serial.println("Bomba_on");
62   }
63   else{
64     digitalWrite(Port_Bomba, HIGH);
65     Serial.println("Bomba_off");
66   }
67 }

```

REFERÊNCIAS

- ADAFRUIT-IO. **Site Oficial**. [S.l.: s.n.], 2021. Disponível em: <https://io.adafruit.com/>. Acesso em: 03 de fev. 2021.
- ADOLPHS, R. S. **Sistema de Irrigação Automático Supervisionado para Hortas**. 2020. f. 34. Trabalho de Conclusão de Curso (Bacharel em Engenharia de Controle e Automação) – Universidade Federal do Rio Grande do Sul, Porto Alegre.
- ARDUINO-UNO. **Datasheet**. [S.l.: s.n.], 2021. Disponível em: <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>. Acesso em: 12 de fev. 2021.
- CONCEIÇÃO, W. N. E. DA; RESENDE COSTA, R. M. DE. **Análise do Protocolo MQTT para Comunicação IoT através de um Cenário de Comunicação**. *Caderno de Estudos em Sistemas de Informação*, v. 5, n. 2, 2019.
- DHT11. **Datasheet**. [S.l.: s.n.], 2021. Disponível em: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>. Acesso em: 12 de fev. 2021.
- EASYEDA. **Ferramenta para Projeto de Placas de Circuito Impresso**. [S.l.: s.n.], 2021. Disponível em: <https://easyeda.com/>. Acesso em: 12 de fev. 2021.
- FOLTECH. **Site Oficial**. [S.l.: s.n.], 2021. Disponível em: https://www.foltechautomacao.com.br/?gclid=Cj0KCQiAip-PBhDVARIsAPP2xc2R0Y0wcrmZjmKaw8hq4vNzVcV0smhlm3BgDPubwgYsesh9WStY2Y8aAmNEEALw_wcB. Acesso em: 26 de jan. 2021.
- FRIZZONE, J. A. **Texto Complementar 2 - Necessidade de Água para Irrigação**. [S.l.: s.n.], 2017. Disponível em: http://www.leb.esalq.usp.br/leb/disciplinas/Frizzone/LEB_1571/TEXTO_COMPLEMENTAR_2_-_NECESSIDADE_DE_AGUA_PARA_IRRIGACAO.pdf. Acesso em: 06 de mar. 2022.
- HLK-PM01. **Datasheet**. [S.l.: s.n.], 2021. Disponível em: <https://www.filipeflop.com/img/files/download/Datasheet-HLK-PM01.pdf>. Acesso em: 12 de fev. 2021.
- HOMESYSTEMS. **Site Oficial**. [S.l.: s.n.], 2021. Disponível em: <http://www.homesystems.com.br/solucoes/>. Acesso em: 26 de jan. 2021.
- LDR. **Datasheet**. [S.l.: s.n.], 2021. Disponível em: <https://www.sunrom.com/p/light-sensing-module-ldr>. Acesso em: 12 de fev. 2021.
- MOD.RELE.3V.4CH. **Datasheet**. [S.l.: s.n.], 2021. Disponível em: <https://www.masterwalkershop.com.br/modulo-rele-3v-33v-4-canais>. Acesso em: 12 de fev. 2021.

- NICAWÉ. *Medidores de sensor capacitivo e resistivo: entenda a diferença*. [S.l.: s.n.], 2021. Disponível em: <https://www.nicawe.com.br/medidores-de-sensor-capacitivo-e-resistivo-entenda-a-diferenca/>. Acesso em: 12 de fev. 2021.
- NODEMCU-32S. *Datasheet*. [S.l.: s.n.], 2021. Disponível em: <https://blogmasterwalkershop.com.br/arquivos/datasheet/Datasheet%20ESP-WROOM-32.pdf>. Acesso em: 12 de fev. 2021.
- NODEMCU-8266. *Datasheet*. [S.l.: s.n.], 2021. Disponível em: https://component.s101.com/sites/default/files/component_datasheet/ESP8266-NodeMCU-Datasheet.pdf. Acesso em: 12 de fev. 2021.
- PILOTI, J. S. *Sistema de Automação Residencial: Acessibilidade no Controle Doméstico*. 2014. f. 96. Trabalho de Conclusão de Curso (Bacharel em Tecnologias Digitais) – Centro de Ciências Exatas e das Tecnologias, Universidade de Caxias do Sul, Caxias do Sul, 2014.
- QUANTASMARTHOME. *Site Oficial*. [S.l.: s.n.], 2021. Disponível em: https://quantasmarthome.com.br/?gclid=Cj0KCQiAip-PBhDVARIsAPP2xcltXPqwQ5nG9uGe2NQ_VXeI1RQRNET28HWJiN0_wWsiCdctYICxmsUaApm_EALw_wcB. Acesso em: 26 de jan. 2021.
- RAINBIRD. *Site Oficial*. [S.l.: s.n.], 2021. Disponível em: <https://www.rainbird.com.br/>. Acesso em: 26 de jan. 2021.
- SEN0193. *Datasheet*. [S.l.: s.n.], 2021. Disponível em: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SEN0193_Web.pdf. Acesso em: 12 de jan. 2021.
- TEIXEIRA, E. A. S. *Arquitetura IoT Para Aplicação em Automação Residencial*. 2019. f. 34. Trabalho de Conclusão de Curso (Bacharel em Engenharia de Controle e Automação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2019.
- WANZELER, T.; FULBER, H.; MERLIN, B. **Desenvolvimento de um sistema de automação residencial de baixo custo aliado ao conceito de Internet das Coisas (IoT)**. XXXIV *Simpósio Brasileiro de Telecomunicações*. Santarém, PA, p. 40–44, 2016. Anais eletrônicos. Disponível em: <https://biblioteca.sbtt.org.br>. Acesso em: 23 de fev. 2021.
- YUAN, M. *Conhecendo o MQTT: Por que o MQTT é um dos melhores protocolos de rede para a Internet das Coisas?* [S.l.: s.n.], 2017. Disponível em: <https://developer.ibm.com/br/articles/iot-mqtt-why-good-for-iot/#:~:text=de%20software%20livre.-,Por%20que%20o%20MQTT,limitada%20e%20de%20alta%20lat%C3%Aancia>. Acesso em: 06 de mar. 2022.
- ZAPIER. *Site Oficial*. [S.l.: s.n.], 2021. Disponível em: <https://zapier.com/>. Acesso em: 03 de fev. 2021.
- ZOLETT, D.; RAMIREZ, A. R. G. **Desenvolvimento de uma Interface de Monitoração Remota para o Sistema Robótico ROBIX, Integrando o Protocolo MQTT e oROS**. *Anais do Computer on the Beach*, v. 11, n. 1, p. 405–412, 2020. Anais eletrônicos. Disponível em: bit.ly/3LlOhBg. Acesso em: 03 de fev. 2021.