



INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E
TECNOLOGIA DO AMAZONAS
CAMPUS MANAUS DISTRITO INDUSTRIAL
ENGENHARIA DE CONTROLE E AUTOMAÇÃO



JANDERSON MIKE DO NASCIMENTO JATÍ

SISTEMA PARA O CULTIVO DE PLANTAS UTILIZANDO PRINCÍPIOS
DE INTERNET DAS COISAS

MANAUS - AM
2023

JANDERSON MIKE DO NASCIMENTO JATÍ

SISTEMA PARA O CULTIVO DE PLANTAS UTILIZANDO PRINCÍPIOS
DE INTERNET DAS COISAS

Trabalho de Conclusão de Curso
apresentado como requisito parcial à
obtenção do título de Bacharel em
Engenharia de Controle e Automação, do
Instituto Federal de Educação, Ciência e
Tecnologia do Amazonas, Campus
Manaus Distrito Industrial – IFAM/CMDI.

Orientador: Prof. Dr. Alyson de Jesus
dos Santos

MANAUS - AM

2023

Dados Internacionais de Catalogação na Publicação (CIP)

J11s	<p>Jatí, Janderson Mike do Nascimento.</p> <p>Sistema para cultivo de plantas utilizando princípios de Internet das coisas / Janderson Mike do Nascimento Jatí. — Manaus, 2023. 57f.: il. color.</p> <p>Monografia (Graduação) — Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, <i>Campus</i> Manaus Distrito Industrial, Curso de Engenharia de Controle e Automação, 2023. Orientador: Prof.º Alyson de Jesus dos Santos, Dr.</p> <p>1. Sistemas embarcados. 2. Sensores. 3. Atuadores. 4. Protocolo MQTT. I. Santos, Alyson de Jesus dos. II. Instituto Federal de Educação, Ciência e Tecnologia do Amazonas. III. Título.</p> <p>CDD 629.89</p>
------	--

Elaborada por Oziane Romualdo de Souza (CRB11/ nº 734)

ANEXO 7

ATA DE DEFESA PÚBLICA DO TRABALHO DE CONCLUSÃO DE CURSO

Aos 24 dias do mês de Maio, de 2023, às 13h, o(a) discente Janderson Mike do Nascimento Jatí apresentou o seu Trabalho de Conclusão de Curso para avaliação da Banca Examinadora constituída pelos seguintes integrantes: Prof(a). Dr. Alyson de Jesus dos Santos (docente-orientador), Prof(a). Msc. Michella Fialho (Membro 1) e Prof(a). Gabriel Pinheiro Compto (Membro 2). A sessão publica de defesa foi aberta pelo(a) presidente da banca, que apresentou a Banca Examinadora e deu continuidade aos trabalhos, fazendo uma breve referência ao TCC, que tem como título Sistema para o cultivo de plantas utilizando princípios de internet das coisas. Na sequência, o(a) discente teve até 30 minutos para a comunicação oral de seu trabalho. Cada integrante da banca examinadora fez suas arguições após a defesa do mesmo. Ouvidas as explicações do(a) discente, a banca examinadora, reunida em caráter sigiloso, para proceder à avaliação final, deliberou e decidiu pela APROVAÇÃO com média final 9,2 (NOVE, DOIS) do referido trabalho.

Foi dada ciência ao(à) discente que a versão final do trabalho deverá ser entregue até o dia 23/06/2023, com as devidas alterações sugeridas pela banca. Nada mais havendo a tratar, a sessão foi encerrada às 14h15 min, sendo lavrada a presente ata, que, uma vez aprovada, foi assinada por todos os membros da Banca Examinadora e pelo(a) discente.

Prof.(a) Orientador(a)/Presidente: Alyson de Jesus dos Santos
Prof.(a) Avaliador 1: Michella Socorro Grace Fialho
Prof.(a) Avaliador 2: Gabriel Pinheiro Compto
Discente: Janderson Mike do N. Jatí

AGRADECIMENTOS

Primeiramente a Deus que me guiou durante toda a minha jornada, me dando força e sabedoria.

Agradeço ao meu pai Andersson e minha mãe Merelucia que sempre me ajudaram e estiveram presentes, me dando forças para continuar principalmente nos momentos mais difíceis e a superar cada obstáculo.

Agradeço aos meus amigos de turma Giovane, Hylbert, Ícaro, Lucas, Misael, Pedro Neto, Rebecca, Yure entre outros, que participaram deste percurso, sempre me auxiliando nas dificuldades.

Agradeço ao meu orientador Prof. Dr. Alyson de Jesus dos Santos, pela paciência e sabedoria com que me orientou, sempre presente e disposto a me ajudar.

Por fim, ao IFAM que foi essencial para o meu processo de formação e por tudo que aprendi durante o curso.

“Se só fizer o que sabe, nunca será mais do que é agora.”

(Shifu)

RESUMO

Neste trabalho será desenvolvido um sistema para o cultivo de plantas utilizando princípios de internet das coisas para um melhor cuidado das plantas locais. O sistema atuará quando o mesmo identificar a necessidade de alterar um estado físico da planta, mantendo o ambiente estável para o crescimento do mesmo. A programação do sistema será feita na IDE do Arduino, com a adição de algumas bibliotecas para um funcionamento mais estável e completo dos componentes. Nesse sistema, serão utilizados sensores e atuadores e será possível acompanhar os valores de temperatura e umidade que serão transmitidos via o protocolo de comunicação MQTT para uma aplicação mobile.

Palavras-chave: MQTT, Sensores, Atuadores, Sistemas embarcados.

ABSTRACT

In this work, a system will be developed using Internet of Things principles for better care of local plants. The system will act when it identifies the need to change the physical state of the plant, maintaining a stable environment for its growth. System programming will be performed in the Arduino IDE, with the addition of some libraries for a more stable and complete functioning of the components. In this system, sensors and actuators will be used and it will be possible to monitor the temperature and humidity values that will be transmitted via the MQTT communication protocol to a mobile application.

Keywords: MQTT, Sensors Actuators, IoT, Embedded systems.

LISTA DE ILUSTRAÇÕES

Figura 1 – Kevin Ashton, criador do termo Internet das Coisas	17
Figura 2 – Possibilidades da Automação Residencial	19
Figura 3 – Esquema de funcionamento do protocolo MQTT	22
Figura 4 – Algumas Placas de Desenvolvimento presente no mercado	24
Figura 5 – Irrigação superficial.	26
Figura 6 – Irrigação por aspersão	27
Figura 7 – Irrigação localizada.....	28
Figura 8 – Materiais Utilizados	32
Figura 9 – ESP32 WROOM-32	33
Figura 10 – Sensor de Umidade de Solo	34
Figura 11 – Sensor LDR.....	35
Figura 12 – Sensor DHT11	36
Figura 13 – LM2596	37
Figura 14 – Módulo Relé	38
Figura 15 – Bomba	39
Figura 16 – Iluminação do Plantário.....	40
Figura 17 – Funcionamento do Sistema: Primeira Parte	41
Figura 18 – Funcionamento do Sistema: Segunda Parte	42
Figura 19 – Ambiente para o vaso de planta	43
Figura 20 – Variáveis GPIO e tópicos MQTT	45
Figura 21 – Variáveis auxiliares e objetos da classe WiFi.....	46
Figura 22 – Função Setup.....	47
Figura 23 – Função loop.....	48
Figura 24 – Ícone do MQTT <i>dash</i>	49
Figura 25 – Tela Inicial do aplicativo MQTT <i>dash</i>	50
Figura 27 – Adicionando um nome para o <i>dashboard</i> e o endereço do <i>broker</i> público.....	50
Figura 28 – Tipos de blocos disponíveis	51
Figura 29 – <i>Dashboard</i> finalizado	53
Figura 30 – Esquema do circuito elétrico.....	55
Figura 31 – Protótipo em funcionamento	56
Figura 32 – Valores capturados.....	57

LISTA DE TABELAS

Tabela 1 – Lista de materiais.....	31
Tabela 2 – Entradas utilizadas do GPIO	54
Tabela 3 – Perfil das plantas	55

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas
BLE	<i>Bluetooth Low Energy</i>
GPIO	<i>General Purpose Input/Output</i>
IoT	Internet das Coisas
LDR	<i>Light-Dependent Resistor</i>
MQTT	Message Queuing Telemetry Transport
PC	Computador Pessoal (<i>Personal Computer</i>)
PDF	<i>Portable Document Format</i>
QoS	<i>Quality of service</i>
RFID	<i>Radio-Frequency Identification</i>
RSSF	Redes de Sensores Sem Fio
SSL	<i>Socket Layer</i>
TCC	Trabalho de Conclusão de Curso

SUMÁRIO

1	INTRODUÇÃO	13
2	REFERENCIAL TEÓRICO	15
2.1	INTERNET DAS COISAS	15
2.1.1	Perspectiva Histórica	15
2.1.2	Conceito de IoT	17
2.2	DOMÓTICA	17
2.2.1	História da Automação Residencial	17
2.2.2	Definição de Robótica	19
2.3	SENSORES E ATUADORES	19
2.4	PROTOCOLO MQTT	20
2.4.1	Tópico da Mensagem	20
2.4.2	<i>Subscriber e Publisher</i>	20
2.4.3	Broker	21
2.5	ANDROID.....	22
2.6	PLACAS DE DESENVOLVIMENTO	22
2.7	CULTIVO <i>INDOOR</i> DE PLANTAS.....	24
2.7.1	Tipos de irrigação	24
2.7.1.1	Irrigação superficial	24
2.7.1.2	Irrigação por aspersão	25
2.7.1.3	Irrigação localizada	26
2.7.2	Iluminação <i>indoor</i>	27
3	MATERIAIS E MÉTODOS	29
3.1	METODOLOGIA	29
3.2	MATERIAIS UTILIZADOS.....	30
3.2.1	ESP32 WROOM-32 WiFi	31
3.2.2	Higrômetro	32
3.2.3	Sensor LDR	33
3.2.4	Sensor DHT11	34
3.2.5	LM2596	35
3.2.6	Módulo Relé	36
3.2.7	Mini Bomba água	37

3.2.8	LED	38
4	RESULTADOS E DISCUSSÕES	40
4.1	FLUXOGRAMA DE FUNCIONAMENTO DO SISTEMA.....	40
4.2	LOCAL DA IMPLEMENTAÇÃO DO PROTÓTIPO.....	41
4.3	DESENVOLVIMENTO DO SISTEMA E IMPLEMENTAÇÃO DO CÓDIGO.....	42
4.3.1	Declaração das variáveis e bibliotecas utilizadas	43
4.3.2	Função <i>Setup</i>.....	45
4.3.3	Função <i>Loop</i>.....	46
4.4	APLICATIVO MQTT <i>DASH</i>	47
4.5	MONTAGEM DO <i>HARDWARE</i> DO SISTEMA	53
4.6	SISTEMA DE IRRIGAÇÃO EM FUNCIONAMENTO	55
5	CONSIDERAÇÕES FINAIS.....	57
	REFERÊNCIAS.....	58
	APÊNDICE.....	60

1 INTRODUÇÃO

Conforme o modelo internet das coisas cresce, o número de empresas que oferecem sistemas para facilitar a conexão dos objetos por meio da rede também aumenta. A automação industrial é uma das áreas que mais utilizam do esquema de conexão de objetos em rede e criar um sistema de jardinagem está totalmente ligado à domótica.

Este projeto de pesquisa delimitou-se em fazer a prototipagem de um plantário capaz de medir a umidade do solo, irrigar as plantas e controlar a luminosidade automaticamente através de uma aplicação mobile.

Para cultivar uma planta, alguns fatores como umidade, espaço e incidência solar devem ser levados em consideração. Existem muitas pessoas que gostam de plantas em suas casas, seja para decoração ou consumo, porém algumas delas não possuem um ambiente adequado ou estão ocupadas o dia inteiro, fazendo com que o cultivo dessas plantas se torne uma atividade inviável por requerer maior atenção. Assim, a pergunta problema que norteará a pesquisa é como a automatização das variáveis de umidade, temperatura e incidência solar pode auxiliar no cultivo das plantas para essas pessoas?

Com a criação de um sistema que verifica os valores como umidade e incidência solar, é possível monitorar as plantas e, se preciso, intervir de acordo com a necessidade das mesmas. O sistema é composto de sensores que serão responsáveis pelo monitoramento dos valores, com uma grande autonomia e sem a necessidade de estar conectado em uma fonte hídrica a todo o momento.

O objetivo deste trabalho é desenvolver um dispositivo capaz de monitorar a umidade do solo e a incidência solar na planta. Com essas informações, o sistema irá atuar controlando esses valores para mantê-la saudável. Deste objetivo geral, pode-se considerar para este trabalho 5 objetivos específicos, são eles:

- a) Aprofundar os conceitos teóricos adquiridos durante o curso;
- b) Investigar de que forma automatização pode monitorar e controlar a umidade e incidência solar, valores importantes para manter a saúde da planta;
- c) Organizar e adquirir os materiais necessários para a montagem do dispositivo;
- d) Realizar a montagem do sistema e demonstrar sua funcionalidade;

- e) Monitorar os valores de temperatura e umidade através da aplicação MQTT *Dash*;
- f) Estudar os resultados da pesquisa.

A popularização de objetos conectados à internet está crescendo cada vez mais, por isso a criação de um jardim inteligente para plantas de pequeno porte é algo viável e relevante para as pessoas que querem cultivar algum tipo de planta e não pode, seja por falta de espaço adequado ou falta de tempo. Este trabalho tem sua relevância pois abrange conceitos de domótica utilizando os princípios de IoT, além dos próprios conceitos adquiridos durante o curso de Engenharia de Controle e Automação.

O trabalho está organizado da seguinte forma: O primeiro, essa própria introdução, onde estão apresentadas as características da pesquisa e uma breve explicação do projeto, assim como problemática, objetivos e justificativa. No segundo capítulo está a fundamentação teórica, onde é apresentado uma revisão bibliográfica do tema proposto. No terceiro capítulo será apresentado a metodologia aplicada para o desenvolvimento do sistema. O quarto capítulo apresentará os resultados obtidos e discussão desses resultados, seguindo do quinto capítulo onde estará presente as considerações finais e propostas para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta toda a base de informações teóricas e pesquisa bibliográfica que foi realizada para a melhor compreensão do TCC. Portanto, o foco deste capítulo é apresentar e definir termos importantes para o conhecimento do leitor e apresentar um pouco do impacto que a automação pode trazer para o cenário ambiental.

2.1 INTERNET DAS COISAS

Nesta seção, será abordado o conceito e uma perspectiva histórica sobre a Internet das Coisas.

2.1.1 Perspectiva Histórica

A Internet das Coisas surgiu dos avanços em vários campos, incluindo sistemas embarcados, microeletrônica, comunicações e sensores. De fato, a IoT tem recebido muita atenção tanto da academia quanto da indústria porque pode ser utilizada nas mais diversas áreas da atividade humana.

Kevin Ashton (2009) afirma que o termo “Internet das Coisas” foi apresentado pela primeira vez em seu artigo "*I made at Procter and Gamble*" em 1999. Naquela época, a IoT estava associada ao uso da tecnologia RFID (*Radio-Frequency Identification*), um método de identificação por radiofrequência que identifica, recupera e armazena dados um procedimento de identificação automática Sinais de rádio.

Figura 1 – Kevin Ashton, criador do termo Internet das Coisas



Fonte:(ASHTON, 2009)

Após uma extensa pesquisa sobre Redes de Sensores Sem Fio (RSSF) por volta de 2005, a relação entre IoT e redes de sensores sem fio tornou-se clara, em grande demanda, inovador nos setores doméstico e privado Indústria.

Se tivéssemos computadores que soubessem tudo o que há para saber sobre coisas diferentes usando dados que coletados, sem intervenção humana, poderíamos monitorar e medir tudo, reduzindo desperdícios, desperdícios e custos. Gostamos de saber quando as coisas devem ser substituídas, reparo ou atualização e se são novos ou obsoletos (ASHTON, 2009).

Esse pensamento de Ashton nos permite ver o que realmente mudou em nossas vidas ao longo dos anos graças aos avanços da tecnologia foi motivado por tais considerações e pesquisas.

No Brasil esse tipo de tecnologia já é uma realidade e faz parte do dia a dia dos brasileiros, televisões do tipo *Smart TV* que se conectam a internet e podem acessar aplicativos como *YouTube* e *Netflix* podem ser encontradas em muitos lares. Jogos que conectam pessoas do mundo inteiro através da internet e até mesmo câmeras de segurança que podem ser acessadas via *smartphones* são exemplos de como a Internet das Coisas está presente na vida dos brasileiros.

2.1.2 Conceito de IoT

Para Carrion (2019), Internet das Coisas é um ecossistema que conecta objetos físicos através de uma rede de comunicação com o objetivo de trocar, coletar e armazenar dados utilizando uma aplicação de *software*.

Para o mesmo autor, o desenvolvimento de aplicações para a IoT cria soluções benéficas ao público alvo e um desses grandes benefícios está em transformar objetos em dispositivos inteligentes, objetos que não possuem qualquer tipo de inteligência passam a fornecer dados e responder a comandos externos, com a finalidade de aperfeiçoar o funcionamento.

Um exemplo interessante que explica como esse tipo de tecnologia impacta na sociedade atual: Uma balança tradicional de banheiro basicamente calcula o peso e o exibe para o usuário, se essa pessoa tenha o desejo de acompanhar os valores do seu peso, ele fará por conta própria. Com a implementação do IoT, a balança pode fornecer mais valores e conectado a um serviço que guarde essas informações, o usuário poderá acompanhar a evolução do seu peso e a balança será capaz de armazenar valores de até mais de um usuário (DURAES; FERREIRA; MANZAN, 2022).

2.2 DOMÓTICA

2.2.1 História da automação residencial

Para Sena (2005), a automação residencial teve origem a partir dos conceitos utilizados em automação industrial. Esses dois mercados se diferem, pois, na automação industrial é necessário uma alta precisão e respostas rápidas, já na automação residencial o foco está em construir interfaces cada vez mais intuitiva e acessível a todos, um produto com acabamento superior.

Durante a década de 70 foram lançados os primeiros módulos inteligentes de automação chamados X-10, esses módulos permitem a comunicação via rede elétrica (removendo a necessidade de novos cabamentos) de produtos compatíveis com a tecnologia.

Os sistemas de automação residencial foram amplamente expandidos durante a década de 90, conforme o uso de computadores pessoais se normalizava e outras tecnologias se difundiam, como a internet e o telefone celular.

Figura 2 – Possibilidades da Automação Residencial



Fonte:(SENA, 2005)

2.2.2 Definição de robótica

A palavra “Domótica” resulta de dois termos, *Domus* (palavra latina que significa casa) e Robótica (ANGEL; FRAIGI, 1993).

Também para Angel (1993), domótica é um domínio de aplicação tecnológica que possui como objetivo melhorar a qualidade de vida das pessoas, reduzir o trabalho doméstico além de aumentar a segurança e a qualidade de vida. Outras denominações para a domótica podem ser citadas, como “Casa Inteligente” e “Ambiente Inteligente”.

Os primeiros sistemas domóticos basicamente utilizavam sensores e atuadores ligados a um controlador com o objetivo de controlar a iluminação, melhorar a segurança ou acompanhar as condições climáticas. Atualmente o objetivo permanece o mesmo, mas voltado para um contexto doméstico e que seja acessível a todas as pessoas (MESSIAS, 2007).

Integrar internet das coisas com a domótica possibilita criar vários sistemas robustos na área da agricultura, umas das possibilidades é programar horários para a realização de atividades como irrigação ou iluminação durante o dia, além de verificar com sensores de luz e umidade se há a necessidade de executar tal atividade ou não (QUIRICONI, 2022).

2.3 SENSORES E ATUADORES

Os sensores são dispositivos de entrada responsáveis por transformar um sinal de determinada grandeza em outro sinal (geralmente elétrico) para que possam ser analisados por sistemas domóticos (BOLZANI; NETTO, 2004).

Para o mesmo autor, atuadores são componentes eletromecânicos que podem ser interligados em dispositivos inteligentes ou a uma rede de dados por meio de pequenas interfaces para que sejam utilizados pelos sistemas domóticos, as características dos atuadores são alteradas de acordo com os impulsos elétricos que recebem.

2.4 PROTOCOLO MQTT

Segundo Ibusuki (2021), O protocolo de comunicação *Message Queuing Telemetry Transport* (MQTT) foi um protocolo inventado por Arlen Nipper e Andy Stanford-Clark com o objetivo de conectar oleodutos por meio de satélites em 1999.

No protocolo MQTT, a publicação e recebimento de dados é realizada por meio de um servidor chamado *broker*. Ou seja, nessa comunicação haverá um cliente, que tem o papel de publicador e transmitirá uma mensagem com um tópico alvo e seu *payload* (conteúdo da mensagem). Em seguida, a mensagem em questão será entregue ao corretor responsável por gerenciá-la, e repassá-la aos assinantes que assinaram o tópico anteriormente.

Da mesma forma, caso um cliente queira se tornar assinante de determinado tema, deverá encaminhar a solicitação ao *broker*, que poderá estabelecer uma conexão entre o cliente e o tema relacionado (SANTOS; IBUSUKI; SCHÜTZER, 2021).

2.4.1 Tópico da Mensagem

O primeiro ponto para entender esse protocolo é entender os canais pois é por ele que a mensagem é enviada. No MQTT, as mensagens são enviadas por meio de tópicos. Outro fato importante é que esses tópicos são canais de mão dupla, ou seja, um cliente pode enviar informações pelo tópico e outro cliente pode receber informações. Esta dupla função irá gerar os *Subscriber* e *Publisher* (SANTOS; IBUSUKI; SCHÜTZER, 2021).

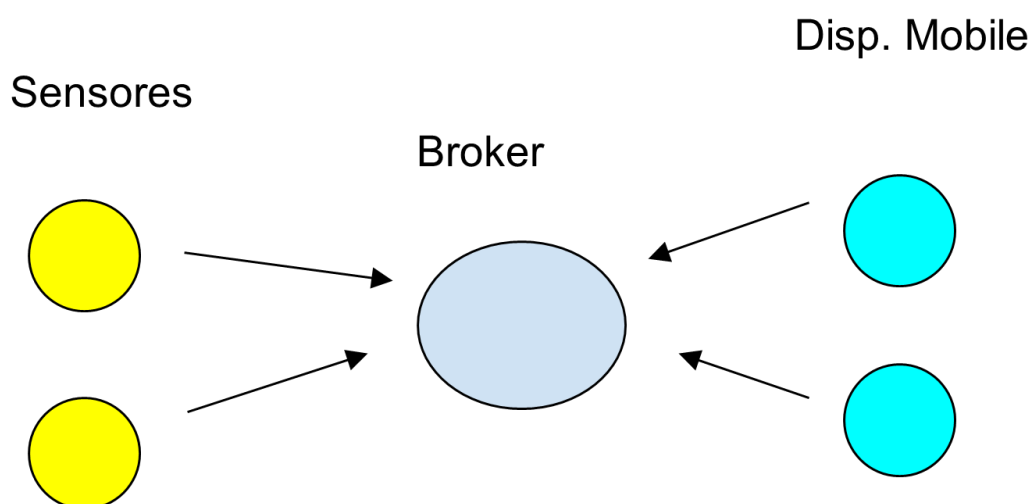
2.4.2 Subscriber e Publisher

Ainda para Santos (2021), o cliente no protocolo MQTT pode ser *Subscriber* ou *Publisher*, quando o cliente é um *Publisher* ele publica informações sobre um determinado tópico de mensagem, uma vez que o cliente é um *Subscriber*, ele é um assinante deste tópico, ou seja, ele recebe informações através do mesmo. Em alguns casos, um mesmo cliente pode ser *Publisher* de um tópico e *Subscriber* em um outro tópico.

2.4.3 Broker

O *Broker* é o principal ponto de operação desse protocolo, ele é o servidor, onde os dados vão chegar e ser enviados pelo protocolo. Assim como um servidor, pode ser local ou online. São várias opções de acordo com as expectativas do projeto.

Figura 3 – Esquema de funcionamento do protocolo MQTT



Fonte: Próprio Autor

Por exemplo, existe o *Mosquitto*, que é um *broker* de código aberto (*Open source*, ou seja, não é preciso pagar para utilizar), que pode ser instalado em um PC ou *Raspberry Pi* localmente, e outra opção é a *CloudMQTT*, que é uma corretora online, mas a desvantagem é que precisa ser paga (SANTOS; IBUSUKI; SCHÜTZER, 2021).

Ainda Inusuki (2021), as vantagens de usar o MQTT no projeto são grandes, pois diferente dos outros protocolos, esse possui um baixo consumo de memória o que influencia no processamento dos dados para o transporte de dados, que também será baixo.

As mensagens que são transmitidas estão protegidas através do uso de certificado *Secure Socket Layer* (SSL) com criptografias avançadas. Além disso, a confiabilidade para garantir que as mensagens sejam entregues é feita através de configurações QoS (*Quality of service*).

2.5 ANDROID

Para Rodrigues (2016), O uso de telefones celulares aumentou significativamente nos últimos anos, inicialmente, estava disponível apenas como um celular muito caro, limitado a pessoas de alto poder aquisitivo. Com o desenvolvimento da tecnologia da informação e da tecnologia eletrônica, os telefones celulares tradicionais foram substituídos por telefones inteligentes, telefones inteligentes com funções avançadas.

Com a intenção de padronizar uma plataforma aberta e celulares gratuitos, para atender às expectativas e tendências do mercado global, um grupo de empresas liderado pela *Open Handset Alliance* desenvolveu a plataforma *Android*. De acordo com a *Open Handset Alliance*, a plataforma Android foi construída desde o início com o objetivo de ser a plataforma aberta, completa e gratuita criada especificamente para dispositivos móveis (RODRIGUES, 2016).

2.6 PLACAS DE DESENVOLVIMENTO

A placa de desenvolvimento é uma plataforma de *hardware* montada com a finalidade de facilitar a realização de projetos, além de facilitar o aprendizado de uma determinada tecnologia. Difundidas na cultura *Maker*, as placas de desenvolvimento são usadas para educação em ciência e tecnologia, projetos de engenharia e computação, prototipagem eletrônica e construção de sistemas personalizados (MADEIRA, 2018).

Figura 4 – Algumas Placas de Desenvolvimento presente no mercado



Fonte: (DORNELES; RIBEIRO, 2016)

O conceito de Indústria 4.0 traz uma série de inovações nas áreas de automação e tecnologia da informação, que são aplicadas principalmente aos processos de fabricação industrial. Portanto, a integração de placas de desenvolvimento como as placas Arduino e *Raspberry* com o conceito de IoT (Internet of Things) pode melhorar todos os processos industriais e aumentar a eficiência em diversos aspectos (DORNELES; RIBEIRO, 2016).

Para o mesmo autor, a chamada indústria inteligente é resultado de grandes revoluções tecnológicas que impulsionam melhorias nos processos de fabricação, utilizando produtos muitas vezes desenvolvidos nas plataformas de desenvolvimento citadas para melhorar as capacidades operacionais por meio da rastreabilidade e monitoramento das atividades.

2.7 CULTIVO *INDOOR* DE PLANTAS

Um sistema de cultivo *indoor* é um método de produção agrícola que ocorre em um ambiente fechado, controlado e artificial, que não depende da luz solar e da chuva. Ele envolve o uso de iluminação artificial, controle de temperatura, umidade, CO₂, nutrientes e irrigação para criar um ambiente de crescimento ideal para as plantas (AGOSTINHO, 2021). A utilização dessa técnica pode trazer diversos benefícios, como maior controle das condições de cultivo e menor exposição a pragas e doenças, além de permitir a produção de alimentos em locais urbanos ou em áreas com condições climáticas desfavoráveis (AGOSTINHO, 2021).

2.7.1 Tipos de irrigação

O desenvolvimento das hortaliças é fortemente influenciado pelas condições de umidade do solo. A falta de água costuma ser o fator mais limitante para a obtenção de altos rendimentos e produtos de qualidade, mas o excesso de água também pode ser danoso. Repor a água do solo por irrigação, em quantidade suficiente e no momento certo, é crucial para o sucesso da jardinagem (MAROUELLI; SILVA; SILVA, 1996).

A escolha do sistema depende de várias condições, como: quantidade e qualidade da água disponível, custo de implantação do sistema, tipo de solo e tipo de planta. Existem vários métodos de irrigação, como irrigação superficial, de aspersão e localizada. (TESTEZLAF, 2011).

2.7.1.1 Irrigação superficial

A irrigação de superfície também é conhecida como irrigação por gravidade. Isso ocorre porque os agricultores jogam água sobre o solo e usam os efeitos da gravidade para cobrir todo ou parte do solo e permitir que ele penetre durante esse movimento e após o represamento (TESTEZLAF, 2011).

Figura 5 – Irrigação superficial



Fonte: (AGUIAR, 2021)

Para o mesmo autor, as principais vantagens dessa técnica são o baixo custo de manutenção e implantação do sistema, além do aumento de fotossíntese das plantas em consequência da reflexão da luz na água. No entanto, água estagnada por muito tempo pode prejudicar as plantas, pois reduz a respiração das raízes.

2.7.1.2 Irrigação por aspersão

A aspersão é o método de irrigação que melhor simula a chuva. É aplicado na forma de gotas na superfície das plantas e da terra. A água é bombeada de uma fonte de água (rios, reservatórios, represas, poços, etc.), distribuída por uma rede de tubulações e aspergida (pulverizada) no ar na forma de pequenas gotas. A formação de gotículas é conseguida forçando a água sob pressão através de orifícios em tubulações ou dispositivos mecânicos chamados aspersores ou pulverizadores.

O baixo custo de mão de obra dessa técnica é a grande vantagem, além do melhor controle de irrigação que possibilita em uma elevada eficiência de aplicação. Assim como a irrigação superficial, os declives e desníveis do terreno pode ser uma limitação para o terreno (TESTEZLAF, 2011).

Figura 6 – Irrigação por aspersão



Fonte: (AGUIAR, 2021)

2.7.1.3 Irrigação localizada

Este método de irrigação é baseado no princípio de aplicação "local" de água. Ou seja, ao contrário dos outros métodos de irrigação, a água é aplicada mais perto das raízes da planta, para que possa ser utilizada de forma mais eficaz. A irrigação local é caracterizada por frequentes e pequenas quantidades de água (TESTEZLAF, 2011).

Figura 7 – Irrigação localizada



Fonte: (AGUIAR, 2021)

Segundo Testezlaf (2011), essa técnica mantém o solo úmido e com oxigênio por mais tempo, além de reduzir o consumo de água. Declives e desníveis no terreno não é uma desvantagem desse sistema, mas deve-se tomar cuidado com a constante disponibilidade de água que a planta receberá pois isso pode afetar as raízes, diminuindo a profundidade das raízes que conseqüentemente diminui a estabilidade da planta.

2.7.2 Iluminação *indoor*

A iluminação *indoor* para cultivo de plantas é um tema importante para aqueles que desejam cultivar plantas em ambientes fechados, como em estufas ou em sistemas hidropônicos *indoor*. A escolha adequada do tipo de iluminação pode afetar significativamente o crescimento e a produtividade das plantas.

Um estudo realizado por Cavalcante (2019) avaliou o desempenho do cultivo de alface em um sistema hidropônico utilizando diferentes fontes de luz, incluindo LED, CFL, HPS e MH. Os resultados indicaram que o uso de lâmpadas LED foi o mais eficiente em termos de produtividade e economia de energia. Além disso, as plantas cultivadas sob luz LED apresentaram maiores teores de clorofila e carotenoides em comparação com as plantas cultivadas sob as outras fontes de luz.

3 MATERIAIS E MÉTODOS

Neste tópico será fornecido informações sobre os aspectos do estudo, incluindo a metodologia aplicada nesta pesquisa e a natureza da pesquisa. Além disso, será apresentado também os detalhes sobre os equipamentos e instrumentos utilizados.

3.1 METODOLOGIA

Esse trabalho irá abordar o processo de criação de um sistema automático para o cultivo de plantas na esfera da Internet das Coisas, iniciando na escolha do ambiente de simulação do projeto, protocolo de comunicação, desenvolvimento do protótipo simulado e por fim a simulação funcionando, de tal forma que seja fácil o entendimento e que esse trabalho possa servir de inspiração para trabalhos futuros para quem se interessar e querer criar o seu próprio plantário.

A metodologia aplicada neste trabalho é uma pesquisa de abordagem qualitativa, porque o trabalho é voltado para a análise dos valores encontrados e em seguida, definir critérios de acordo com esses valores.

A metodologia é fundamental para garantir a qualidade e a credibilidade dos resultados de uma pesquisa, pois ela orienta a escolha dos métodos e técnicas mais adequados para responder às perguntas de pesquisa e solucionar os problemas propostos. De acordo com Gil (2018), "a metodologia de pesquisa é um conjunto de procedimentos ou etapas a serem vencidos ordenadamente na busca do conhecimento".

Quanto à natureza da pesquisa, se trata de dois tipos de pesquisa, a básica e a aplicada. É básica pois abre espaço para possíveis trabalhos futuros como a adição de mais sensores para ter um maior controle do estado da planta. Ela também pode ser considerada aplicada pois possui a sua parte de desenvolvimento e possível solução para o problema proposto.

Sobre o objetivo, será uma pesquisa explicativa, pois tem como foco explicar o porquê de determinados comportamentos acontecerem, tentar explicar como esses comportamentos funcionam e tentar encontrar uma solução para eles.

Os procedimentos utilizados neste trabalho foram as pesquisas bibliográfica e experimental. Uma pesquisa bibliográfica é importante pois assim é possível usar pesquisas comparativas e exploratórias que ajudam a investigar muitos fatores muito mais amplos. Já a pesquisa experimental é importante pois com ela será possível analisar variáveis e encontrar formas de controlar essas variáveis (neste caso, luminosidade e irrigação).

3.2 MATERIAIS UTILIZADOS

A Tabela 1 apresenta os materiais utilizados para o desenvolvimento do protótipo, todos esses materiais foram adquiridos localmente por conta de maior segurança, garantia e contato com o fornecedor de cada componente.

Tabela 1: Lista de Materiais

Item	Descrição	Quant.
1	ESP32 WROOM-32 WiFi	1
2	Sensor de Umidade de Solo Higrômetro	1
3	Mini Bomba de Água 12V RS-385	1
4	Módulo Relé 4 canais 5V	1
5	Módulo Sensor de Luz LDR	1
6	Sensor de umidade e temperatura DHT11	1
7	Regulador de tensão Step Down LM2596	1
8	Fita LED 5V	1

Fonte: Próprio Autor

Outros materiais como *jumpers*, fontes, conexões e estruturas foram utilizados para a montagem juntamente com os componentes principais.

Figura 8 – Materiais Utilizados.



Fonte: Próprio Autor

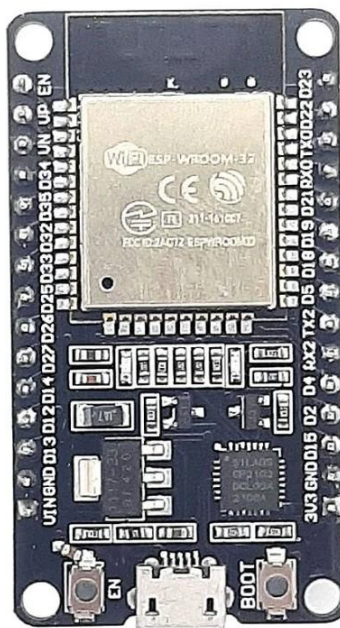
3.2.1 ESP32 WROOM-32WiFi

O ESP32 WROOM-32 é um módulo Wi-Fi de alto desempenho e baixo custo desenvolvido pela empresa *Espressif Systems*. Ele é baseado no chip ESP32, que é um microcontrolador de 32 *bits* com núcleos de processamento *dual-core*, *Wi-Fi* integrado, *Bluetooth 4.2* e uma ampla variedade de periféricos.

Este microcontrolador é amplamente utilizado nos projetos de IoT, devido a sua alta eficiência, baixo consumo de energia e capacidade de se conectar a uma ampla variedade de dispositivos sem fio. Ele será utilizado no projeto pois suporta protocolos de comunicação como *Wi-Fi*, *Bluetooth Classic* e *Bluetooth Low Energy (BLE)*, o que o torna uma excelente escolha para projetos que exigem conectividade sem fio.

Portanto, o ESP32 WROOM-32 é uma ótima opção para o projeto devido à sua alta eficiência energética, acessibilidade conectividade sem fio e ampla gama de periféricos integrados. No projeto, ele é responsável por todo o processo do sistema, com toda a programação feita na IDE do Arduino.

Figura 9 – ESP32 WROOM-32



Fonte: (SMART, 2023)

3.2.2 Higrômetro

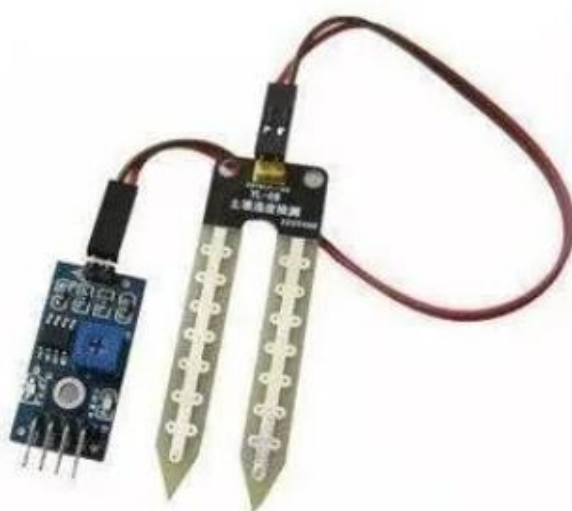
O higrômetro é um sensor de umidade de solo. Ele pode ser utilizado em uma variedade de aplicações, como o monitoramento da umidade do solo em ambientes industriais ou, no cenário do projeto, o controle de temperatura e umidade em estufas para plantas.

Existem muitos tipos de higrômetros disponíveis e cada tipo de higrômetro funciona de maneira diferente, mas todos eles medem a umidade relativa do material com o qual estão em contato.

Então, o higrômetro é um dispositivo importante para medir a umidade solo. Esse sensor pode ser utilizado em várias aplicações para garantir que a umidade do solo esteja dentro do limite estabelecido, para evitar problemas como mofo, danos a planta, deixando as condições salubres para a mesma.

No sistema, a haste será fincada no solo onde ficará a planta enquanto a placa de aquisição de dados será ligada aos pinos do microcontrolador ESP32 WROOM.

Figura 10 – Sensor de Umidade de Solo



Fonte: (SMART, 2023)

3.2.3 Sensor LDR

O LDR (*Light-Dependent Resistor*) é um tipo de sensor que muda sua resistência elétrica de acordo com a variação da intensidade da luz que está incidindo sobre ele. Este sensor é composto por um material semicondutor que se torna mais condutivo quando exposto à luz.

O LDR, também conhecido como fotoresistor, é utilizado em várias aplicações, como em sistemas de iluminação automatizados, câmeras fotográficas e dispositivos de segurança. Este sensor é uma opção popular para medir a intensidade de luz ambiente em uma variedade de aplicações.

Ele é um sensor de baixo custo, o que o torna uma ótima escolha para esse projeto e aplicações industriais simples. Ele será responsável por verificar a integridade e funcionalidade da iluminação do plantário.

Figura 11 – Sensor LDR



Fonte: (SMART, 2023)

3.2.4 Sensor DHT11

O DHT11 é um sensor que mede a umidade relativa e a temperatura do ar em tempo real. Ele é formado por um termistor para medir a temperatura e um sensor capacitivo de umidade.

O DHT11 é um sensor de baixo custo, simples de usar e pode ser facilmente integrado a sistemas de automação residencial ou industrial. Ele é amplamente utilizado em projetos que exigem a medição precisa da temperatura e umidade do ar, como em sistemas de monitoramento ambiental, sistemas de controle de temperatura e umidade, e em projetos de automação de jardins e hortas.

Esse componente será utilizado para monitorar os valores de temperatura e umidade relativa do ar no plantário, oferecendo mais informações do estado da planta.

Figura 12 – Sensor DHT11



Fonte: (SMART, 2023)

3.2.5 LM2596

O LM2596 é um regulador de tensão *step-down*, projetado para converter uma tensão de entrada alta em uma tensão de saída baixa. Este dispositivo é capaz de suportar uma faixa de tensão de entrada de 3V a 40V, útil para projetos que utilizam uma variedade de fontes de alimentação, incluindo baterias, fontes de alimentação AC/DC e outros dispositivos eletrônicos.

O LM2596 é capaz de converter a maior parte da energia de entrada em energia de saída, minimizando a dissipação de calor e aumentando a vida útil do componente. Esse componente é usado em diversos projetos eletrônicos, incluindo alimentação de motores, alimentação de *displays*, fontes de alimentação de sistemas embarcados, entre outros.

Esse componente é uma opção popular para projetos de eletrônica devido à sua eficiência típica de até 92 por cento, facilidade de uso e flexibilidade. No projeto, a fonte externa será ligada no LM2596 que alimentará a ESP32 WROOM e os demais sensores.

Figura 13 – LM2596



Fonte: (SMART, 2023)

3.2.6 Módulo Relé

O módulo relé é um dispositivo eletromecânico que permite controlar um circuito elétrico com uma baixa tensão, como um microcontrolador, por exemplo. Seu funcionamento é simples, ele permite abrir ou fechar um circuito quando um sinal elétrico é aplicado em suas bobinas. Ele é muito utilizado em projetos de automação residencial, industrial e sistemas embarcados.

Os módulos relé podem ser encontrados em circuitos de controle de luzes, ventiladores, fechaduras eletrônicas, portões automáticos, entre outros. Eles também podem ser encontrados na indústria em circuitos criados para controlar motores, válvulas, bombas e outros equipamentos elétricos.

A principal vantagem do módulo relé é a possibilidade de controlar circuitos elétricos de necessitam de alta tensão com um microcontrolador de baixa tensão e no projeto, ele terá essa finalidade.

Figura 14 – Módulo Relé



Fonte: (SMART, 2023)

3.2.7 Mini Bomba de água

A bomba d'água é um dispositivo utilizado para transferir água de um local para outro, ela trabalha convertendo energia mecânica em energia hidráulica para movimentar a água através de uma tubulação.

A escolha da bomba d'água vai depende das necessidades específicas da aplicação, como o volume de água que precisa ser transportado ou o tipo de fonte de energia disponível. Existem muitos modelos e, dependendo da aplicação e das condições de uso, a bomba de água pode ser alimentada por diferentes fontes de energia, como eletricidade, gasolina, diesel e energia solar ou eólica.

Esses dispositivos são amplamente utilizados em sistemas de irrigação agrícola para fornecer água para as plantações, aumentando a produtividade e a eficiência do processo. Para o projeto, foi escolhido uma bomba simples de 12V para o circuito de irrigação.

Figura 15 – Mini Bomba de água



Fonte: (SMART, 2023)

3.2.8 LED

O LED escolhido para esse projeto foi uma fita LED (*Light-Emitting Diode*) simples utilizada na maioria dos casos para decoração, ela é composta por um circuito impresso flexível, com LEDs integrados em sua superfície.

As fitas LED podem ser encontradas em diversos comprimentos e formatos, permitindo uma grande flexibilidade na hora da instalação no projeto. A iluminação é obtida através da passagem de corrente elétrica nos LEDs na superfície da fita, permitindo que emitem luz em vários espectros de cor, no caso do projeto, a fita escolhida emite apenas uma cor.

Outras vantagens da fita LED incluem a alta eficiência energética e a longa vida útil. A fita LED consome muito menos energia em comparação com outras tecnologias de iluminação, além de ter uma vida útil muito maior, o que significa que ela pode ser utilizada por muitos anos antes de precisar ser substituída.

Figura 16 – Iluminação do Plantário



Fonte: (SMART , 2023)

4 RESULTADOS E DISCUSSÕES

Nesse capítulo, serão apresentados os dados obtidos do projeto aplicado, tal como tabelas, figuras e resultados gerais, relacionando-os com teorias e conceitos discutidos anteriormente. A finalidade do trabalho é desenvolver e implantar um sistema para o cultivo automático de plantas de pequeno porte.

4.1 FLUXOGRAMA DE FUNCIONAMENTO DO SISTEMA

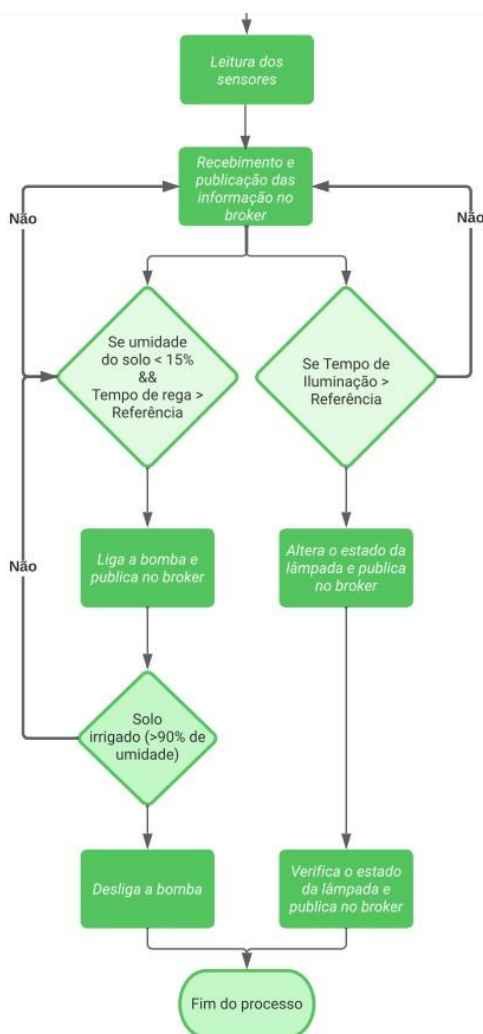
Um fluxograma que explica passo a passo pode ser observado nas Figuras 17 e 18, para simplificar o funcionamento do sistema e explicá-lo de forma mais didática. O fluxograma foi criado pelo próprio autor utilizando o site *Lucidchart*, uma plataforma online que permite criar gráficos, diagramas, fluxos, organogramas e diversos outros processos. É uma ferramenta gratuita muito utilizada que ajuda a entender e desenvolver vários projetos.

Figura 17 – Funcionamento do Sistema: Primeira Parte



Fonte: Próprio Autor

Figura 18 – Funcionamento do Sistema: Segunda Parte



Fonte: Próprio Autor

4.2 LOCAL DA IMPLEMENTAÇÃO DO PROTÓTIPO

O local da implementação do sistema mostrado na Figura 19 foi ajustado para suportar um vaso de pequeno porte, os sensores que irão capturar os valores do ambiente e os atuadores, responsáveis pela manutenção e ajustes dessas variáveis. O ambiente foi projetado para sustentar plantas de pequeno porte como suculentas e cactos decorativos. A construção foi desafiadora pois foi feita a partir de materiais reutilizáveis, então todos os cortes, desbastes, pintura, uniões e conexões foram feitos manualmente.

Figura 19 – Ambiente para o vaso de planta



Fonte: Próprio Autor

4.3 DESENVOLVIMENTO DO SISTEMA E IMPLEMENTAÇÃO DO CÓDIGO

Este tópico aborda o processo de desenvolvimento e implementação do código feito para o funcionamento do protótipo, usado para a leitura dos sensores, transmissão de mensagens via rede e envio de sinais para os atuadores. Como toda a programação foi feita na IDE do Arduino, o código está dividido em três partes: A declaração das variáveis, *setup* e *loop*, com funções sendo utilizadas para mais clareza e organização no código.

4.3.1 Declaração das variáveis e bibliotecas utilizadas

A declaração das variáveis e instalação de bibliotecas é a etapa inicial para o processo de desenvolvimento do código. Primeiramente foi feita uma pesquisa para identificar quais bibliotecas seriam importantes para o projeto, foram selecionadas quatro bibliotecas fundamentais:

- *<WiFi.h>* - Essa biblioteca tem as funções necessárias para a conexão do microcontrolador a redes *Wi-Fi*, sendo possível definir credenciais de autenticação para a rede e obter informações sobre o estado da conexão. É uma biblioteca muito utilizada em projetos de IoT que necessitam de comunicação sem fio com a internet ou com outros dispositivos conectados à rede.
- *<PubSubClient.h>* - Utilizada para a comunicação do microcontrolador com os servidores MQTT, com essa biblioteca é possível publicar e enviar dados nos tópicos para receber mensagens enviadas por outros dispositivos.
- *<string.h>* - Essa biblioteca oferece funções para realizar operações com *strings*, como concatenar, comparar, copiar ou pesquisar caracteres. Ela é importante pois algumas mensagens serão publicadas em *strings*, então a *string.h* facilitará a manipulação dessas *strings*.
- *<DHT.h>* - O uso dessa biblioteca é essencial para o funcionamento do sensor DHT11, pois nela contém as funções para ler os valores de umidade e temperatura do ambiente, além das demais configurações possíveis, como inicializar o sensor ou escolher a pinagem. Ela pode ser utilizada para outros sensores da mesma linha como DHT11, DHT21 e DHT22.

Na Figura 20 está descrito as variáveis conectadas ao GPIO do ESP32, variáveis que representam os sensores, LEDs e atuadores. É importante observar qual é a característica do pino que está utilizando, pois muito deles possuem múltiplas funções e pode causar conflito quanto ao envio de dados ou leitura dos valores dos sensores. Portanto, a escolha dos pinos utilizados foi feita consultando o *datasheet* do microcontrolador para evitar esses conflitos.

Figura 20 – Variáveis GPIO e tópicos MQTT

```
// Variáveis conectadas aos pinos da ESP
#define SoilMoisture      35 //Sensor de umidade do solo
#define LDR                34 //Fotoresistor
#define DHTPIN            25 //Sensor de umidade e temperatura do ambiente
#define PIN_LED_1         14 //Primeiro Perfil
#define PIN_LED_2         27 //Segundo Perfil
#define PIN_LED_3         26 //Terceiro Perfil

#define PIN_BOMBA         2 //Bomba d'água
#define PIN_LAMP          4 //Lâmpada

#define DHTTYPE           DHT11

//Definição dos tópicos publish/subscribe do MQTT
#define TOPICO_SUBSCRIBE_LED      "topico_liga_led_jand"
#define TOPICO_SUBSCRIBE_PLANT_TYPE "tipo_planta_jandmike"
#define TOPICO_PUBLISH_PLANT_SEL  "perf_planta_jandmike"
#define TOPICO_PUBLISH_UMIDADE_SOLO "topic_soil_moisture_jandmike"
#define TOPICO_PUBLISH_LDR        "topic_ldr_sensor_jandmike"
#define TOPICO_PUBLISH_BOMBA      "topico_liga_bomba_jandmike"
#define TOPICO_PUBLISH_LAMPADA    "topico_liga_lampada_jandmike"
#define TOPICO_PUBLISH_TEMPERATURA "topico_sensor_temperatura_jandmike"
#define TOPICO_PUBLISH_UMIDADE    "topico_sensor_umidade_jandmike"

#define ID_MQTT                "esp32_mqtt_jandMike"
```

Fonte: Próprio Autor

Foram definidos também objetos da classe *WiFiCliente* e *PubSubClient* que serão utilizados para realizar a conexão *Wi-Fi* e a conexão com MQTT, respectivamente. Além disso, foram criadas variáveis para auxiliar na programação que estão apresentadas, juntamente com os objetos de *WiFiClient* na Figura 21.

Figura 21 – Variáveis auxiliares e objetos da classe WiFi

```
//Objetos da classe Wifi
const char* SSID = "Jati";
const char* PASSWORD = "a1n14d4e5r18s19s19n14";

const char* BROKER_MQTT = "mqtt.eclipseprojects.io";
int BROKER_PORT = 1883;

WiFiClient espClient; // Cria o objeto espClient
PubSubClient MQTT(espClient);

//Variaveis auxiliares globais
DHT dht(DHTPIN, DHTTYPE);
int timerBomba = 0;
int timerLDR = 0;
int timerLDR_1 = 0;
int tempo_bomba=10000;
int tempo_luz=10000;
int tempo_luzD=0;
unsigned long tempoSensor = 0;

unsigned long tempoBomb = 0;
unsigned long tempoBombL = 0;

bool estadoLED=LOW;
```

Fonte: Próprio Autor

4.3.2 Função Setup

Nesta etapa do programa ocorre a inicialização dos pinos de entrada/saída (GPIO) que serão utilizados, além da comunicação do microcontrolador ESP32 com a rede *Wi-Fi* e o servidor MQTT antes que o programa comece a ser executado. Na Figura 22 é possível observar a inicialização das variáveis e as funções *initWiFi()* e *initMQTT()*, responsáveis pela comunicação com o *Wi-Fi* e MQTT, respectivamente.

Figura 22 – Função Setup

```
// Função de setup
void setup()
{
  Serial.begin(115200);
  initWiFi();
  initMQTT();

  pinMode(SoilMoisture, INPUT);
  pinMode(LDR, INPUT);
  pinMode(PIN_LED_1, OUTPUT);
  pinMode(PIN_LED_2, OUTPUT);
  pinMode(PIN_LED_3, OUTPUT);
  pinMode(PIN_BOMBA, OUTPUT);
  pinMode(PIN_LAMP, OUTPUT);

  digitalWrite(PIN_LED_1, LOW);
  digitalWrite(PIN_LED_2, LOW);
  digitalWrite(PIN_LED_3, LOW);
  digitalWrite(PIN_BOMBA, LOW);
  digitalWrite(PIN_LAMP, LOW);

  dht.begin();
}
```

Fonte: Próprio Autor

4.3.3 Função Loop

Após a inicialização do sistema no *setup()*, na função *loop()* que está o código responsável por controlar as ações do ESP32. Esta parte do código é executada continuamente até que o microcontrolador seja desligado ou receba uma nova compilação.

Pode-se observar na Figura 23 o *loop()*, com as principais funções responsáveis pela leitura dos sensores, controle de irrigação e iluminação, é possível observar também algumas das mensagens que são enviadas via MQTT. As funções apresentadas na imagem abaixo estão descritas no Apêndice A, assim como o programa completo e detalhado.

Figura 23 – Função loop

```

void loop()
{
    //Verificando a estabilidade das conexões
    VerificaConexoesWiFiMQTT();
    MQTT.loop();
    //Enviando os valores para o mqtt
    MQTT.subscribe(TOPICO_SUBSCRIBE_PLANT_TYPE);
    //String enviada para o mqtt
    char umidade_solo_str[10] = {0};
    char temperatura_str[10] = {0};
    char umidade_str[10]     = {0};

    bomba();
    luz();
    ilumina();

    if (millis() - tempoSensor >= 4500){
        //Tranferindo o valor das funções para as strings que serão publicadas
        dtostrf(umidade_solo(), 3, 0,umidade_solo_str);
        dtostrf(faz_leitura_temperatura(), 3, 0,temperatura_str);
        dtostrf(faz_leitura_umidade(), 3, 0,umidade_str);
        //Tópicos publicados no mqtt
        MQTT.publish(TOPICO_PUBLISH_UMIDADE_SOLO, umidade_solo_str);
        MQTT.publish(TOPICO_PUBLISH_TEMPERATURA, temperatura_str);
        MQTT.publish(TOPICO_PUBLISH_UMIDADE, umidade_str);
    }
    delay(500);
}

```

Fonte: Próprio Autor

4.4 APLICATIVO MQTT DASH

Este tópico apresenta a criação do *dashboard* no aplicativo MQTT *dash*, onde as informações de temperatura, umidade, estado de irrigação e iluminação são apresentadas para o usuário. Para essa transmissão de mensagens via protocolo MQTT ocorrer, se faz necessário a utilização de um *Broker MQTT* que permite a troca de informações entre dispositivos via *MQTT*, ele é a ponte de comunicação entre a ESP32 e o aplicativo *MQTT dash*.

Para esse trabalho, foi utilizado um *Broker* MQTT publico disponibilizado pela empresa *Eclipse Foundation*, ele é gratuito e possui todas as *features* necessárias para o objetivo final. Não é necessário configurar o *broker*, é preciso somente do endereço (*mqtt.eclipseprojects.io*).

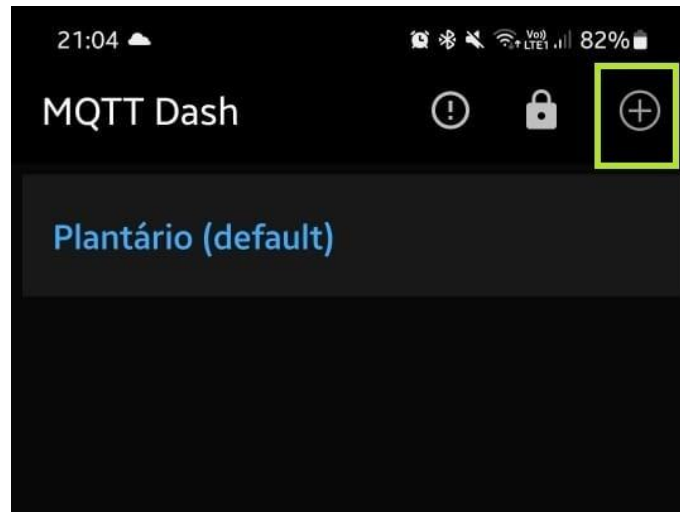
Figura 24 – Ícone do MQTT dash



Fonte: Próprio Autor

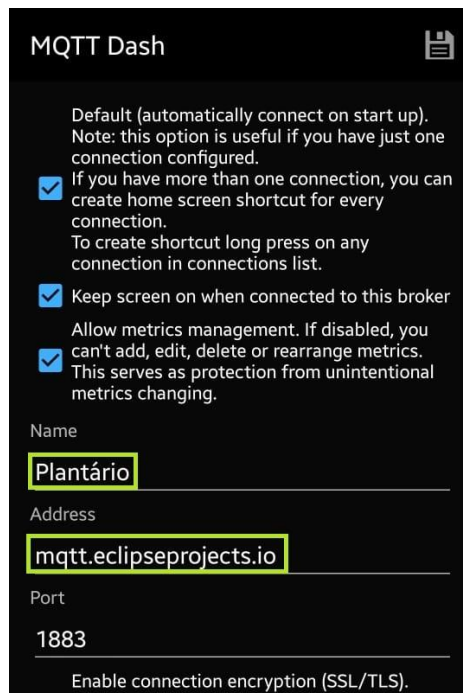
O aplicativo *MQTT dash* pode ser adquirido no *Google Play*, na Figura 24 mostra o ícone do aplicativo. A configuração do *MQTT dash* é simples e intuitivo, o que facilita a criação da UI. Após o download do aplicativo, cria-se um *dashboard* para abrigar todas as informações que serão apresentadas ao usuário. A tela inicial da aplicação está sendo apresentada na Figura 25.

Figura 25 – Tela Inicial do aplicativo MQTT dash



Fonte: Próprio Autor

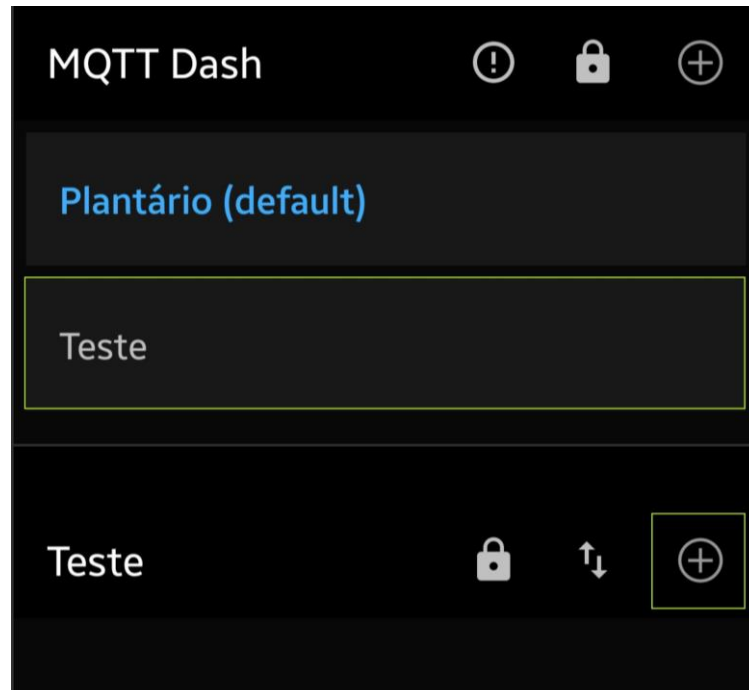
Durante a configuração do *dashboard*, é preciso adicionar o endereço do *broker* como mostrado na Figura 26.

Figura 26 – Adicionando um nome para o *dashboard* e o endereço do *broker*

Fonte: Próprio Autor

Após configurar a conexão e salvar o *dashboard* (tela inicial do *dashboard* criado apresentado na Figura 27) é possível criar os blocos que recebem os tópicos.

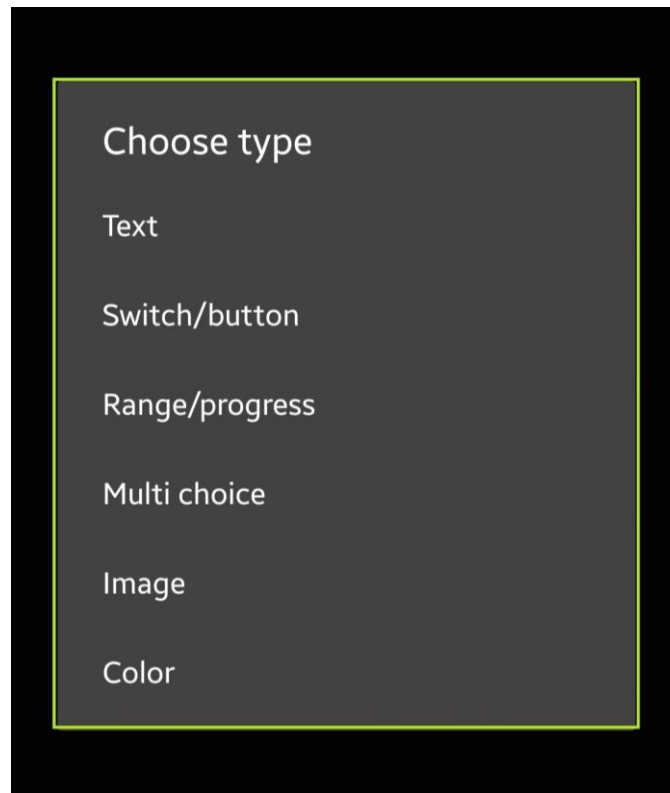
Figura 27 – Primeira tela o *dashboard* criado e segunda tela o botão para criar blocos



Fonte: Próprio Autor

Para o projeto, foi necessário criar oito blocos para apresentar todas as informações, seja as capturadas pelos sensores ou de configuração do sistema (Figura 28). Os blocos escolhidos são de tipos diferentes, três deles são do tipo "color", três do tipo "range/progress", um do tipo "multi-choice" e um do tipo "text".

Figura 28 – Tipos de blocos disponíveis



Fonte: Próprio Autor

- Os blocos do tipo *color* foram criados para representar o estado da bomba (ligado ou desligado), o estado da iluminação (ligado ou desligado) e um para confirmar o estado da iluminação (utilizado para identificar se a iluminação ainda está funcionando).
- Os blocos do tipo *range/progress* apresentam a porcentagem da umidade do solo, umidade relativa do ar e a temperatura.
- Os blocos do tipo *multi-choice* e *text* se complementam no sistema, pois o primeiro é importante para escolher o tipo de planta que está sendo monitorada e o segundo confirma essa escolha para usuário.

A Figura 29 mostra o *dashboard* após a criação dos blocos. Com isso, finaliza-se a etapa de desenvolvimento do sistema e o MQTT *Dash*.

Figura 29 – Dashboard finalizado



Fonte: Próprio Autor

4.5 MONTAGEM DO HARDWARE DO SISTEMA

Nesta seção será apresentado a descrição e montagem do hardware. Inicialmente foi preciso escolher quais pinos da ESP32 serão utilizados, na Tabela 2 está a lista dos pinos escolhidos para a montagem.

É importante observar que, para os sensores, os pinos escolhidos foram do tipo ADC1 (circuito capaz de converter um valor analógico, como a tensão, em uma representação digital) pois os conversores ADC2 são compartilhados com o driver do módulo de comunicação sem fio e não podem ser utilizados enquanto o *Wi-Fi* estiver ativo.

Tabela 2 – Entradas utilizadas do GPIO

GPIO	TIPO	DESCRIÇÃO	TIPO
35	INPUT	Sensor de Umidade do Solo	Analógico
34	INPUT	Fotoresistor	Digital
25	INPUT	Sensor de Umidade e Temperatura	Analógico
14	OUTPUT	Perfil 1	Digital
27	OUTPUT	Perfil 2	Digital
26	OUTPUT	Perfil 3	Digital
2	OUTPUT	Módulo relé - Bomba D'água	Digital
4	OUTPUT	Módulo relé - LED	Digital

Fonte: Próprio Autor

Para o Sensor de umidade de solo (Higrômetro) e o Sensor de umidade e temperatura do ar (DHT11) foram escolhidos os pinos analógicos, o fotoresistor e os LEDs indicadores de perfil estão conectados em pinos digitais. Quatro pinos foram utilizados no relé para o controle da bomba d'água e do LED.

O sistema possui três níveis de tensão em diferentes pontos. Todo o sistema está ligado em uma fonte de 12V, que alimenta o LED, bomba d'água e o regulador de tensão *step down*.

O regulador de tensão, por sua vez, é responsável por alimentar o relé e o ESP32 com uma tensão de 5V. Por fim, os sensores Higrômetro, DHT11 e fotoresistor estão sendo alimentados por uma tensão de 3,3V que é originária do pino do ESP32.

Antes da montagem final, foi feita uma calibração em todos os sensores para que os valores captados sejam precisos. Juntamente com esse ajuste, foi feita a configuração dos perfis das plantas, o tempo de rega e iluminação de cada perfil pode ser visualizado na Tabela 3.

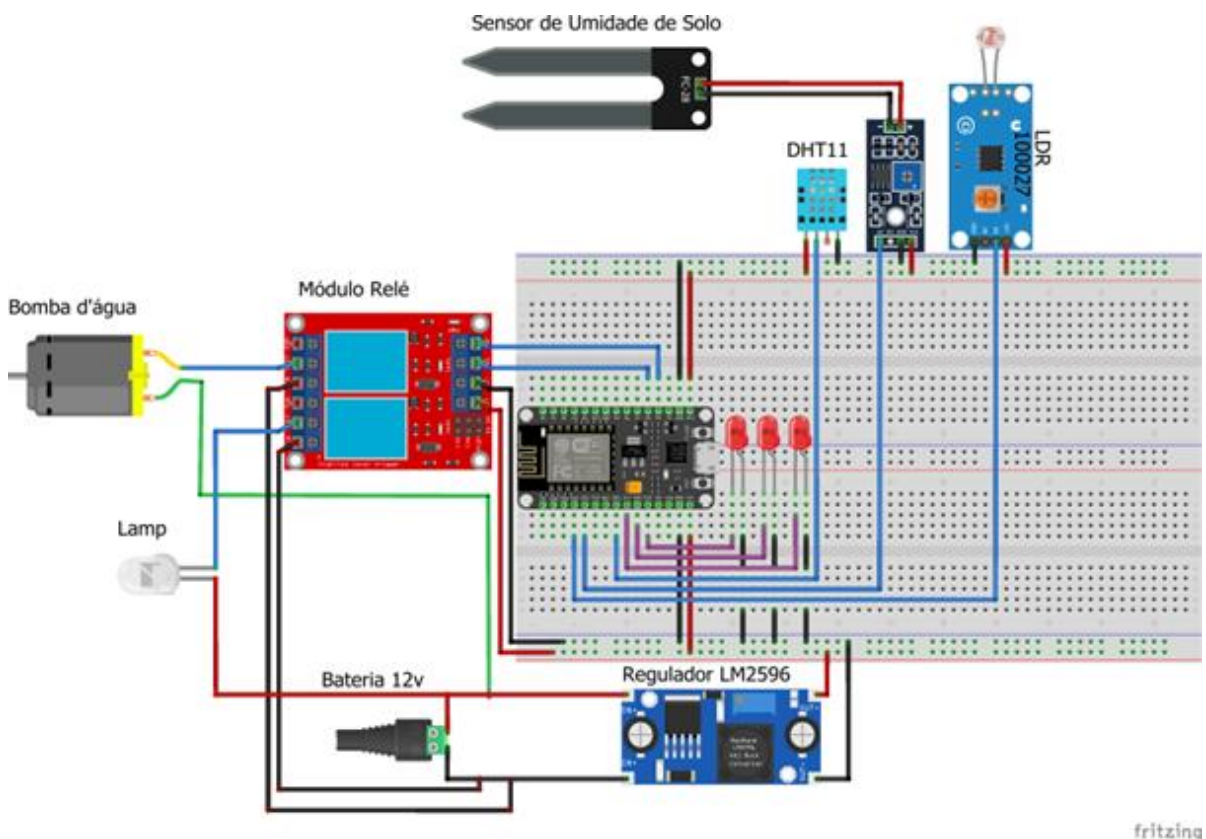
Tabela 3 – Perfil das plantas

Perfil	Planta	Irrigação	Iluminação (hr/dia)
Perfil 1	Suculentas	1 vez por semana	6
Perfil 2	Cacto	1 vez a cada duas semanas	8
Perfil 3	Manjericão	Dias alternados	6

Fonte: Próprio Autor

Na Figura 30 é possível observar o esquema do projeto feito no *fritzing*, uma ferramenta própria para a montagem de sistemas, esse recurso foi utilizado para facilitar a visualização dos componentes e ligações do projeto.

Figura 30 – Esquema do circuito elétrico

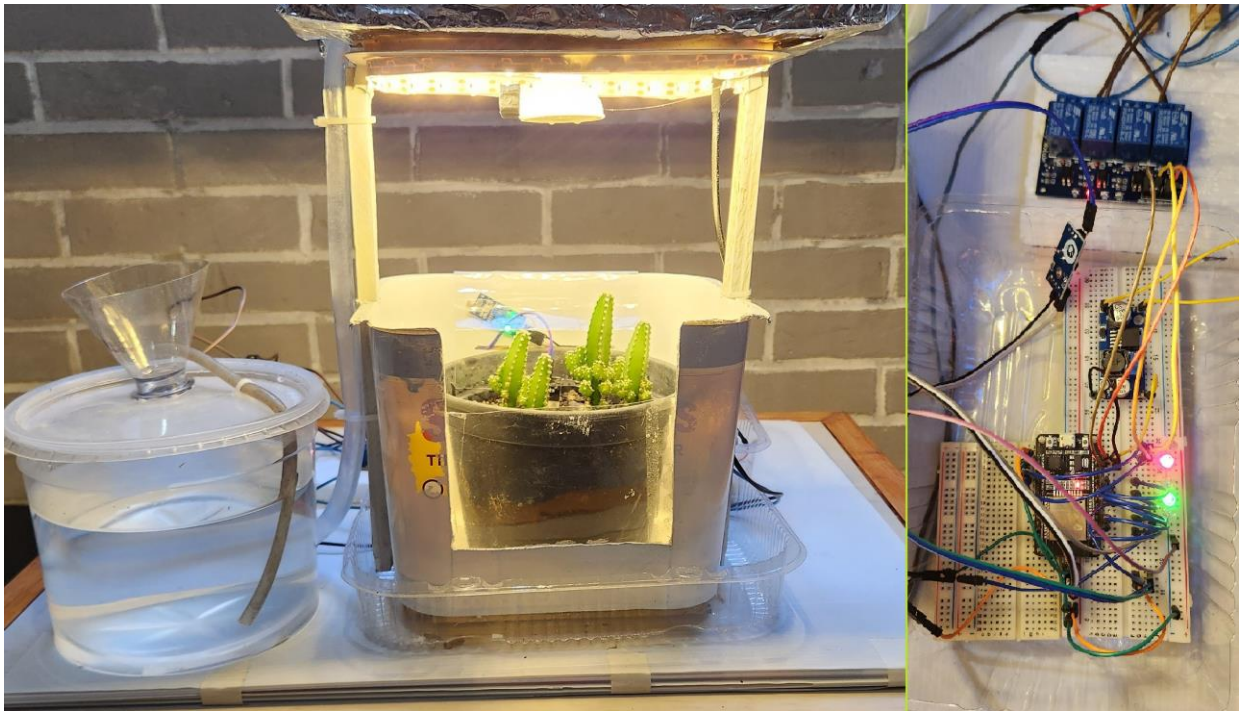


Fonte: Próprio Autor

4.6 SISTEMA DE IRRIGAÇÃO EM FUNCIONAMENTO

A Figura 31 apresenta o sistema em funcionamento com uma planta sendo irrigada, iluminada e monitorada. O sistema possui um reservatório de água de 1,8L, suficiente para manter uma irrigação por 2,5 semanas, além de adicionar uma maior mobilidade do protótipo. A parte elétrica foi posicionada atrás do plantário, dentro de um recipiente de plástico para evitar a umidade ou partículas prejudiciais ao circuito.

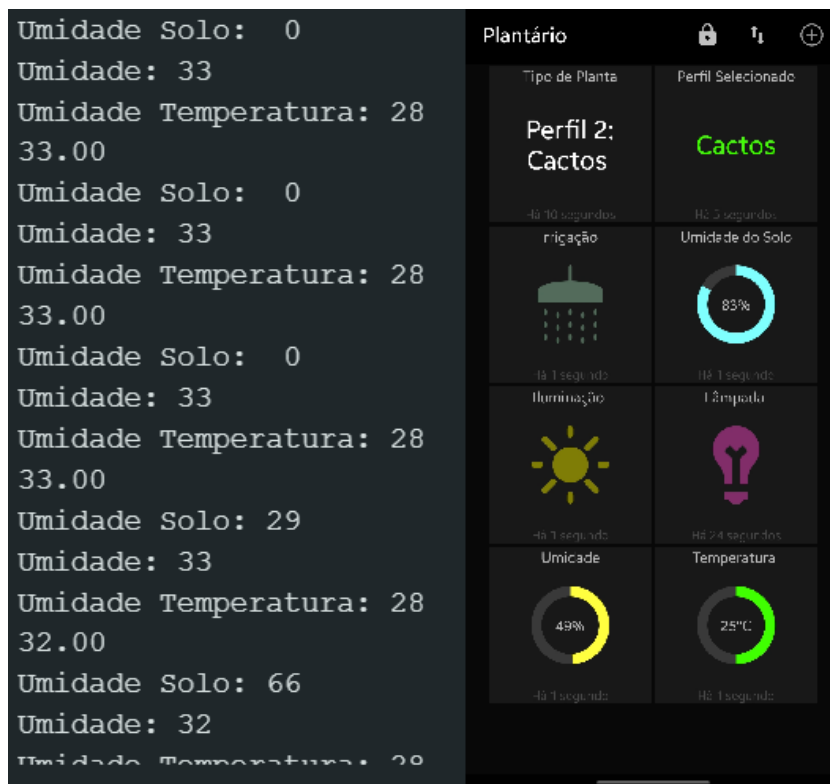
Figura 31 – Protótipo em funcionamento



Fonte: Próprio Autor

A Figura 32 apresenta a parte do software do sistema em funcionamento e com as mensagens sendo entregues no MQTT *dashboard*. Os dados foram capturados durante a fase de testes, enquanto o ESP32 estava conectado no computador e com o monitor serial aberto.

Figura 32 – Valores capturados



Fonte: Próprio Autor

5 CONSIDERAÇÕES FINAIS

A partir dos dados e informações apresentados neste trabalho, conclui-se que o principal objetivo de desenvolver um dispositivo capaz de monitorar a umidade do solo e a iluminação de uma planta foi alcançado. Também foi possível controlar esses valores a partir da automatização desse sistema.

Os materiais utilizados estão disponíveis em lojas de eletrônicos tornando a montagem do sistema muito acessível. Todos os softwares e programas utilizados estão disponíveis na internet e são gratuitos para utilização e com as informações apresentadas é certo de que o sistema pode ser replicado.

Como forma de melhoria e aprimoramento, para futuras pesquisas, a criação de uma estrutura mais dedicada para sustentar a temperatura por mais tempo e posicionar plantas maiores.

A implementação de mais sensores e um LED GROW, iluminação mais apropriada para plantários, também é recomendável para garantir uma planta mais saudável. Por último, criar uma placa PCB para posicionar todo o circuito pode garantir uma melhor manutenção nos componentes, além de deixar o sistema mais resistente, organizado, com menos ruídos e mal-contato.

Portanto, o sistema desenvolvido se mostrou uma ótima ferramenta para o controle da irrigação e iluminação de plantas que se localizam dentro de um ambiente com pouca incidência solar, viabilizando também uma excelente forma de acompanhamento dos valores do ambiente em que essa planta se encontra.

REFERÊNCIAS

- AGOSTINHO, L. C. Plantaê: sistema de cultivo para ambientes internos. *Trabalho de Conclusão de Curso (Graduação em Desenho Industrial-Projeto do Produto)-Escola de Belas Artes, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2021.*
- AGUIAR, V. C. Sistema automatizado de irrigação e monitoramento para plantas em ambientes indoor. Universidade Federal de Uberlândia, 2021.
- ANGEL, P. M.; FRAIGI, L. B. Introducción a la domótica. *Notas. Buenos Aires: Ebay, 1993.*
- ASHTON, K. et al. That ‘internet of things’ thing. *RFID journal*, Hauppauge, New York, v. 22, n. 7, p. 97–114, 2009.
- BOLZANI, C. A. M.; NETTO, M. L. Desenvolvimento de um simulador de controle de dispositivos residenciais inteligentes: uma introdução aos sistemas domóticos. 2004.
- CARRION, P.; QUARESMA, M. Internet da coisas (iot): Definições e aplicabilidade aos usuários finais. *Human Factors in Design*, v. 8, n. 15, p. 049–066, 2019.
- DORNELES, P.; RIBEIRO, J. *Curso de Arduino para Professores*. [S.l.: s.n.], 2016.
- DURAES, W.; FERREIRA, F. H. I. B.; MANZAN, R. *Arquitetura de soluções IoT: Desenvolva com Internet das Coisas para o mundo real*. [S.l.]: Casa do Código, 2022.
- JAVED, S. et al. Utilization of led grow lights for optical wireless communication-based rf-free smart-farming system. *Sensors*, v. 21, n. 20, 2021. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/21/20/6833>.
- MADEIRA, D. *A Revolução das Placas de Desenvolvimento*. 2018. Disponível em: <https://portal.vidadesilicio.com.br/revolucao-das-placas-de-desenvolvimento/>.
- MARQUELLI, W. A.; SILVA, W. d. C. e; SILVA, H. da. *Manejo da irrigação em hortaliças*. [S.l.]: EMBRAPA-SPI Brasília, 1996.
- MESSIAS, A. Edifícios “inteligentes”. *A Domótica aplicada à realidade brasileira. Ouro Preto-MG. Monografia de Graduação*, 2007.
- QUIRICONI, L. Desenvolvimento de um sistema de irrigação residencial automatizado utilizando técnicas de internet das coisas. *Manaus, AM. Monografia de Graduação*, 2022.

SANTOS, O. Gonçalves dos; IBUSUKI, U.; SCHÜTZER, K. Development of the communication with mqtt protocol for industry 4.0. 2021.

SENA, D. C. Automação residencial. *Projeto de Graduação (Graduação em Engenharia Elétrica)–Departamento de Engenharia Elétrica. Espírito Santo: Universidade Federal do Espírito Santo, 2005.*

SMART Projects Brasil. 2023. <https://www.smartprojectsbrasil.com.br/>. Acesso em 26 de março de 2023.

TESTEZLAF, R. Irrigação: métodos, sistemas e aplicações. Faculdade de Engenharia Agrícola Unicamp-FEAGRI, 2011.

APÊNDICE

Apêndice A – Código completo utilizado no Sistema para o cultivo de plantas

```

#include <WiFi.h>
#include <PubSubClient.h>
#include <string.h>
#include <DHT.h>

#define SoilMoisture           35
#define LDR                    34
#define DHTPIN                 25
#define PIN_LED_1              14
#define PIN_LED_2              27
#define PIN_LED_3              26
#define PIN_BOMBA              2
#define PIN_LAMP               4
#define DHTTYPE                DHT11
#define TOPICO_SUBSCRIBE_LED   "topico_liga_led_jand"
#define TOPICO_SUBSCRIBE_PLANT_TYPE "tipo_planta_jandmike"
#define TOPICO_PUBLISH_PLANT_SEL "perf_planta_jandmike"
#define TOPICO_PUBLISH_UMIDADE_SOLO "topic_soil_moisture_jandmike"
#define TOPICO_PUBLISH_LDR      "topic_ldr_sensor_jandmike"
#define TOPICO_PUBLISH_BOMBA    "topico_liga_bomba_jandmike"
#define TOPICO_PUBLISH_LAMPADA  "topico_liga_lampada_jandmike"
#define TOPICO_PUBLISH_TEMPERATURA "topico_sensor_temperatura_jandmike"
#define TOPICO_PUBLISH_UMIDADE  "topico_sensor_umidade_jandmike"
#define ID_MQTT "esp32_mqtt_jandMike"

const char* SSID = "Jati";
const char* PASSWORD = "senha do wifi";

const char* BROKER_MQTT = "mqtt.eclipseprojects.io";
int BROKER_PORT = 1883;

WiFiClient espClient; // Cria o objeto espClient
PubSubClient MQTT(espClient);

//Variaveis globais
DHT dht(DHTPIN, DHTTYPE);
int timerBomba = 0;
int timerLDR = 0;
int timerLDR_1 = 0;
int tempo_bomba=10000;
int tempo_luz=10000;
int tempo_luzD=0;
unsigned long tempoSensor = 0;

```

```

unsigned long tempoBomb = 0;
unsigned long tempoBombL = 0;

bool estadoLED=LOW;

// Funções do Código
void initWiFi(void);
void initMQTT(void);
void mqtt_callback(char* topic, byte* payload, unsigned int length);
void reconnectMQTT(void);
void reconnectWiFi(void);
void VerificaConexoesWiFIEMQTT(void);
void bomba(void);
void ligar_bomba(void);
void desligar_bomba(void);
void ilumina(void);
void ligar_lamp(void);
void desligar_lamp(void);
double umidade_solo(void);
double faz_leitura_temperatura(void);
double faz_leitura_umidade(void);
void perfil_1(void);
void perfil_2(void);
void perfil_3(void);

//Iniciar a conexão com a internet via Wi-Fi
void initWiFi(void)
{
    delay(10);
    Serial.println("*****Conexao WI-FI*****");
    Serial.print("Conectando-se na rede: ");
    Serial.println(SSID);
    Serial.println("Aguarde");

    reconnectWiFi();
}

//Conectar-se ao Broker MQTT
void initMQTT(void)
{
    MQTT.setServer(BROKER_MQTT, BROKER_PORT); //informa qual broker e porta
deve ser conectado
    MQTT.setCallback(mqtt_callback); //atribui função de callback
(funcão chamada quando qualquer informação de um dos tópicos subscritos
chega)
}

//Recebe informações do broker (subscribe)

```

```

void mqtt_callback(char* topic, byte* payload, unsigned int length)
{
    String msg;

    // Obtém a string do payload recebido
    for(int i = 0; i < length; i++)
    {
        char c = (char)payload[i];
        msg += c;
    }

    Serial.print("Chegou a seguinte string: ");
    Serial.println(msg);

    /* toma ação dependendo da string recebida */
    if (msg.equals("1"))
    {
        digitalWrite(PIN_LED_1, HIGH);
        digitalWrite(PIN_LED_2, LOW);
        digitalWrite(PIN_LED_3, LOW);
        Serial.println("Planta 1 selecionada");
        perfil_1();
    }
    if (msg.equals("2"))
    {
        digitalWrite(PIN_LED_1, LOW);
        digitalWrite(PIN_LED_2, HIGH);
        digitalWrite(PIN_LED_3, LOW);
        Serial.println("Planta 2 selecionada");
        perfil_2();
    }
    if (msg.equals("3"))
    {
        digitalWrite(PIN_LED_1, LOW);
        digitalWrite(PIN_LED_2, LOW);
        digitalWrite(PIN_LED_3, HIGH);
        Serial.println("Planta 3 selecionada");
        perfil_3();
    }
}

//Reconexão com o broker
void reconnectMQTT(void)
{
    while (!MQTT.connected())
    {
        Serial.print("* Tentando se conectar ao Broker MQTT: ");
        Serial.println(BROKER_MQTT);
        if (MQTT.connect(ID_MQTT))

```

```

    {
        Serial.println("Conectado com sucesso ao broker MQTT!");
    }
    else
    {
        Serial.println("Falha ao reconectar no broker.");
        Serial.println("Havera nova tentatica de conexao em 2s");
        delay(2000);
    }
}
}

//Verificação das conexões
void VerificaConexoesWiFIEMQTT(void)
{
    if (!MQTT.connected())
        reconnectMQTT(); //se não há conexão com o Broker, a conexão é refeita

    reconnectWiFi(); //se não há conexão com o WiFi, a conexão é refeita
}

//Reconexão com o WiFi
void reconnectWiFi(void)
{
    //se já está conectado a rede WI-FI, nada é feito.
    //Caso contrário, são efetuadas tentativas de conexão
    if (WiFi.status() == WL_CONNECTED)
        return;

    WiFi.begin(SSID, PASSWORD); // Conecta na rede WI-FI

    while (WiFi.status() != WL_CONNECTED)
    {
        delay(100);
        Serial.print(".");
    }

    Serial.println();
    Serial.print("Conectado com sucesso na rede ");
    Serial.print(SSID);
    Serial.println("IP obtido: ");
    Serial.println(WiFi.localIP());
}

//Perfil das plantas
void perfil_1(void){

```



```

tempo_bomba= 604800000;
tempo_luz= 21600000;
tempo_luzD= 21600000;
MQTT.publish(TOPICO_PUBLISH_PLANT_SEL, "Suculentas");

}
void perfil_2(void){
  tempo_bomba= 1209600000;
  tempo_luz= 28800000;
  tempo_luzD= 28800000;
  MQTT.publish(TOPICO_PUBLISH_PLANT_SEL, "Cactos");
}
void perfil_3(void){
  tempo_bomba= 172800000;
  tempo_luz= 21600000;
  tempo_luzD= 21600000;
  MQTT.publish(TOPICO_PUBLISH_PLANT_SEL, "Manjeriçã");
}

//Leitura do sensor umidade de solo
double umidade_solo(void)
{
  double t_solo = analogRead(SoilMoisture);
  double inv_t_solo = map(t_solo, 4095, 1500, 0, 100);
  if(inv_t_solo>100)
    inv_t_solo=100;

  return inv_t_solo;
}

//Ativação da bomba d'agua
void bomba(void)
{
  if(umidade_solo()<10 && (millis()-timerBomba)>= tempo_bomba)
  {
    ligar_bomba();
  }
  else if (umidade_solo()>80)
  {
    desligar_bomba();
  }
}

void ligar_bomba(void)
{
  digitalWrite(PIN_BOMBA, HIGH);
  MQTT.publish(TOPICO_PUBLISH_BOMBA, "#67F6EC");
}

```

```

void desligar_bomba(void)
{
    timerBomba=millis();
    digitalWrite(PIN_BOMBA, LOW);
    MQTT.publish(TOPICO_PUBLISH_BOMBA, "#526B5B");
}

//Leitura do LDR
double luz(void)
{
    double t_ldr = digitalRead(LDR);
    if (t_ldr == HIGH)
    {
        MQTT.publish(TOPICO_PUBLISH_LDR, "#807D06");
    }
    else if (t_ldr == LOW)
    {
        MQTT.publish(TOPICO_PUBLISH_LDR, "#FFFA0C");
    }
    return t_ldr;
}

//Ativação do LED de iluminação
void ilumina(void)
{
    if(millis()-timerLDR>=tempo_luz)
    {
        if(estadoLED == LOW)
            ligar_lamp();
        else desligar_lamp();
        timerLDR=millis();
    }
}

void ligar_lamp(void)
{
    estadoLED=!estadoLED;
    digitalWrite(PIN_LAMP, estadoLED);
    MQTT.publish(TOPICO_PUBLISH_LAMPADA, "#FF0DBE");
    //MQTT.publish(TOPICO_PUBLISH_LDR, "#FFFA0C"); FF0DBE
}

void desligar_lamp(void)
{
    estadoLED=!estadoLED;
    digitalWrite(PIN_LAMP, estadoLED);
    MQTT.publish(TOPICO_PUBLISH_LAMPADA, "#802D69");
    //MQTT.publish(TOPICO_PUBLISH_LDR, "#807D06"); 802D69
}

```

```
}

//Leitura da temperatura com o sensor DHT11
double faz_leitura_temperatura(void)
{
    double t = dht.readTemperature();
    double result_tmp;

    if (! (isnan(t)) )
        result_tmp = t;
    else
        result_tmp = -99.99;

    return result_tmp;
}

//Leitura da umidade do ar com o sensor DHT11
double faz_leitura_umidade(void)
{
    double h = dht.readHumidity();
    double result_umd;

    if (! (isnan(h)) )
    {
        result_umd = h;
        Serial.println(h);
    }
    else
    {
        result_umd = -99.99;
    }

    return result_umd;
}

// Função de setup
void setup()
{
    Serial.begin(115200);
    initWiFi();
    initMQTT();

    pinMode(SoilMoisture, INPUT);
    pinMode(LDR, INPUT);
}
```

```

pinMode(PIN_LED_1, OUTPUT);
pinMode(PIN_LED_2, OUTPUT);
pinMode(PIN_LED_3, OUTPUT);
pinMode(PIN_BOMBA, OUTPUT);
pinMode(PIN_LAMP, OUTPUT);

digitalWrite(PIN_LED_1,LOW);
digitalWrite(PIN_LED_2,LOW);
digitalWrite(PIN_LED_3,LOW);
digitalWrite(PIN_BOMBA, LOW);
digitalWrite(PIN_LAMP, LOW);

dht.begin();

//tempoSensor = millis();
}

//loop
void loop()
{
  /*Verificando a estabilidade das conexões*/
  VerificaConexoesWiFiMQTT();
  MQTT.loop();

  /*Enviando os valores para o mqtt*/
  MQTT.subscribe(TOPICO_SUBSCRIBE_PLANT_TYPE);

  /*String enviada para o mqtt*/
  char umidade_solo_str[10] = {0};
  char temperatura_str[10] = {0};
  char umidade_str[10] = {0};

  bomba();
  luz();
  ilumina();

  if (millis() - tempoSensor >= 4500){

    /*Transferindo o valor das funções para as strings que serão publicadas*/
    dtostrf(umidade_solo(), 3, 0,umidade_solo_str);
    dtostrf(faz_leitura_temperatura(), 3, 0,temperatura_str);
    dtostrf(faz_leitura_umidade(), 3, 0,umidade_str);

    MQTT.publish(TOPICO_PUBLISH_UMIDADE_SOLO, umidade_solo_str);
  }
}

```

```
MQTT.publish(TOPICO_PUBLISH_TEMPERATURA, temperatura_str);
MQTT.publish(TOPICO_PUBLISH_UMIDADE, umidade_str);

/*Mostrando no monitor serial as funções que serão publicadas no mqtt*/
Serial.print("Umidade Solo:");
Serial.println(umidade_solo_str);
Serial.print("Umidade:");
Serial.println(umidade_str);
Serial.print("Umidade Temperatura:");
Serial.println(temperatura_str);
tempoSensor = millis();
}

delay(500);
}
```