

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CAMPUS TIMÓTEO**

Karine Rodrigues Sampaio de Moraes

**PROTÓTIPO DE IRRIGAÇÃO AUTOMATIZADA E CUSTOMIZÁVEL
PARA PLANTAS RESIDENCIAIS**

Timóteo

2023

Karine Rodrigues Sampaio de Moraes

**PROTÓTIPO DE IRRIGAÇÃO AUTOMATIZADA E CUSTOMIZÁVEL
PARA PLANTAS RESIDENCIAIS**

Monografia apresentada à Coordenação de Engenharia de Computação do Campus Timóteo do Centro Federal de Educação Tecnológica de Minas Gerais para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Dr. Elder de Oliveira Rodrigues

Timóteo

2023

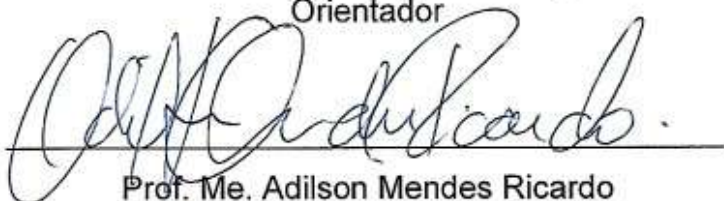
**PROTÓTIPO DE IRRIGAÇÃO AUTOMATIZADA E CUSTOMIZÁVEL
PARA PLANTAS RESIDENCIAIS**

Trabalho de Conclusão de Curso
apresentado ao Curso de Engenharia de Computação
do Centro Federal de Educação Tecnológica de
Minas Gerais, campus Timóteo, como requisito
parcial para obtenção do título de Engenheiro de
Computação.

Trabalho aprovado. Timóteo, 8 de Dezembro de 2023:



Prof. Dr. Elder de Oliveira Rodrigues
Orientador



Prof. Me. Adilson Mendes Ricardo
Professor Convidado



Profª. Dra. Viviane Cota Silva
Professora Convidada

Dedico esse trabalho aos meus pais, Márcio e Renata,
ao meu irmão Gustavo e meu marido Guilherme.

Agradecimentos

Primeiramente, agradeço a Deus, que me permitiu alcançar todos os meus objetivos durante todos esses anos de estudos. Agradeço a toda minha família, em especial aos meus pais, meu irmão e meu marido, que sempre estiveram presentes em todos os momentos importantes da minha vida, contribuindo com palavras motivacionais, acreditando e me apoiando sempre, foram meu alicerce e me ajudaram a superar todos os desafios. Agradeço aos meus melhores amigos por toda a parceria durante essa jornada, Matheus Otero e Alysson Kelvim, o trio inseparável que levarei pro resto da minha vida. Agradeço ao meu orientador, Elder Rodrigues, por sua orientação valiosa, paciência e incentivo ao longo deste processo. Por fim, agradeço à instituição de ensino por fornecer um ambiente propício ao aprendizado e pesquisa, bem como pelos recursos e oportunidades oferecidos.

Cada um de vocês desempenhou um papel fundamental neste percurso, e por isso, expresso minha profunda gratidão. Muito obrigada.

"Não perca seu tempo sempre procurando pelos anos perdidos. Encare, agente firme.
Perceba que você está vivendo nos anos dourados."
Iron Maiden.

Resumo

A irrigação adequada é essencial para a saúde das plantas, impactando diretamente no seu crescimento e desenvolvimento, e a automação residencial representa uma revolução na maneira como interagimos com nossos ambientes domésticos, trazendo eficiência e conveniência para o dia a dia. Neste contexto, este trabalho detalha o desenvolvimento de um sistema de monitoração e irrigação automática para uma planta residencial, empregando o módulo NodeMCU ESP8266. Visando oferecer uma solução prática e eficiente para a manutenção de plantas em ambientes domésticos, este projeto é particularmente benéfico para indivíduos com rotinas intensas ou limitada experiência em jardinagem. O sistema integra sensores de umidade e temperatura do ar, bem como um sensor de umidade do solo, todos conectados ao NodeMCU ESP8266, fornecendo dados vitais sobre as condições ambientais e do solo em tempo real para um manejo eficaz da água. Utilizando as plataformas Blynk e Thingspeak, possibilita-se a criação de *dashboards* personalizados para uma visualização intuitiva e interativa, além de permitir o controle remoto e monitoramento por meio de um aplicativo móvel, agregando conveniência e flexibilidade ao sistema. Notavelmente, a válvula de irrigação do sistema é versátil, podendo ser ativada manualmente, por programação temporizada, ou automaticamente com base nos limites de umidade do solo estabelecidos pelo usuário, assegurando a hidratação adequada das plantas conforme diversas necessidades e preferências. Projetado para ser econômico e de fácil instalação, o sistema democratiza a automação da irrigação para um público extenso. Testes confirmaram sua eficácia no monitoramento e ativação da irrigação, proporcionando controle aprimorado sobre a umidade do solo e incentivando a economia de água.

Palavras-chave: Irrigação automática, NodeMCU ESP8266, Blynk, Thingspeak.

Abstract

Proper irrigation is essential for the health of plants, directly impacting their growth and development, and home automation represents a revolution in the way we interact with our domestic environments, bringing efficiency and convenience to everyday life. In this context, this work details the development of a monitoring and automatic irrigation system for a residential plant, employing the NodeMCU ESP8266 module. Aiming to offer a practical and efficient solution for maintaining plants in domestic environments, this project is particularly beneficial for individuals with intense routines or limited experience in gardening. The system integrates air humidity and temperature sensors, as well as a soil moisture sensor, all connected to the NodeMCU ESP8266, providing vital data on environmental and soil conditions in real-time for effective water management. Using the Blynk and Thingspeak platforms, it enables the creation of customized *dashboards* for intuitive and interactive visualization, as well as allowing for remote control and monitoring through a mobile application, adding convenience and flexibility to the system. Notably, the irrigation valve of the system is versatile, capable of being activated manually, by timed programming, or automatically based on the soil moisture limits set by the user, ensuring adequate hydration of the plants according to various needs and preferences. Designed to be economical and easy to install, the system democratizes irrigation automation for a wide audience. Tests have confirmed its effectiveness in monitoring and activating irrigation, providing enhanced control over soil moisture and encouraging water conservation.

Keywords: Automatic irrigation, NodeMCU ESP8266, Blynk, Thingspeak.

Lista de ilustrações

Figura 1 – Sistema de irrigação localizada por gotejamento	15
Figura 2 – Escopo da IoT	15
Figura 3 – NodeMCU com ESP-12E	17
Figura 4 – Organização do Blynk: Blynk App, Server e Libraries	19
Figura 5 – Tabela de preços Blynk	20
Figura 6 – Exemplo Blynk Apps	21
Figura 7 – Exemplo Blynk Console	21
Figura 8 – Sensor capacitivo de umidade do solo	23
Figura 9 – Sensor DHT11	23
Figura 10 – Esboço do Projeto de um sistema de irrigação automatizado com Arduino . .	24
Figura 11 – Topologia do sistema de irrigação automático supervisionado para hortas . .	25
Figura 12 – Projeto completo do sistema de irrigação automatizado de baixo custo . . .	26
Figura 13 – Fluxograma da metodologia utilizada para o desenvolvimento do projeto. . .	28
Figura 14 – Lista de preços	30
Figura 15 – Visão esquemática do sistema de irrigação	31
Figura 16 – Diagrama do circuito elétrico utilizado no projeto	32
Figura 17 – Protótipo físico do projeto	34
Figura 18 – Biblioteca do ESP8266 no Gerenciador de Placas	35
Figura 19 – Bibliotecas DHT e Blynk	35
Figura 20 – Criação de um template	36
Figura 21 – <i>Datastreams</i> utilizados no protótipo	37
Figura 22 – <i>Web Dashboard</i>	38
Figura 23 – Configuração de <i>automations</i> para os <i>datastreams</i> do protótipo	38
Figura 24 – Automações configuradas para o projeto	39
Figura 25 – Configuração da automação para válvula definida por umidade do solo . . .	39
Figura 26 – Configuração da automação para válvula definida por período de tempo . . .	40
Figura 27 – Interface Blynk no aplicativo mobile	41
Figura 28 – Aba de automações no aplicativo Blynk	41
Figura 29 – Logs de chamads do webhook no console Blynk	42
Figura 30 – Página web ThingSpeak contendo o canal do projeto	43
Figura 31 – Fluxograma das etapas desde a obtenção de dados até o controle da válvula .	44
Figura 32 – Notificações relacionadas à válvula ligada	46
Figura 33 – Notificações relacionadas à válvula desligada	47
Figura 34 – Interface Blynk Console após o teste de umidade do solo	47
Figura 35 – Medição de temperatura no Blynk Console	48
Figura 36 – Medição de temperatura hora a hora	48
Figura 37 – Gráfico com os valores de todos os sensores	49
Figura 38 – Página do ThingSpeak com os gráficos gerados	50

Lista de abreviaturas e siglas

IOT	Internet of Things
GPIO	General Purpose Input/Output
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
USART	Universal Synchronous and Asynchronous Receiver-Transmitter
RTOS	Real Time Operation System
WiFi	Wireless Fidelity
SGBD	Sistema Gerenciador de Banco de Dados
AURESIDE	Associação Brasileira de Automação Residencial e Predial
HTTP	Hypertext Transfer Protocol
PWM	Pulse-Width Modulation
I2S	Inter-Integrated-Chip Sound
OTA	Over The Air
HTML	HyperText Markup Language
API	Application Programming Interface
XML	eXtensible Markup Language
JSON	JavaScript Object Notation
URL	Uniform Resource Locator
MQTT	Message Queuing Telemetry Transport
CSV	Comma-Separated Values

Sumário

1	INTRODUÇÃO	11
1.1	Problema	12
1.2	Justificativa	12
1.3	Objetivos	12
2	REVISÃO BIBLIOGRÁFICA	14
2.1	Métodos de irrigação	14
2.2	Internet das coisas	15
2.3	Módulo ESP8266	16
2.4	ThingSpeak	17
2.5	Protocolo HTTP	17
2.6	APIs e <i>Webhooks</i>	18
2.7	<i>Software</i> Blynk	18
2.7.1	Blynk Apps e Console	20
2.7.2	Blynk Server	21
2.7.3	Blynk Libraries	22
2.8	Componentes	22
2.8.1	Sensor de umidade do solo	22
2.8.2	Sensor de umidade e temperatura DHT11	23
2.9	Trabalhos Relacionados	24
3	MÉTODOS E MATERIAIS	27
4	CONSTRUÇÃO DE UM PROTÓTIPO DE SISTEMA DE IRRIGAÇÃO	31
4.1	Criação da estrutura	31
4.2	Circuito eletrônico	32
4.3	Implementação física do protótipo	33
4.4	Implementação do software	34
4.4.1	Blynk Console	35
4.4.1.1	<i>Template</i>	36
4.4.1.2	<i>Datastreams</i>	36
4.4.1.3	<i>Web Dashboard</i>	37
4.4.1.4	<i>Automations</i>	38
4.4.2	Blynk App	40
4.4.3	ThingSpeak	41
5	TESTES E RESULTADOS	45
5.1	Discussão	45
5.2	Testes	45

5.3	Resultados	49
6	CONCLUSÃO	51
6.1	Trabalhos futuros	52
	REFERÊNCIAS	53
	APÊNDICE A – CÓDIGO DESENVOLVIDO NO ARDUINO IDE	55

1 Introdução

Com o crescimento da tecnologia e a disponibilidade de dispositivos eletrônicos com preços acessíveis, o mercado de automação, segundo a AURESIDE (Associação Brasileira de Automação Residencial e Predial), deve aumentar cerca de 20% até 2023 (ZANATTA, 2021). Neste contexto, Accardi e Dodonov (2012) definem a automação residencial como a "integração de serviços e tecnologias, que tem por finalidade tornar uma residência automatizada e obter aumento em relação a segurança, conforto e praticidade".

O cultivo de plantas e hortas caseiras é uma prática em ascensão, pois permite ao indivíduo dedicar um tempo a si mesmo, combater o estresse, além de ser utilizada como uma fonte de produtos naturais em casa (CARVALHO et al., 2019). A automação surge como um aliado e facilitador à esta prática.

Um ponto de atenção para o horticultor é a quantidade de água necessária para cada planta, pois como aponta Santos (2014), a escassez de água extrema impossibilita que os nutrientes do solo ou da adubação sejam solubilizados de maneira correta para que as plantas consigam absorver. A falta de água também faz com que as plantas murchem e eventualmente morram por desidratação. Além disso quando existe água em excesso, a fertilidade da terra pode ser prejudicada, as raízes das plantas podem ser asfixiadas, o que as levará ao apodrecimento e fará com que deixem de absorver os nutrientes, e a própria água de que necessitam.

No cultivo caseiro o tipo de irrigação mais utilizado é a irrigação localizada, que consiste em um método onde a água é aplicada ao solo em pequenas quantidades, com alta frequência, mantendo-o assim constantemente úmido. Esse tipo de irrigação vem ganhando vários adeptos, principalmente em regiões onde o fator água é limitante (BISCARO, 2014). Desta forma, a eficiência do método de irrigação se mostra vital, pois segundo Coelho e Simoes (2015), quando há um "menor consumo de energia e menor perda de água, contribuindo com a maior conservação dos recursos hídricos, por ser aplicada no momento correto, isto é, aquele em que a umidade do solo começa a comprometer a absorção de água pela planta".

Um exemplo da importância de uma irrigação correta foi observada por Castro (2019) que mostrou como a disponibilidade hídrica é capaz de influenciar tanto a produção como a composição dos metabólitos secundários de uma planta. Uma das amostras analisadas durante o estudo foi o manjeriço Alfavaca basilicão, que foi submetido a diferentes disponibilidades hídricas e apresentou crescimento e desenvolvimento aceitável com um valor a partir de 80% da quantidade de água considerada como ideal. Nos casos em que esta disponibilidade ficou abaixo deste percentual, o desenvolvimento das plantas foi prejudicado.

Plantas como o manjeriço tem uma enorme gama de utilidades, como é mostrado por Loughrin e Kasperbauer (2001) e citado por Barreiro et al. (2006):

Ocimum basilicum, o manjeriço, planta tradicionalmente usada como erva medicinal no tratamento de dor de cabeça, tosse, diarreia, entre outros, é também

considerada fonte de componentes aromáticos. Seu óleo essencial tem sido muito usado como condimento em carnes, saladas, bebidas não alcoólicas, sorvetes e na indústria de perfume e produtos de higiene bucal.

Devido a todos os benefícios citados, se faz essencial um meio de controlar e disponibilizar a água na quantidade correta e no momento certo para que a planta cresça bem em residências, mas com potencial de adaptação e aplicação em um contexto mais amplo, como por exemplo, para pequenos agricultores.

1.1 Problema

No ritmo acelerado da vida moderna, caracterizado por longas horas de trabalho e frequentes viagens, muitos indivíduos enfrentam dificuldades em manter o cuidado adequado de suas plantas e hortas domésticas. Este desafio é exacerbado pela falta de conhecimento específico sobre os cuidados necessários para diferentes tipos de plantas, pela limitação de tempo disponível e, frequentemente, pelo simples esquecimento. Tal realidade levanta uma questão crítica: como desenvolver uma solução prática e eficaz que possa atenuar essas dificuldades e facilitar o cuidado adequado de plantas domésticas?

1.2 Justificativa

A implementação de um sistema de irrigação automatizado e personalizável é justificada pela necessidade de otimizar o uso de recursos hídricos, reduzir custos e melhorar a saúde e qualidade de hortaliças e plantas domésticas. Este sistema permite que os usuários configurem perfis de irrigação adaptados às necessidades específicas de cada planta, baseando-se em um entendimento básico de fatores cruciais destacados por Biscaro (2014), como as exigências hídricas, nutricionais e sanitárias das diferentes culturas, as propriedades do solo, incluindo sua textura e condutividade hidráulica, bem como as características climáticas e os padrões de molhamento. Além disso, o protótipo apresenta um grande potencial para ser utilizado em futuras pesquisas em laboratório na área agrícola, abrindo novos horizontes em estudos de solos, plantas, agricultura familiar e agronegócio. A versatilidade do protótipo permite experimentos controlados, contribuindo para o avanço científico no campo da agronomia e da horticultura sustentável.

1.3 Objetivos

Tem-se como objetivo geral deste trabalho desenvolver um sistema de irrigação automatizado e customizável utilizando o módulo NodeMCU ESP8266 para auxiliar e otimizar o cultivo de plantas residenciais, através de um aplicativo que enviará comandos via rede de *internet* para uma válvula que será acionada de acordo com a configuração do usuário para determinada planta.

Também, objetivam-se mais especificamente:

1. Dimensionar os componentes eletrônicos do circuito que vão exercer o controle sobre a válvula;
2. Construir o protótipo para um vaso de planta com materiais de baixo custo;
3. Permitir ao usuário realizar a irrigação de três maneiras: manualmente, por horário programado ou através de automações;
4. Elaborar a comunicação do *Smartphone* (celular) e a página web com o dispositivo NodeMCU ESP8266;
5. Coletar dados de todos os sensores e exibi-los em um *dashboard*;
6. Permitir ao usuário acesso ao histórico de irrigação de suas plantas para que possa comparar a eficácia entre os métodos baseando-se no desenvolvimento das plantas.

2 Revisão Bibliográfica

“As pessoas têm medo das mudanças. Eu tenho medo que as coisas nunca mudem”.

Chico Buarque

O objetivo desse capítulo é apresentar uma revisão bibliográfica composta de informações teóricas que serão utilizadas para o desenvolvimento deste trabalho, além de apresentar trabalhos similares que também propõem a irrigação automatizada como solução.

2.1 Métodos de irrigação

Os métodos de irrigação determinam a maneira com a qual a água será aplicada às plantas. Segundo Andrade (2001), os quatro principais métodos de irrigação são: por superfície, por aspersão, localizada e sub irrigação. O princípio aplicado na irrigação por superfície é a distribuição da água, por meio da gravidade, através da superfície do solo. No método por aspersão, jatos de água são lançados no ar e caem sobre o solo em forma de chuva. Já a técnica conhecida como localizada consiste na aplicação de água em parte do sistema radicular das plantas por meio de emissores pontuais. Finalmente, a sub irrigação parte do princípio de manter o lençol freático a uma profundidade tal que permita o fluxo de água adequado à cultura através de, por exemplo, um sistema de drenagem.

Para a realização deste trabalho foi utilizada a irrigação localizada. Dentro desse método, segundo Andrade e Brito (2006), existem dois tipos de sistema: gotejamento, onde os emissores molham parte do sistema radicular da planta, atingindo uma proporção de área molhada de 20 a 80% de sua área total e resultando em uma economia de água; e subsuperficial, que se diferencia do anterior somente na instalação, sendo seus emissores enterrados no solo.

De acordo com Andrade e Brito (2006), no sistema de gotejamento apresentado na figura 1, a aplicação da água é feita diretamente na superfície do solo, evitando molhar a folhagem e o colmo das plantas, o que diminui a probabilidade do aparecimento de fungos. Além disso, se comparado com o subsuperficial, apresenta maior facilidade de instalação, inspeção, limpeza, reposição, avaliação da área molhada e até mesmo medição da vazão dos emissores. Uma desvantagem é a possibilidade de entupimento da mangueira, caso a água não seja bem filtrada. Devido as características citadas, esse sistema foi o escolhido.

Figura 1 – Sistema de irrigação localizada por gotejamento



Fonte: (BISCARO, 2014)

2.2 Internet das coisas

A *internet* das coisas segundo Filho (2016) "surgiu recentemente como um novo conceito de 'rede', que abrange comunicações e processamento dos mais diversos equipamentos". De acordo com Magrani (2018), IoT (*Internet of Things*) seria um ecossistema de computação onipresente, criado através da interconectividade de objetos físicos conectados à *internet* através de pequenos sensores, cuja finalidade é a facilitação do cotidiano da sociedade, apresentando soluções funcionais para processos rotineiros. Uma visão geral do escopo da IoT pode ser observado na figura 2 que contém diversos equipamentos que podem ser encaixados nesse conceito.

Figura 2 – Escopo da IoT



Fonte: (FILHO, 2016)

2.3 Módulo ESP8266

O módulo ESP8266, produzido pela empresa chinesa Espressif, possui um núcleo microprocessado de Tensilica L106 de 32 bits e opera na frequência-padrão de 80MHz, podendo atingir até 160MHz. Ele possui um módulo WiFi (Wireless Fidelity), tensão nominal de funcionamento de 3,3 volts, 17 interfaces de entrada e saída, também chamadas de GPIO (*General Purpose Input/Output*), quatro interfaces de saída PWM (*Pulse-Width Modulation*), uma entrada analógica com 10 bits de precisão e interfaces de series síncronas, como SPI (*Serial Peripheral Interface*), I2C (*Inter-Integrated Circuit*) e I2S (*Inter-Integrated-Chip Sound*) e assíncronas, como USART (*Universal Synchronous and Asynchronous Receiver-Transmitter*) (OLIVEIRA, 2017).

Ainda segundo Oliveira (2017), o ESP8266 incorpora a atualização OTA (*Over The Air*) que permite que o *firmware* seja atualizado por meio da rede WiFi, em contraste com a tradicional atualização via porta USB/Serial. Essa funcionalidade é crucial para a atualização de software, especialmente para corrigir erros após a implantação. O dispositivo pode receber atualizações em intervalos regulares, de maneira semelhante ao que acontece com os sistemas operacionais modernos.

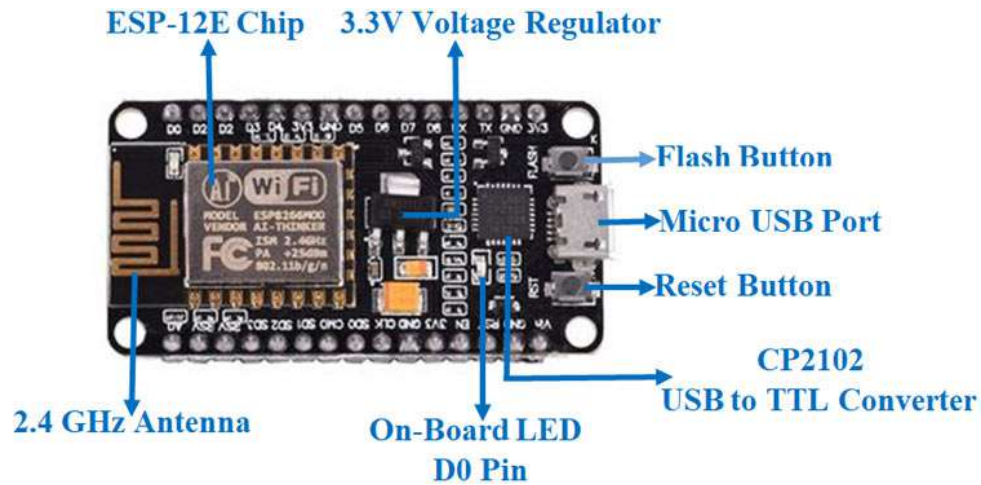
Para o desenvolvimento das aplicações do ESP8266, existem três ambientes apontados por Oliveira (2017):

- Utilizando a linguagem Lua;
- Baseado no sistema operacional de tempo real RTOS (*Real Time Operation System*);
- Ambiente do Arduino e todas as suas bibliotecas e interfaces.

Além disso o ESP8266 possui interoperabilidade com todos os módulos existentes para o Arduino, implicando em uma grande disponibilidade de placas, sensores, atuadores e diferentes ferramentas de comunicação. Todas essas características, seu baixo custo, documentação vasta e a facilidade de implementação, tornam o ESP8266 muito atraente para ser utilizado principalmente no desenvolvimento de aplicações IoT (OLIVEIRA, 2017).

Na criação do protótipo deste trabalho será utilizado, como é visto na figura 3, o módulo NodeMCU que é composto pelo ESP8266 contido dentro da placa ESP-12, juntamente com um conversor USB/Serial, um regulador de tensão e uma memória *Flash* de 4MB (OLIVEIRA, 2017).

Figura 3 – NodeMCU com ESP-12E



Fonte: (COMPONENTS101, 2020)

2.4 ThingSpeak

O ThingSpeak é uma plataforma aberta de IoT com análise MatLAB que suporta vários protocolos para entrada e saída de dados e permite agregar, visualizar e analisar fluxos de dados em tempo real na nuvem. É possível enviar dados para o ThingSpeak a partir de seus dispositivos, criar visualização instantânea de dados ao vivo e emitir alertas. É gratuito para usuários não comerciais, com limites em determinadas funcionalidades e se integra bem com outras plataformas e serviços, incluindo o MatLAB e APIs (*Application Programming Interface*) de serviços de terceiros, ampliando suas capacidades de processamento e análise de dados. Além de permitir compartilhar os dados e visualizações de forma pública ou privada, facilitando a colaboração e o acesso remoto aos dados. Devido a esses fatores ele é usado em uma variedade de aplicações, como projetos de automação residencial, monitoramento ambiental, agricultura inteligente, e qualquer projeto que necessite de coleta e análise de dados de sensores (THINGSPEAK, 2023).

2.5 Protocolo HTTP

O HTTP (*Hypertext Transfer Protocol*) é um protocolo que viabiliza a aquisição de recursos, como documentos HTML (*HyperText Markup Language*). É a fundação para todas as interações de dados na Web e opera como um protocolo cliente-servidor clássico, onde um cliente abre uma conexão, executa uma requisição e espera até receber uma resposta. O protocolo HTTP estabelece uma série de métodos de requisição que especificam a ação a ser realizada em relação a um recurso específico. Embora esses métodos possam ser conceituados como substantivos, são frequentemente chamados de verbos HTTP (CONTRIBUTORS, 2023). Os métodos de requisição mais comuns são:

- GET: O método GET solicita a representação de um recurso específico. Requisições utilizando o método GET devem retornar apenas dados;

- POST: O método POST é utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor;
- PUT: O método PUT substitui todas as atuais representações do recurso de destino pela carga de dados da requisição.
- DELETE: O método DELETE remove um recurso específico.

2.6 APIs e Webhooks

Uma API é um conjunto de definições e protocolos que facilitam o desenvolvimento e integração de aplicações. A comunicação entre APIs é comparada a um contrato entre um provedor e um usuário, estabelecendo o formato da solicitação e da resposta. Em geral, as APIs web usam HTTP para solicitar e receber dados, comumente em formatos como XML (*eXtensible Markup Language*) ou JSON (*JavaScript Object Notation*). Quando uma API cliente solicita dados de uma API servidor, ela realiza uma busca enviando solicitações regulares até receber os dados relevantes, conhecidos como *payload*. Essa interação é essencial para verificar a ocorrência de eventos específicos e manter as informações atualizadas (REDHAT, 2022).

O *webhook* é uma função de retorno de chamada baseada em HTTP que facilita a comunicação entre duas interfaces de programação de aplicações (APIs). Esses recursos são empregados por diversas aplicações web para receber quantidades reduzidas de dados provenientes de outras aplicações. Para configurar um *webhook* o cliente deve oferecer uma URL (*Uniform Resource Locator*) exclusiva à API servidor, especificando o evento desejado, sendo que após configurado o servidor enviará automaticamente o *payload* relevante para o URL do *webhook* quando o evento ocorrer. Os *webhooks* são considerados APIs reversas ou de push, transferindo a responsabilidade de comunicação para o servidor. Em vez do cliente solicitar dados, o servidor envia uma solicitação POST exclusiva quando os dados estão disponíveis. Para utilização de um *webhook* é sempre necessário existir uma API, pois eles trabalham juntos (REDHAT, 2022).

2.7 Software Blynk

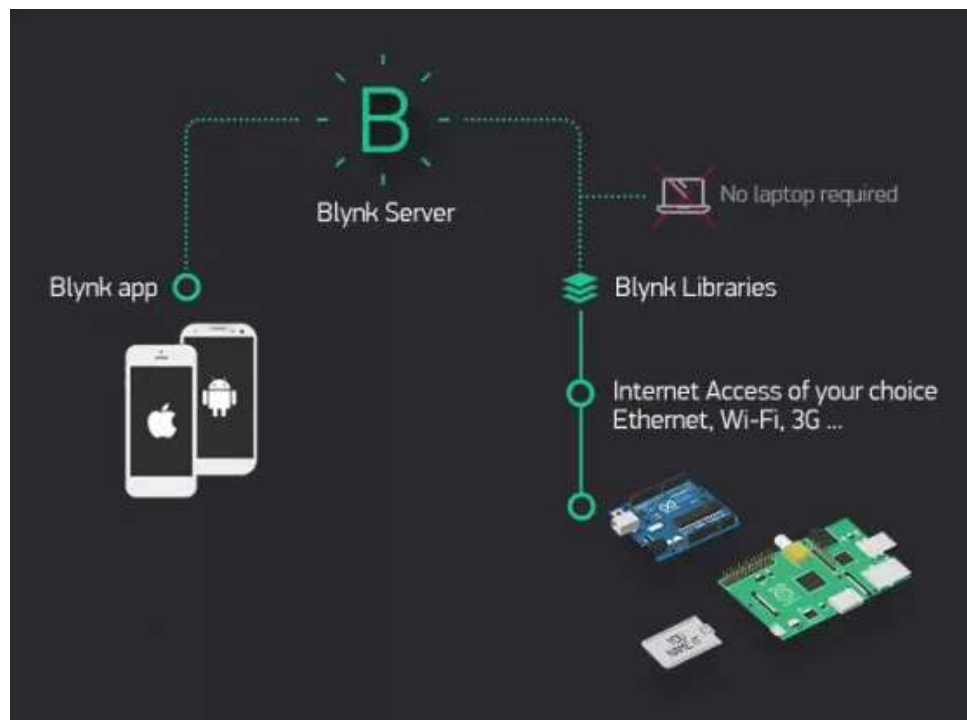
A crescente interconexão de dispositivos através de IoT tem gerado uma demanda crescente por plataformas que facilitem o desenvolvimento de projetos nesse campo. Neste trabalho, foi explorado a plataforma Blynk como uma solução simples, versátil e acessível para o desenvolvimento de projetos IoT (BLYNK, 2023a).

O Blynk oferece uma gama diversificada de recursos para facilitar o desenvolvimento de projetos, fornecendo uma biblioteca de *widgets* pré-construídos, como botões, *sliders*, gráficos e indicadores, que podem ser arrastados e soltos na interface do aplicativo. Esses *widgets* podem ser configurados para interagir com dispositivos físicos conectados, permitindo o controle remoto e o monitoramento em tempo real. Segundo Serrano (2018), a plataforma

também oferece suporte a mais de 400 placas de desenvolvimento, como Arduino, Raspberry Pi e ESP8266, simplificando a integração de dispositivos em projetos. Além disso, possui integrações com serviços em nuvem, permitindo a troca de dados entre o aplicativo e plataformas como o ThingSpeak e o Google Sheets.

De acordo com Serrano (2018) o Blynk é composto pelo Blynk App, o Blynk Server e a Blynk Library e pode ser melhor compreendido através da figura 4.

Figura 4 – Organização do Blynk: Blynk App, Server e Libraries



Fonte: (SERRANO, 2018)

O Blynk possui diferentes tipos de planos que podem ser escolhidos dependendo da finalidade para qual será utilizado. O plano grátis permite a criação e utilização do Blynk conectado a 2 projetos em qual até 5 usuários podem estar conectados, além de um histórico de dados semanal de até 5 *streams* de dados por *template*. A figura 5 mostra especificações de valores e o que está incluso em cada pacote.

Figura 5 – Tabela de preços Blynk

O que está incluído nos planos

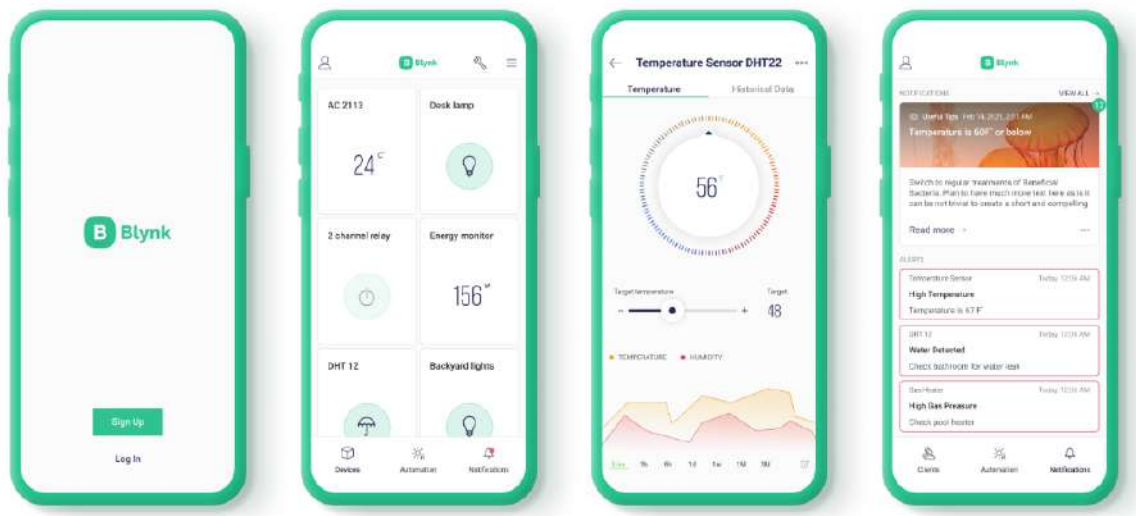
	Livre US\$ 0 mensais	Mais \$ 6,99 - \$ 12,99 mensais	Pró \$ 49 - \$ 449 mensais	Negócios \$ 599 - \$ 1.299 mensais
Aplicativos				
Blynk.App para iOS e Android Crie UI, conecte dispositivos e monitore-os remotamente. Nenhum código.	✓	✓	✓	Aplicativo móvel independente Seu próprio aplicativo IoT de marca pronto para ser publicado na App Store e no Google Play
Adicionar várias páginas a um painel de dispositivo	-	1 página ou 1 guia de informações do dispositivo	20 páginas, 7 guias de informações do dispositivo, 1 página de boas-vindas	20 páginas, 7 guias de informações do dispositivo, 1 página de boas-vindas
Widgets de aplicativos móveis	Widgets móveis básicos <i>i</i>	Além de widgets móveis <i>i</i>	Widgets móveis PRO <i>i</i>	Widgets móveis PRO <i>i</i>
Límite de widgets por modelo	10	80	255	Ilimitado
Leitor de QR/código de barras	-	-	✓	✓
Personalização de widgets	-	-	-	Contate-nos para um orçamento
Blynk.Console Portal web para configurar e gerenciar dispositivos, usuários, dados e organizações.	✓	✓	✓	Portal Web IoT de marca com a marca e URL da sua empresa
	COMECE DE GRAÇA	OBTENHA MAIS	TORNE-SE PROFISSIONAL	SABER MAIS

Fonte: (BLYNK, 2023b)

2.7.1 Blynk Apps e Console

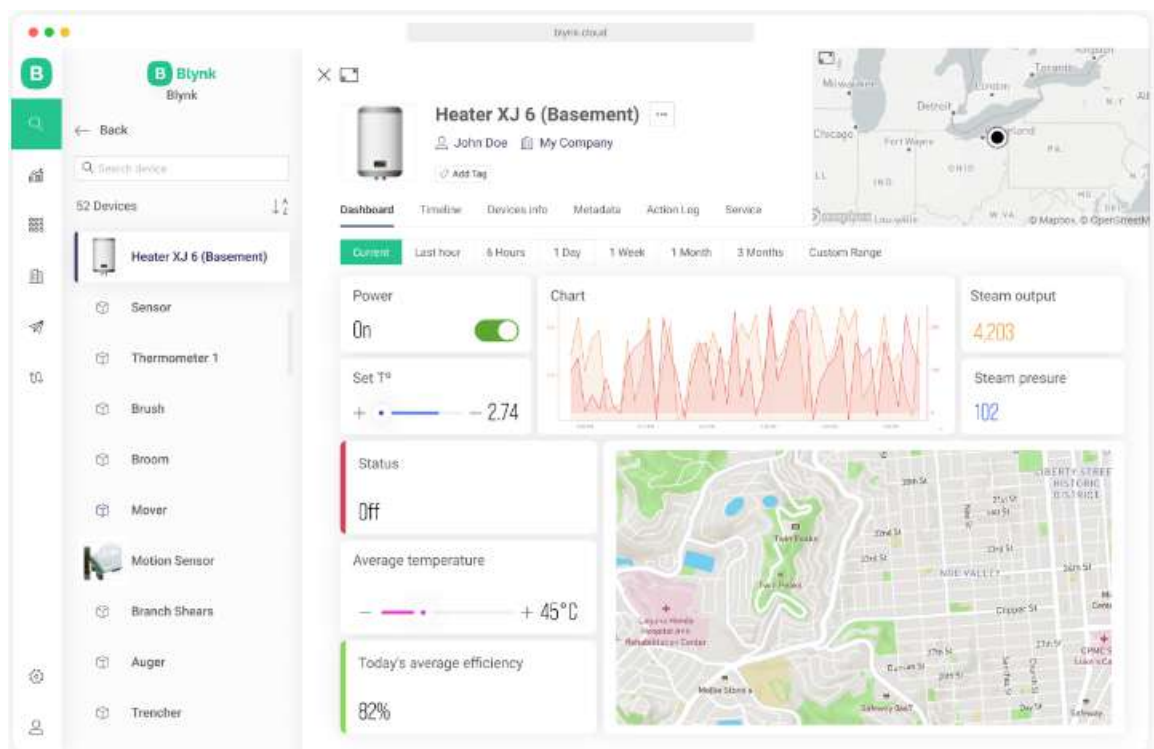
O Blynk Apps, observado na figura 6, é uma ferramenta disponível tanto para dispositivos com sistema operacional Android quanto para iOS, que possibilita aos usuários desenvolverem aplicações que interagem com componentes de *hardware*. Dentro de um ambiente dedicado para cada projeto, o usuário tem a sua disponibilidade *Widgets* que oferecem diversos tipos de funcionalidades, além de recursos de notificação e leitura de informações provenientes do *hardware*, permitindo sua exibição em *displays*, gráficos e mapas. Assim como o Apps o Blynk também possui o Blynk Console que pode ser observado na figura 7 possui as mesmas funcionalidades do Apps porém consiste de uma aplicação web (SERRANO, 2018).

Figura 6 – Exemplo Blynk Apps



Fonte: (BLYNK, 2023b)

Figura 7 – Exemplo Blynk Console



Fonte: (BLYNK, 2023b)

2.7.2 Blynk Server

A comunicação entre o aplicativo e o *hardware* é totalmente feita através da *cloud* Blynk. O servidor desempenha um papel fundamental ao transmitir informações para o *hardware*, manter o registro dos estados tanto do aplicativo quanto do *hardware*, e também ar-

mazenar os dados provenientes dos sensores, mesmo quando o aplicativo está fechado. É importante destacar que os dados armazenados no servidor Blynk podem ser acessados externamente por meio de uma API HTTP, o que abre a oportunidade de utilizar o Blynk para armazenar informações geradas de forma periódica (SERRANO, 2018).

2.7.3 Blynk Libraries

Por fim, no que diz respeito ao *hardware*, existem bibliotecas Blynk compatíveis com várias plataformas de desenvolvimento. Estas bibliotecas desempenham um papel crucial ao cuidar de toda a conexão entre o hardware e o servidor Blynk, além de gerenciar as solicitações de entrada e saída de dados e comandos. A maneira mais simples e ágil de utilizá-las é como bibliotecas para Arduino. No entanto, é possível obter versões da biblioteca adequadas para outras plataformas, como Linux, Python, Lua e muitas outras (SERRANO, 2018).

2.8 Componentes

2.8.1 Sensor de umidade do solo

O funcionamento do sensor capacitivo, apresentado na figura 8, pode ser entendido através da análise de sua construção. Ele é composto por dois terminais que são separados por um material dielétrico (o solo), responsável por armazenar energia elétrica em um campo elétrico. Para efetuar a medição o sensor cria um pequeno campo elétrico ao seu redor e através desse campo o solo começa a se comportar como um capacitor, permitindo posteriormente calcular a capacitância do ambiente com base no tempo de carga. Devido à relação direta entre essa capacitância e a quantidade de água presente, é possível concluir que um solo mais úmido leva a uma diminuição na tensão ou a um aumento no tempo de carga (CASTRO, 2023).

A maioria dos sensores de umidade do solo são baseados no CI LM193, um comparador de baixa voltagem que apresenta uma tensão de saída de acordo com a resistividade percebida nos eletrodos, em que quanto maior for a umidade do solo, maior sua tensão. A medição deve ser ajustada na implantação pois existem alguns fatores que podem interferir nos resultados das medições, como o tipo do solo, o pH, a salinidade, entre outros (CASTRO, 2023).

Figura 8 – Sensor capacitivo de umidade do solo

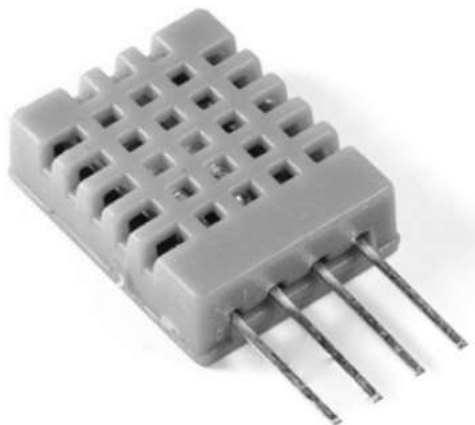


Fonte: (ELETROGATE, 2022)

2.8.2 Sensor de umidade e temperatura DHT11

O sensor de umidade e temperatura DHT11 representado na figura 9 é formado por um pequeno microcontrolador de 8 bits e utiliza apenas um pino de GPIO para transmitir as informações coletadas. Sua faixa de leitura de temperatura está entre 0 °C a 50 °C com precisão de 2 °C. Em relação à umidade, é capaz de realizar medições de 20% a 80%, com 5% de precisão. O componente é composto por três partes, um sensor capacitivo de umidade, um termistor e um chip que faz a conversão do sinal analógico para digital. A alimentação do sensor deve estar em um intervalo de 3,5 a 5 V DC (ADAFRUIT, 2019).

Figura 9 – Sensor DHT11



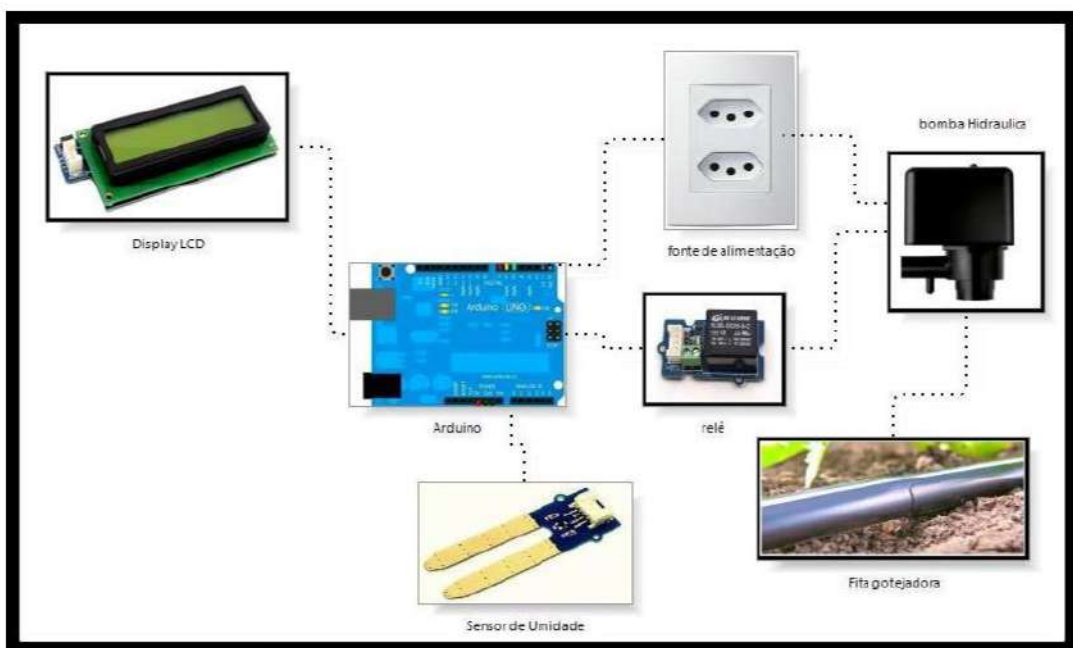
Fonte: (OLIVEIRA, 2017)

2.9 Trabalhos Relacionados

1. **Sistema de irrigação automatizado utilizando plataforma arduino:** A proposta do trabalho de Barbosa (2013) foi a criação de um protótipo de um sistema de irrigação automatizado barato e acessível de forma precisa. Para tal, foi preciso compreender e aprender, através de pesquisas, os conceitos relacionados a sistemas de irrigação, plataforma arduino e o uso de sensores.

A figura 10 representa o esboço do protótipo criado por Barbosa (2013), empregando sensores que coletam informações do solo e as enviam para o Arduino controlar uma bomba hidráulica, responsável pela irrigação do solo. Esta é ativada quando a variável de entrada for menor que 50% de umidade e é desativada quando solo chega a 70% de umidade. Com os resultados obtidos ao concluir o protótipo, o autor concluiu que é possível a criação de um produto com as tecnologias estudadas e que esse protótipo poderia, com algumas melhorias, até mesmo ser comercializado.

Figura 10 – Esboço do Projeto de um sistema de irrigação automatizado com Arduino

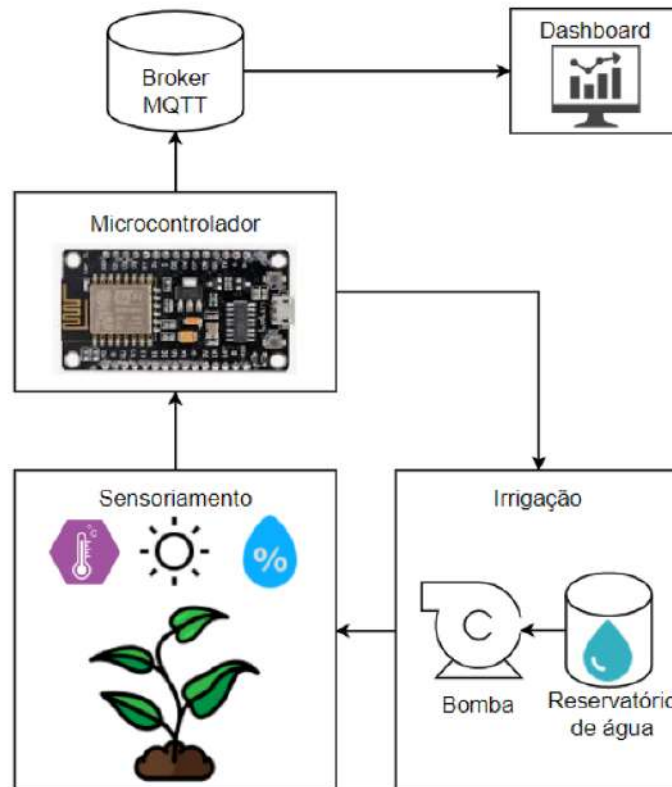


Fonte: (BARBOSA, 2013)

2. **Projeto: Sistema de irrigação automático supervisionado para hortas:** Como um primeiro passo em seu projeto, Adolphs (2020) fez formulário com perguntas relacionadas as características e hábitos das pessoas que criam plantas, obtendo 128 respostas. Destas, 95 pessoas disseram se preocupar sempre ou às vezes com o que acontecerá com suas plantas quando viajam e 68 pessoas afirmaram já ter deixado suas plantas sob o cuidado de outras pessoas. No formulário o autor questiona também o interesse pelo o sistema que pretende desenvolver, inclusive sobre a possibilidade de pagarem para utilizá-lo, obtendo um total de 92% de afirmações positivas.

Com os dados obtidos em sua pesquisa, Adolphs (2020) avalia como validada a necessidade da criação de um sistema automático de irrigação e demonstra sua implementação utilizando NodeMCU e o protocolo MQTT (*Message Queuing Telemetry Transport*). A irrigação nesse sistema seria feita de N em N horas por um período de T segundos de acordo com o desejo do usuário, sendo os dados obtidos pelos sensores disponibilizados em um dashboard de fácil visualização. A figura 11 denota o sistema concebido.

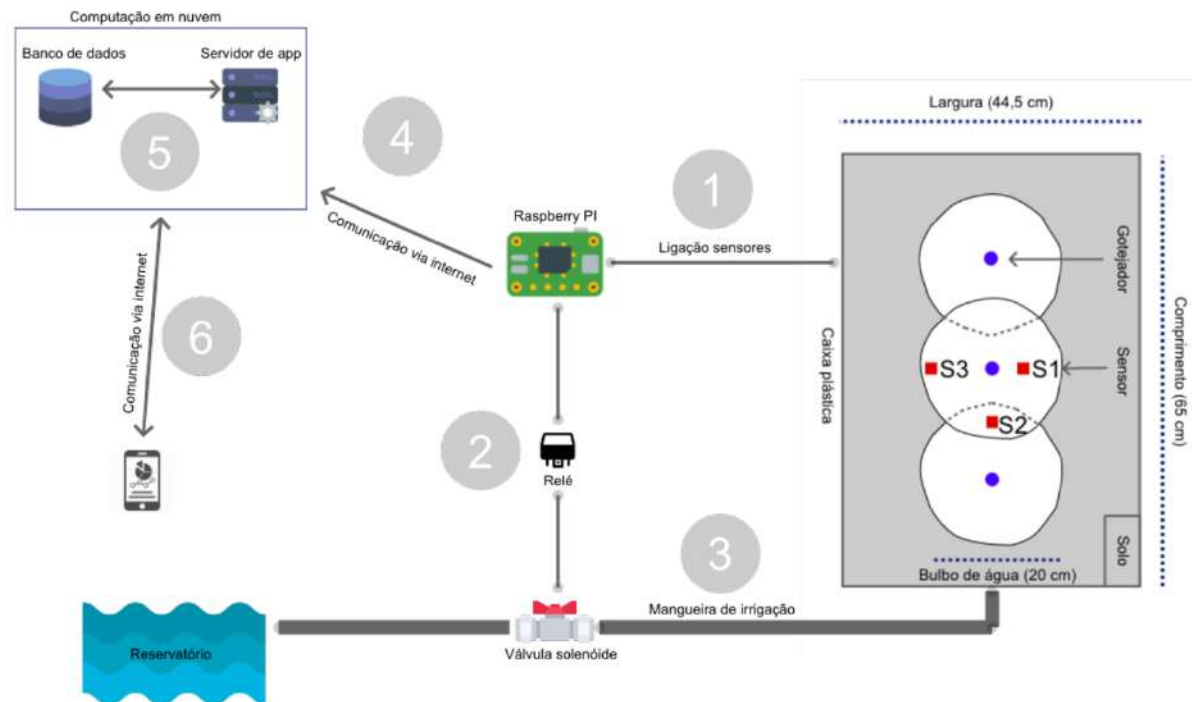
Figura 11 – Topologia do sistema de irrigação automático supervisionado para hortas



Fonte: (ADOLPHS, 2020)

3. **Desenvolvimento de um sistema de irrigação automatizado de baixo custo:** No trabalho de Giomo (2019) o principal objetivo foi desenvolver um sistema automático de irrigação por gotejamento, de baixo custo, que pudesse coletar dados e ser monitorado através de um dispositivo smartphone. Na figura 12 é representado o projeto completo, para o qual foram utilizados na construção três sensores de umidade do solo em posições diferentes, microcontrolador Raspberry, mangueira, gotejador auto compensante, válvula solenoide, relé, conversor de sinal analógico para digital, smartphone e computação em nuvem.

Figura 12 – Projeto completo do sistema de irrigação automatizado de baixo custo



Fonte: (GIOMO, 2019)

O experimento utilizou o cultivo do alface, onde após a calibração dos sensores e estudo sobre as necessidades específicas dessa planta, definiu-se que a irrigação seria iniciada quando um dos sensores atingisse 40% ou menos de umidade do solo, e cessada quando todos os sensores atingissem, no mínimo, um valor de 43%. Enquanto o processo de irrigação está acontecendo, os dados são coletados pelos sensores e salvos em coleções no banco de dados MongoDB a cada 30 segundos. Fora desse período os dados são coletados a cada 1h:30m. O autor concluiu que os resultados obtidos foram satisfatórios e que foi possível manter a umidade do solo nos parâmetros programados.

3 Métodos e Materiais

*“O sábio disse: apenas encontre o seu lugar no olho da tempestade,
procure as rosas ao longo do caminho,
só tome cuidado com os espinhos”.*
Scorpions

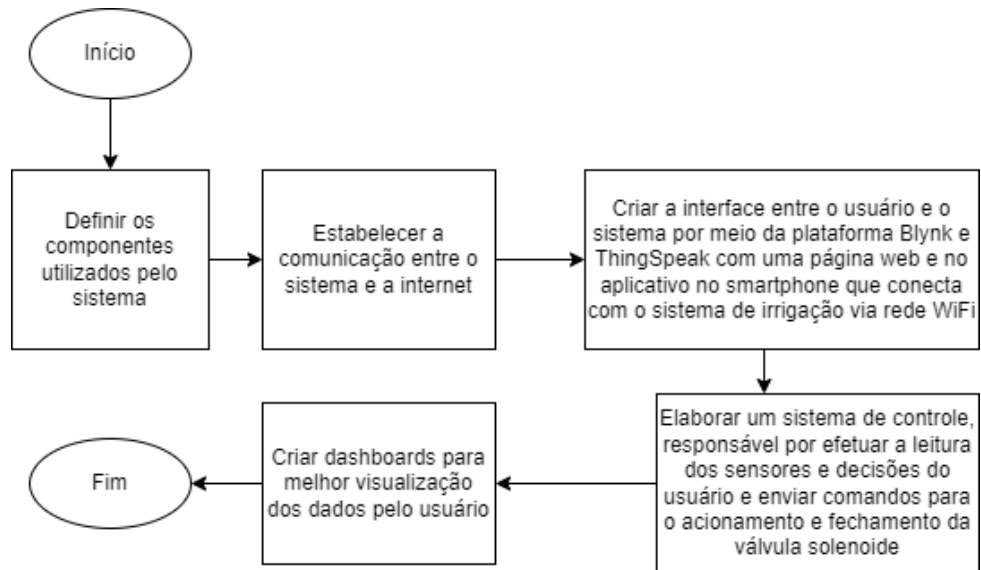
Segundo Gerhardt e Silveira (2009), a presente pesquisa é classificada como exploratória quanto aos objetivos, por envolver o levantamento bibliográfico e a análise de exemplos que estimulam a compreensão. Pode ser considerada quantitativa em relação a abordagem, visto que a análise dos resultados se baseia em quantidades e o foco é maior em uma quantidade pequena de conceitos. No que se refere à natureza é tida como aplicada, pois objetiva gerar conhecimentos para a aplicação prática, com o intuito da criação de um protótipo. Por fim, quanto aos procedimentos, pode ser classificada como uma pesquisa experimental por seguir um planejamento rigoroso em relação ao objeto do estudo e possuir delimitadas as variáveis que atuam sobre ele.

As etapas metodológicas foram organizadas da seguinte maneira:

1. Definir os componentes utilizados pelo sistema;
2. Estabelecer a comunicação entre o sistema e a internet;
3. Criar a interface entre o usuário e o sistema por meio da plataforma Blynk e ThingSpeak com uma página web e no aplicativo no smartphone que conecta com o sistema de irrigação via rede WiFi;
4. Elaborar um sistema de controle, responsável por efetuar a leitura dos sensores e decisões do usuário e enviar comandos para o acionamento e fechamento da válvula solenoide;
5. Criar *dashboards* para melhor visualização dos dados pelo usuário.

Para uma melhor compreensão, os procedimentos citados acima foram dispostos no fluxograma da figura 13.

Figura 13 – Fluxograma da metodologia utilizada para o desenvolvimento do projeto.



Fonte: Autor

Para atingir o objetivo deste trabalho, foram utilizados os seguintes equipamentos:

- Computador
 - Processador: Intel(R) Core(TM) i7-9700F CPU @ 3.00GHz
 - Memória RAM: 16 GB DDR4 3000 MHz
 - Sistema Operacional: Microsoft Windows 10 Pro de 64 bits
- Smartphone Samsung A71 6GB RAM e 128GB de armazenamento
- Roteador Wireless Intelbras Dual Band RF 1200
- Placa de desenvolvimento NodeMCU integrada com ESP-12E e ESP8266
 - Wireless padrão 802.11 b/g/n;
 - Antena embutida e conector micro-usb;
 - Modos de operação: STA/AP/STA+AP;
 - Suporta até 5 conexões TCP/IP;
 - 11 portas GPIO;
 - GPIO com funções de PWN, I2C, SPI, etc;
 - Tensão de operação: 4,5 9V;
 - Atualização remota de *firmware*;
 - Conversor analógico digital (ADC);
 - Distância entre pinos de 2,54mm
- Fonte DC chaveada 12V 2A

- Power Bank Carregador Samsung Portátil - 10000mah, 5V 2A
- Sensor capacitivo de umidade do solo V2.0
 - Tensão de operação: 3,3V DC
 - Pino de saída do sensor analógico
 - Tipo de detecção capacitiva
 - Comparador LM393
- Sensor de temperatura e umidade DHT11
 - Tensão de operação: 3,3V - 5,5V DC
 - Medição de umidade de 20 a 80%
 - Medição de temperatura de 0°C a 50°C
 - Faixa de corrente de 0,5 a 2,5mA
 - Precisão de $\pm 2^\circ\text{C}$
- Módulo relé 1 canal 5V - 80mA
- Módulo de alimentação para circuitos - Tensão Ajustável
- Válvula solenoide de entrada de água - 180° 1/2"x 1/2- 12V
- Amplificador operacional LM324N
- Mangueira de irrigação
- Gotejador Regulável GA 2
- Jumpers de conexão
- Protoboard 830 pontos
- Resistores de 10K Ω e 1K Ω
- Gotejador Regulável GA 2
- Ambiente de desenvolvimento
 - Arduino IDE 1.8.19

A Figura 14 a seguir contém uma tabela com os valores da maioria dos componentes acima, juntamente com seus respectivos preços. As linhas que estão em azul claro são componentes eletrônicos a serem utilizados e os que estão em verde claro são componentes que talvez o usuário já possua em casa e possa economizar.

Figura 14 – Lista de preços

Tabela de preços	
Componente	Preço
Módulo WiFi ESP8266 NodeMcu ESP-12	R\$ 47.90
Sensor DHT11	R\$ 11.90
Sensor Capacitivo de Umidade do Solo V2.0	R\$ 12.90
Módulo Relé 1 Canal 5v	R\$ 8.90
Amplificador operacional LM324N	R\$ 1.05
Válvula de Vazão Solenóide Água 12VDC	R\$ 39.90
Kit Jumpers - 65 Unidades	R\$ 14.90
Resistor 10K Ω - 20 Unidades	R\$ 2.90
Resistor 1K Ω - 20 Unidades	R\$ 2.90
Fonte 12V	R\$ 17.90
Protoboard 830 pontos	R\$ 14.31
Total Parcial	R\$ 175.46
Bico Gotejador Ga 2 Gotejamento Regulável	R\$ 0.72
Mangueira para Aquario - 2m	R\$ 8.99
Adaptador conector torneira mangueira - 3 unidades	R\$ 16.20
Conector fim de linha para mangueira	R\$ 3.30
Total Final	R\$ 204.67

Fonte: Autor

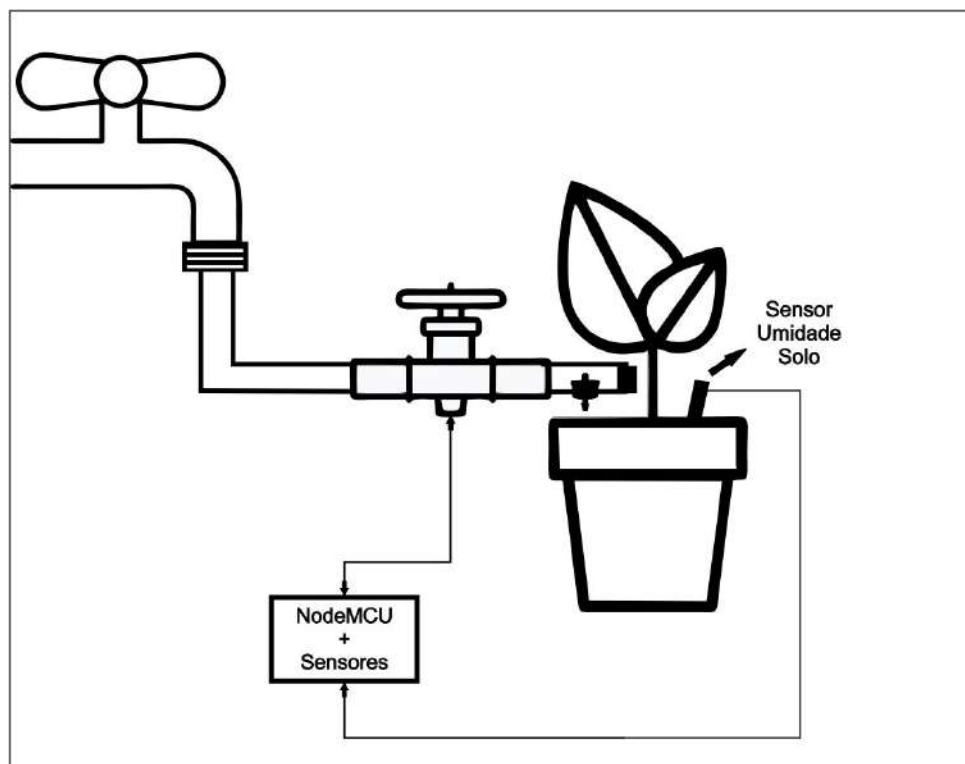
4 Construção de um protótipo de sistema de irrigação

O desenvolvimento deste trabalho consiste em criar um protótipo de um sistema de irrigação de baixo custo com uma válvula que pode ser acionada através de um dispositivo móvel ou aplicação web, por meio de automações facilmente programáveis. Neste capítulo serão apresentados os passos necessários para o desenvolvimento do trabalho, abrangendo tanto a configuração física quanto os componentes de hardware e software necessários para alcançar objetivo estabelecido no trabalho.

4.1 Criação da estrutura

A estrutura do projeto, mostrada na figura 15, parte de uma torneira ligada a uma mangueira que por sua vez é unida a uma válvula solenoide que realizará o controle da passagem de água. A válvula é conectada a outra parte de mangueira que possui um gotejador regulável e na sua extremidade um conector final. Lembrando que todas as conexões entre cada um dos componentes é feita através de conectores rápidos de mangueira.

Figura 15 – Visão esquemática do sistema de irrigação

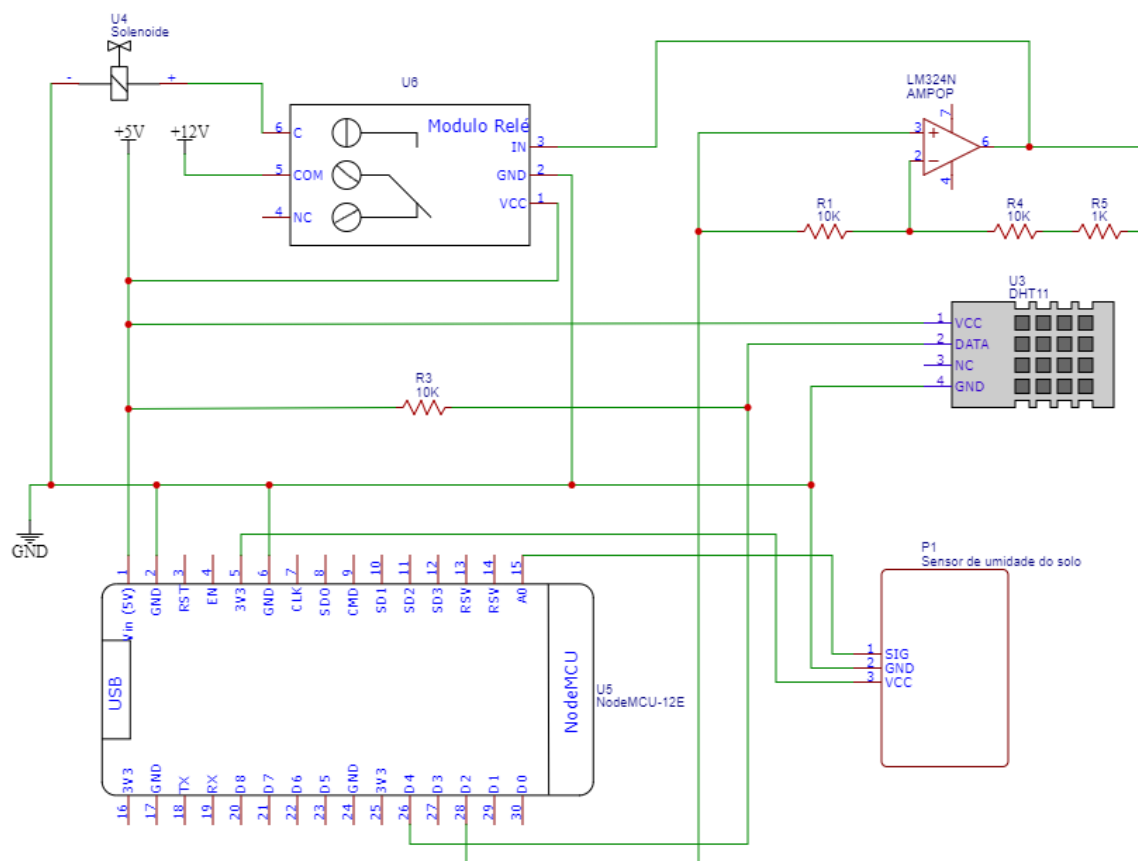


Fonte: Autor

4.2 Circuito eletrônico

O circuito eletrônico, observado na figura 16, foi criado utilizando o software online EasyEDA, acessível por meio do link (<https://easyeda.com/editor>). Ele foi dimensionado considerando a utilização de duas fontes externas, uma de 12V para alimentar a válvula solenoide e uma de 5V para alimentar o NodeMCU. O sensor de umidade do solo é alimentado pela tensão de 3,3V gerado pelo NodeMCU e conectado em A0, a única entrada analógica da placa. Já o sensor DHT11 que mede a umidade e temperatura do ar é alimentado pela fonte externa de 5V e conectado a porta digital D4 da placa. Para o acionamento da válvula solenoide foi utilizado um módulo relé em conjunto com um circuito amplificador operacional com operação em 5V que envia e recebe seus comandos através da porta digital D2.

Figura 16 – Diagrama do circuito elétrico utilizado no projeto



Fonte: Autor

Foi escolhido para o projeto, o sensor capacitivo de umidade do solo por sua particularidade de possuir uma camada externa anticorrosiva que aumenta significativamente a durabilidade do sensor se comparado com os tipos de sensores resistivos que devido ao contato direto com o solo, estão sujeitos a oxidação que além de influenciar na leitura do sensor prejudica também o pH do solo (CASTRO, 2023). Para utilizar esse sensor é necessário uma calibração devido as especificidades de cada solo, sendo assim foi feita uma média aritmética de 100 leituras do sensor com o solo totalmente úmido e 100 leituras com o solo seco e esses valores foram os limites de 0 e 100% de umidade.

O sensor DHT11 de umidade e temperatura do ar foi utilizado para que o usuário tenha visibilidade de forma gráfica como está o clima e como esse fator pode influenciar o crescimento saudável da planta e a quantidade de água necessária.

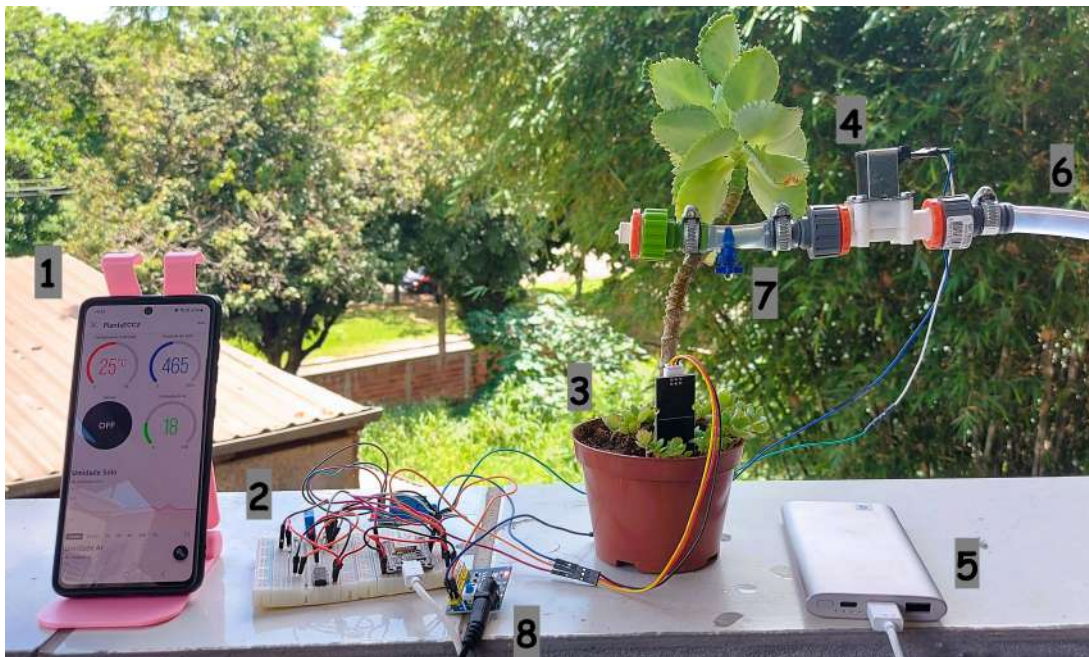
Estão incorporadas dentro do projeto três principais funções: irrigação programada por horário, irrigação manual e irrigação sensorial. Essas funções podem operar de forma independente ou em conjunto de acordo com a preferência do usuário. Para garantir o funcionamento do protótipo parte-se do princípio de que a torneira de água será mantida aberta e a válvula do circuito é normal fechada, sendo assim, a água só chegará ao gotejador quando for gerado um estímulo de abertura.

4.3 Implementação física do protótipo

O protótipo físico final do projeto pode ser observado na figura 17, que pode ser melhor detalhado através da numeração correspondente dentro da figura:

1. **Smartphone:** Exibe a interface do aplicativo Blynk, que monitora as condições ambientais, como a temperatura e umidade, e permite o controle do sistema de irrigação.
2. **Protoboard com componentes eletrônicos:** Contém diversos componentes eletrônicos conectados, responsáveis pela recepção de dados dos sensores e pelo controle da atuação do sistema de irrigação.
3. **Vaso com planta:** A planta que está sendo monitorada e irrigada pelo sistema. Junto à planta, está inserido o sensor de umidade do solo.
4. **Válvula solenoide:** Um atuador controlado eletronicamente que regula o fluxo de água para a irrigação da planta.
5. **Bateria externa:** Fornece energia para os componentes eletrônicos do sistema.
6. **Tubulação de irrigação:** Conduz a água da fonte até a planta, ativada pela válvula solenoide.
7. **Conector rápido de tubulação e gotejador regulável:** O conector permite a conexão e desconexão rápida da tubulação para manutenção ou reconfiguração do sistema. E o gotejador em azul permite a aplicação localizada e controlada de água diretamente na zona radicular das plantas.
8. **Módulo de alimentação:** Permite o ajuste preciso da tensão de saída para atender às diferentes necessidades de energia de um circuito eletrônico, sendo conectado a ele a fonte de 12V e sua saída de 5V alimenta o módulo relé.

Figura 17 – Protótipo físico do projeto



Fonte: Autor

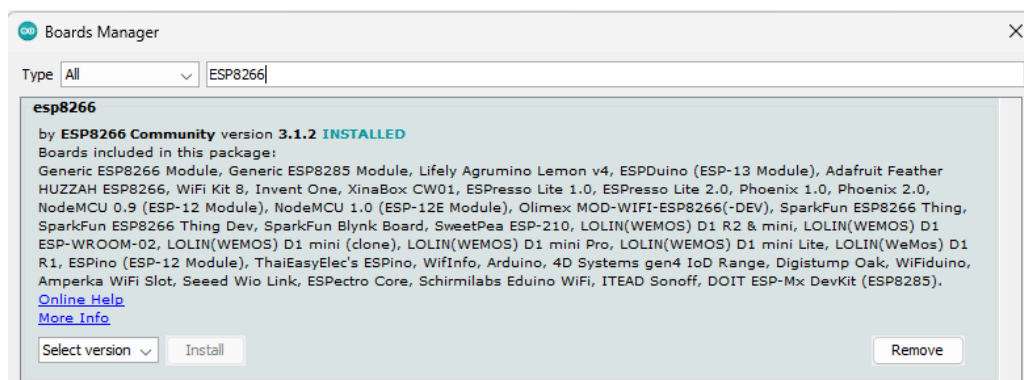
4.4 Implementação do software

A escolha da combinação das plataformas Blynk e ThingSpeak para trabalhar em conjunto com o Arduino oferece uma solução mais amigável e acessível ao usuário, especialmente quando comparada à programação direta e completa no Arduino. Primeiramente, o Blynk se destaca por sua interface gráfica intuitiva, permitindo aos usuários criar painéis personalizados com facilidade, sem necessidade de conhecimento avançado em programação. Isso torna o processo de controle e monitoramento dos dispositivos muito mais simples, uma vez que o usuário pode interagir com o sistema através de *widgets* visuais, como botões e *sliders*. O ThingSpeak complementa essa facilidade ao oferecer uma plataforma robusta para coletar e armazenar dados sensoriais, além de permitir a visualização desses dados em gráficos e análises em tempo real. Com o ThingSpeak, há ainda a possibilidade de uso da ferramenta MATLAB para processar e analisar os dados de forma mais personalizada, utilizando, por exemplo, gráficos avançados e análises específicas. A combinação dessas duas plataformas elimina a complexidade da escrita de códigos extensos no Arduino, proporcionando uma experiência mais amigável e menos técnica, ideal para usuários que não possuem expertise profunda em programação de microcontroladores, mas que desejam implementar projetos de automação eficientes e sofisticados.

Para programar o protótipo foi utilizada a IDE Arduino 1.8.19 que possui uma configuração simples e todas as bibliotecas necessárias para uma abordagem eficaz de conexão entre a placa e o Blynk.

Como primeiro passo, é preciso instalar, através do gerenciador de placas, a biblioteca do ESP8266 mostrado na figura 18.

Figura 18 – Biblioteca do ESP8266 no Gerenciador de Placas



Fonte: Autor

É preciso também a instalação de duas bibliotecas, uma para o sensor DHT11 e outra para conexão com o Blynk, conforme a figura 19.

Figura 19 – Bibliotecas DHT e Blynk



Fonte: Autor

Baseado na documentação fornecida no site oficial do Blynk foi possível entender e programar como seria feito o envio e sincronização dos dados, cujo código final desenvolvido para o projeto pode ser observado no Apêndice A deste trabalho.

4.4.1 Blynk Console

O Blynk Console é uma página web onde é possível configurar e gerenciar dispositivos e projetos IoT. Ele fornece uma interface para adicionar, configurar e monitorar os dispositivos conectados à plataforma Blynk, onde as principais funções e características relacionadas são:

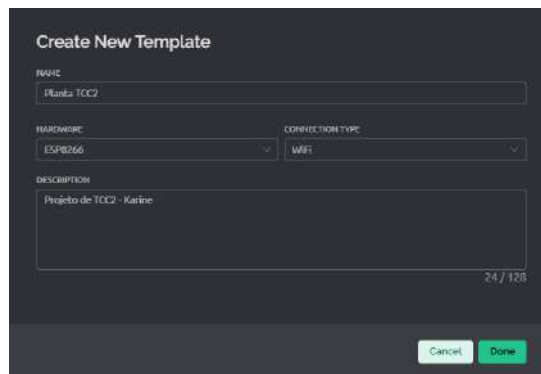
- Adição de Dispositivos: Permite registrar e adicionar novos dispositivos à sua conta Blynk para monitoramento e controle.
- Configuração de Projetos: Facilita a criação e gerenciamento de projetos IoT, onde é possível definir *widgets*, gráficos, botões e outros elementos de controle no Blynk para interagir com seus dispositivos.

- Integração de Hardware: Oferece suporte a uma variedade de plataformas de *hardware* para IoT, como Arduino, Raspberry Pi, ESP8266, ESP32, entre outros.
- Monitoramento Remoto: Permite monitorar e interagir com seus dispositivos conectados de forma remota através do aplicativo Blynk.

4.4.1.1 Template

Para a configuração do Blynk foi preciso primeiramente criar uma conta na página web de cadastro com um e-mail válido ou diretamente no próprio aplicativo mobile. Com o cadastro feito, é preciso criar um template selecionando a placa controladora e o tipo de conexão a serem utilizados, como é mostrado na figura 20.

Figura 20 – Criação de um template



Fonte: Autor

4.4.1.2 Datastreams

O próximo passo seria a configuração dos *datastreams*, que são as variáveis que vão receber os valores enviados pelos sensores, o comando da válvula e o webhook que será enviado ao ThingSpeak. Para configurar um *datastream* deve ser escolhido um nome, o tipo de dado, o tipo de unidade, valor máximo e mínimo, dentre outros. Após configurado basta salvá-lo e utilizar o valor do Virtual PIN dentro do código desenvolvido no Arduino IDE. Os *datastreams* criados para o protótipo podem ser visualizados na figura 21.

Figura 21 – *Datastreams* utilizados no protótipo

Id	Name	Alias	Color	Pin	Data Type	Units	In Row	Min	Max	Decimals	Default Value
3	Temperatura Ambiente	Temperatura Ambiente	Grey	V2	Integer	°C	false	0	50	—	0
2	Umidade Ar	Umidade Ar	Teal	V1	Integer		false	0	100	—	0
1	Umidade Solo	Umidade Solo	Blue	V0	Integer		false	0	1024	—	0
4	Válvula	Válvula	Orange	V3	Integer		false	0	1	—	0
6	Webhook	Webhook	Purple	V4	String		false			—	

Fonte: Autor

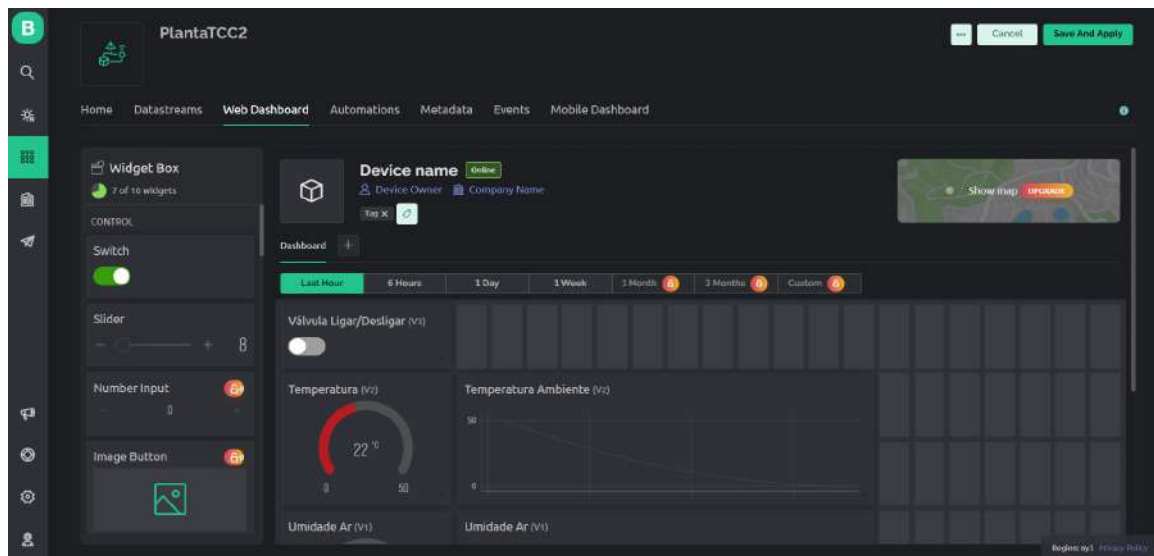
4.4.1.3 Web Dashboard

O *web dashboard* será utilizado para a visualização dos dados dos sensores e controle manual do usuário sobre a válvula. O Blynk na sua versão gratuita conta com vários *widgets* que podem ser utilizados de acordo com o tipo de dado e organizados no painel de maneira a oferecer a melhor experiência aos usuários. No protótipo foram escolhidos os seguintes *widgets*:

- *Switch* - Para permitir ao usuário o controle manual da válvula;
- *Gauge* - Para apresentar o valor da leitura mais recente do sensor;
- *Chart* - Para apresentar em formato de gráfico os valores coletados dentro de um intervalo de tempo, que pode ser desde a ultima hora até todos os dados da última semana.

É possível observar na figura 22 no canto esquerdo uma caixa escrito *Widget Box* que contém todos os *widgets* disponíveis para uso, sendo alguns disponíveis apenas na versão paga. Ao lado da caixa de *Widgets* encontramos o *dashboard* onde o usuário pode arrastar os *widgets* e posicioná-los da maneira que desejar.

Figura 22 – Web Dashboard



Fonte: Autor

4.4.1.4 Automations

A fim de permitir ao usuário usufruir das funções de ligar/desligar a válvula de maneira automatizada por valores máximos e mínimos de umidade do solo ou através de um período de tempo programado, será utilizado o *Automations* do Blynk. Antes de criar a automação de fato é necessário preparar os *datastreams* dentro de configurações do template na aba *Automations*. Sendo as opções disponíveis detalhadas na documentação da plataforma Blynk. Para controlar a válvula automaticamente dentro das funções propostas nesse protótipo foi preciso selecionar o *datastream* de umidade do solo como uma condição e o *datastream* da válvula como condição e ação como é mostrado na figura 23.

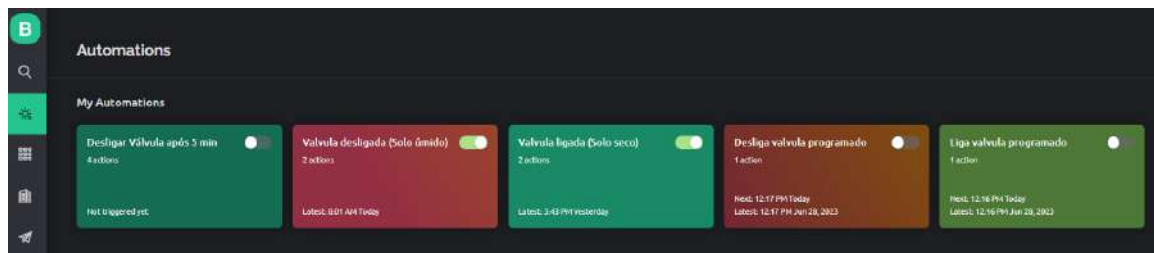
Figura 23 – Configuração de *automations* para os *datastreams* do protótipo

Name	Pin	Data Type	Type Of Automation	Condition	Action
Temperatura Ambiente	V2	Integer	Switch	☐	☐
Umidade Ar	V1	Integer	Switch	☐	☐
Umidade Solo	V0	Integer	Sensor	☒	☐
Válvula	V3	Integer	Switch	☒	☒
Webhook	V4	String	Color	☐	☐

Fonte: Autor

Todas as automações criadas para o projeto podem ser visualizadas na aba *Automations* assim como mostra na figura 24. Essas automações podem ser ligadas ou desligadas e alteradas facilmente dependendo das necessidades do usuário.

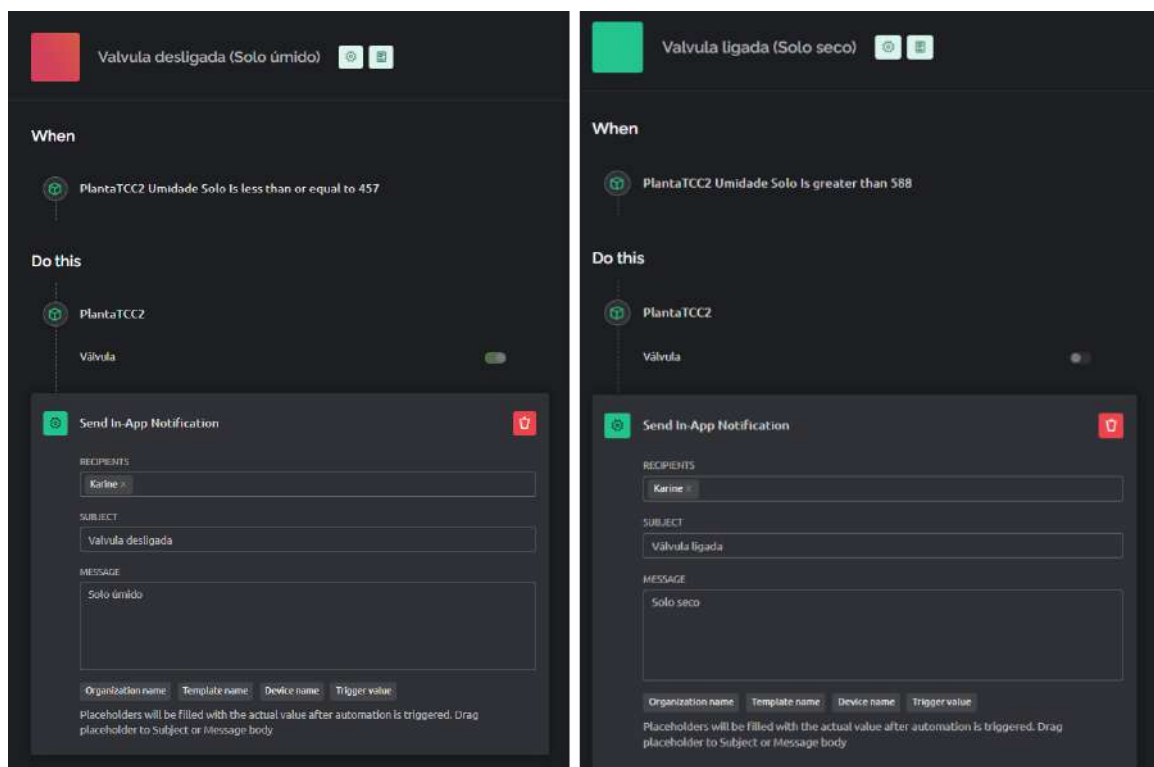
Figura 24 – Automações configuradas para o projeto



Fonte: Autor

A automação definida por valores do sensor de umidade do solo leva em consideração os valores coletados para definir os limites em que o usuário deseja ligar ou desligar a válvula. No projeto foi decidido utilizar o sensor de umidade analógico capacitivo que tem um range de 0 a 1024, sendo que quanto mais próximo de 0, mais úmido o solo está. Foi considerado que a válvula deveria ligar quando o solo estivesse abaixo de 30% de umidade e desligar quando ele atingisse 70% de umidade, para isso foram feitas várias medições com o solo bem úmido e o solo seco para definir os valores máximos e mínimos do sensor para então calcular os valores a serem inseridos na automação. Na figura 25 é possível visualizar a programação para coordenar a válvula de acordo com os valores obtidos no sensor de umidade do solo, além de enviar ao final do processo uma notificação no smartphone do usuário, que pode ser alterado também para o envio de um e-mail.

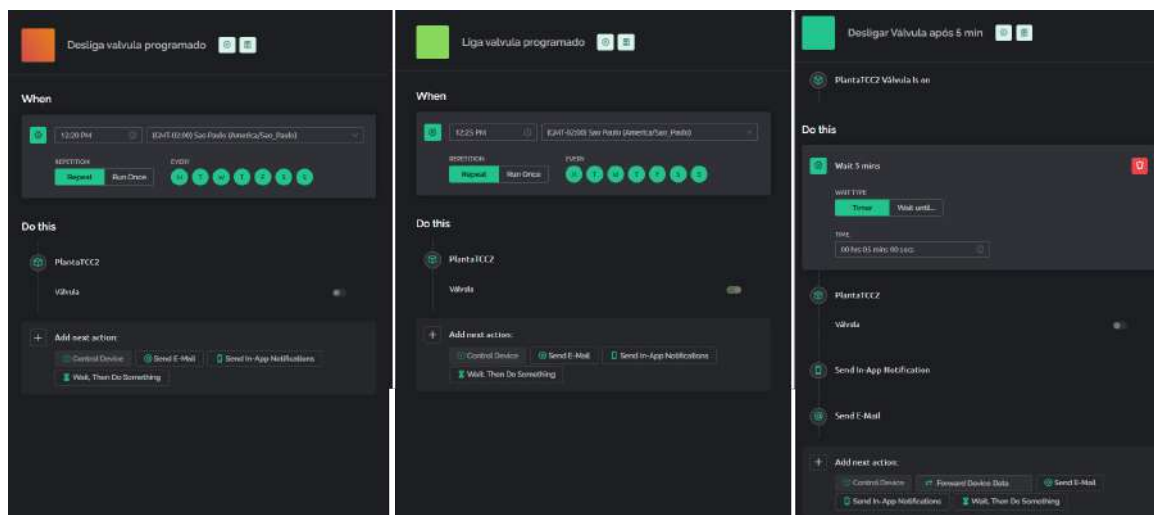
Figura 25 – Configuração da automação para válvula definida por umidade do solo



Fonte: Autor

A automação definida por período de tempo programado demonstrada na figura 26 pode ser feita de duas maneiras. A primeira delas seria configurando um dia e horário para que a válvula seja ligada e desligada, já a segunda seria uma combinação entre ligar a válvula em um dia e horário específicos e programar para que ela seja desligada após um determinado tempo, exemplificado na lateral direita da figura 26.

Figura 26 – Configuração da automação para válvula definida por período de tempo

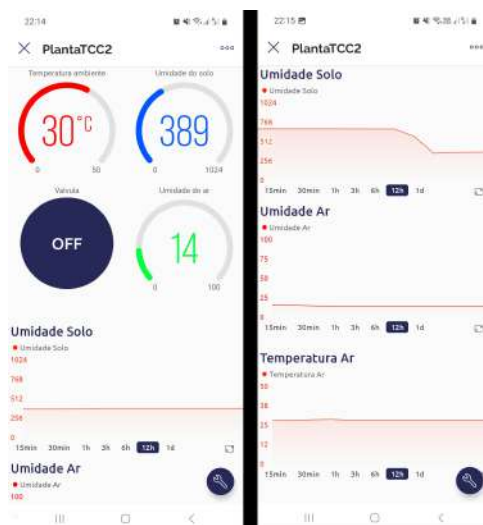


Fonte: Autor

4.4.2 Blynk App

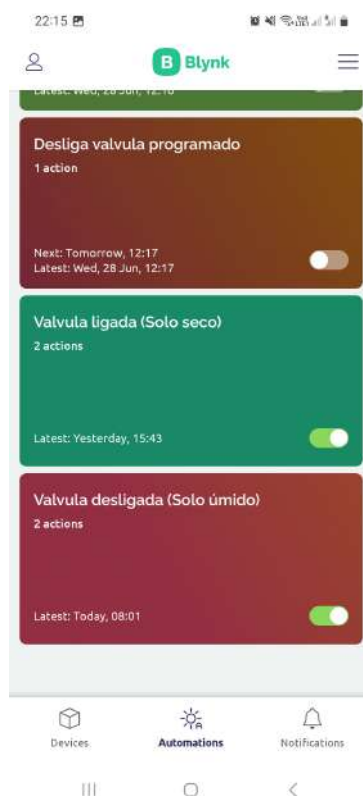
O projeto dentro do aplicativo mobile do Blynk pode ser criado a partir de um template feito anteriormente através da página web ou pode ser desenvolvido diretamente pelo aplicativo. A disposição dos widgets e a configuração de design para melhor visualização podem ser aplicados de maneira bem simples. A figura 27 contém a interface do aplicativo deste projeto com os gráficos dos sensores e seus respectivos valores. As automações também podem ser ativadas, desativadas e ajustadas facilmente por meio do aplicativo, como é mostrado na figura 28.

Figura 27 – Interface Blynk no aplicativo mobile



Fonte: Autor

Figura 28 – Aba de automações no aplicativo Blynk



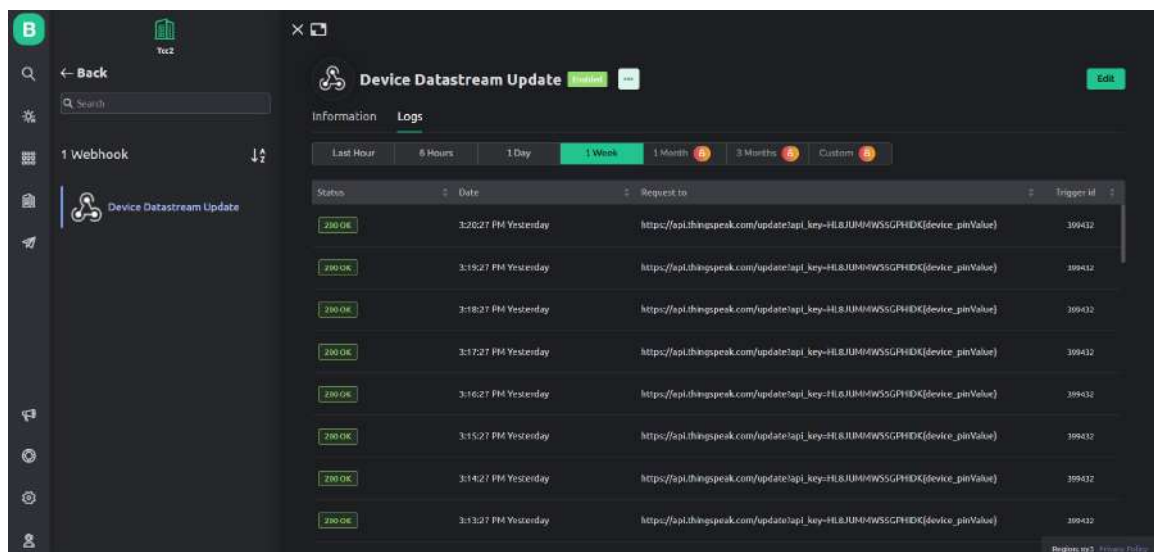
Fonte: Autor

4.4.3 ThingSpeak

Devido às limitações da versão gratuita do Blynk que mantém o histórico da coleta dos dados apenas da última semana e de não ter a opção de exportar esses dados em um CSV

(Comma-Separated Values) ou utilizar o MatLab para melhor visualização e possível criação de gráficos detalhados, optou-se por utilizar o Blynk em conjunto com o ThingSpeak. Existe dentro da plataforma Blynk a opção de criação de um webhook, que será usado para enviar os dados dos sensores para o ThingSpeak por meio de um evento que ocorre a cada nova leitura dos sensores, essa chamada está contida dentro do código desenvolvido no Arduino IDE e para utilizá-la é preciso configurar o Blynk e utilizar API Key que é fornecida quando criamos um canal dentro do ThingSpeak. A figura 29 permite visualizar os logs de chamadas feitas para a URL do ThingSpeak contendo a API Key.

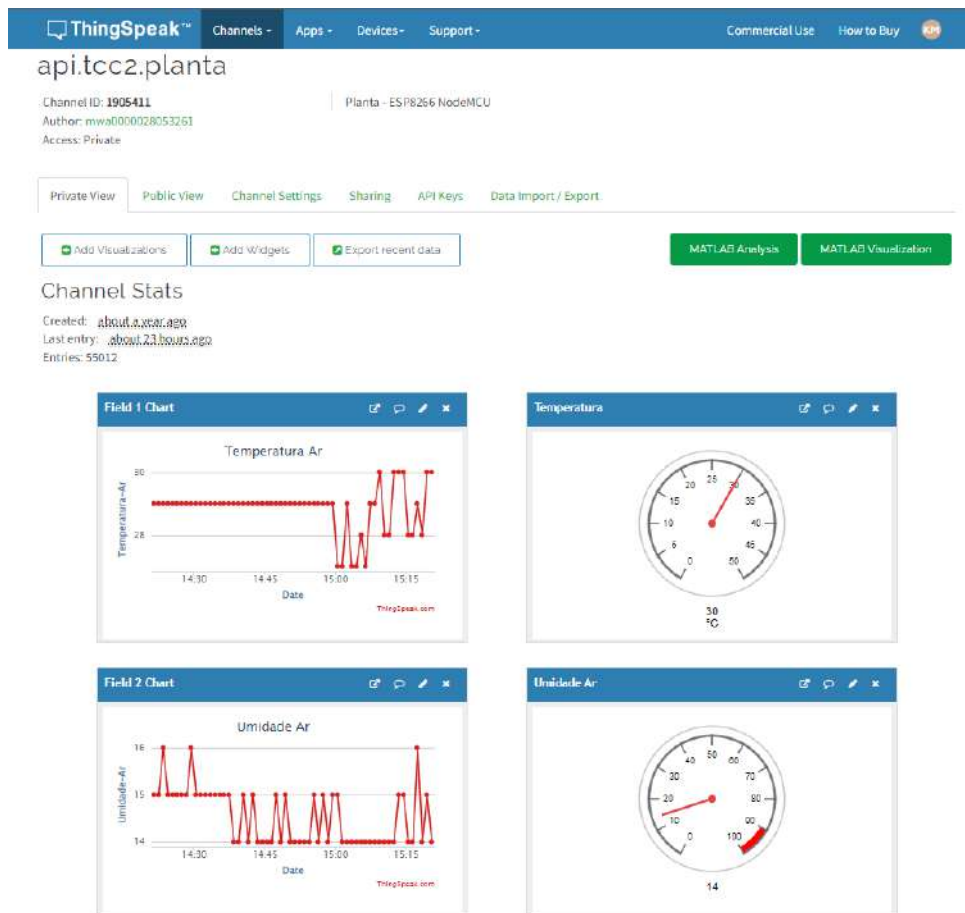
Figura 29 – Logs de chamads do webhook no console Blynk



Fonte: Autor

O ThingSpeak oferece também a possibilidade de criar gráficos e utilizar outros widgets para analisar os dados. No protótipo deste trabalho foi criado uma interface parecida com a do Blynk, como é mostrado na figura 30. Para ter posse de todos os dados desde o início das medições, basta ir na aba de Data Import/Export e exportar o CSV disponível.

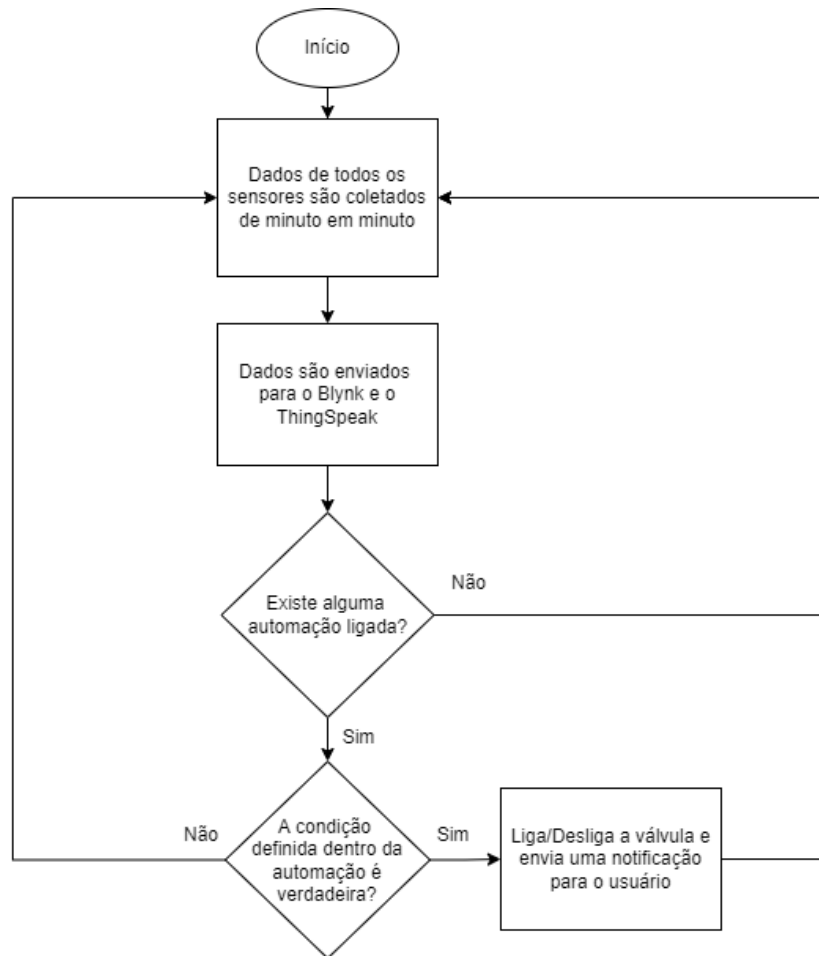
Figura 30 – Página web ThingSpeak contendo o canal do projeto



Fonte: Autor

Um fluxograma foi desenvolvido, como apresentado na figura 31, para facilitar a compreensão dos passos envolvidos no projeto, desde a coleta de dados pelos sensores até a ativação ou desativação da válvula. É importante ressaltar que o fluxograma não inclui o acionamento ou desligamento manual da válvula, pois esta operação pode ser realizada pelo usuário a qualquer momento, mediante um simples clique no botão On/Off apresentado na figura 27.

Figura 31 – Fluxograma das etapas desde a obtenção de dados até o controle da válvula



Fonte: Autor

5 Testes e Resultados

Este capítulo apresenta os testes realizados e os resultados obtidos através do desenvolvimento do protótipo de um sistema de irrigação residencial, bem como discussões pertinentes a todo o processo de desenvolvimento do projeto.

5.1 Discussão

No início do projeto, optou-se pelo uso de um sensor de umidade do solo do tipo resistivo. No entanto, esta escolha logo se mostrou problemática, pois as medições evidenciavam disparidades significativas e incoerências, atribuíveis primordialmente à oxidação nas hastes do sensor. Esta limitação do sensor resistivo impulsionou a reconsideração do método de medição utilizado, sendo um ponto crítico para a precisão dos dados coletados.

Em resposta aos desafios identificados, decidiu-se substituir o sensor resistivo por um modelo capacitivo. Este último oferece uma proteção aprimorada contra os elementos, reduzindo o risco de oxidação e melhorando a confiabilidade das medições. Contudo, esta solução não se mostrou isenta de falhas. Um incidente no qual água foi acidentalmente derramada sobre a parte eletrônica do sensor resultou em sua inutilização. Tal ocorrência evidenciou a necessidade de uma solução de contingência para o projeto.

Com a finalidade de prevenir interrupções futuras no processo de coleta de dados, procedeu-se à aquisição de dois sensores capacitivos. Esta medida preventiva visava assegurar a continuidade do projeto frente a eventuais imprevistos. Paralelamente, uma análise do código de programação revelou uma limitação adicional: a necessidade de recalibrar o sistema a cada substituição do sensor. Esta recalibração implicava na coleta de 200 valores distintos de umidade (100 com o solo úmido e 100 com o solo seco), seguida pela atualização manual dos valores no código.

Para mitigar este problema, optou-se por uma abordagem mais flexível na programação. Ao invés de utilizar uma porcentagem fixa para o valor de umidade do solo, decidiu-se incorporar a leitura analógica direta do sensor, que varia de 0 a 1024. Esta mudança metodológica permitiu eliminar a necessidade de ajustes manuais no código em caso de substituição do sensor. Adicionalmente, implementou-se a automação desta leitura através da plataforma Blynk, proporcionando um método mais eficiente e menos suscetível a erros no processo de coleta e análise de dados.

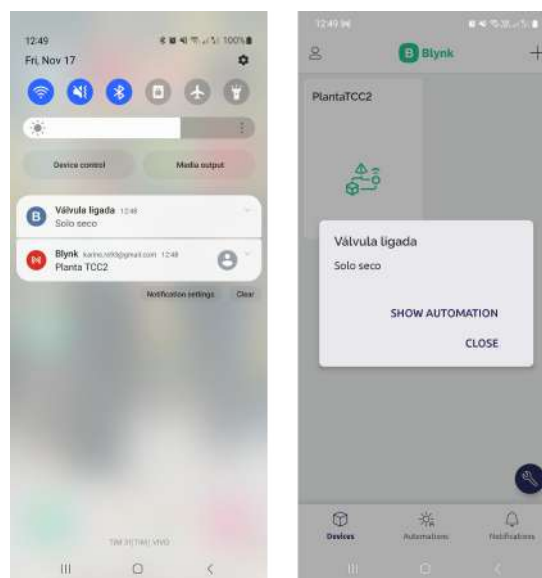
5.2 Testes

Para todos os testes foi utilizado a planta suculenta, sendo que no primeiro teste relacionado à automação baseado nos valores do sensor de umidade do solo, foram definidos como início de irrigação um valor de 30% de umidade e a irrigação cessaria quando atingisse os 70%

de umidade do solo. Sendo assim, o objetivo do primeiro teste era confirmar que quando o valor do sensor de umidade do solo atingisse 30% de umidade, a válvula seria ligada, uma notificação enviada ao *smartphone* com o Blynk instalado e um e-mail chegariam ao usuário. Da mesma forma que quando a umidade do solo atingisse 70% o oposto aconteceria. Para tal, foi necessário primeiro observar o valor do sensor no solo seco e no solo úmido para definir os limites máximo e mínimo. Com o valor obtido é possível fazer um cálculo utilizando a regra de três para definir os valores que serão inseridos na aba de automações do Blynk.

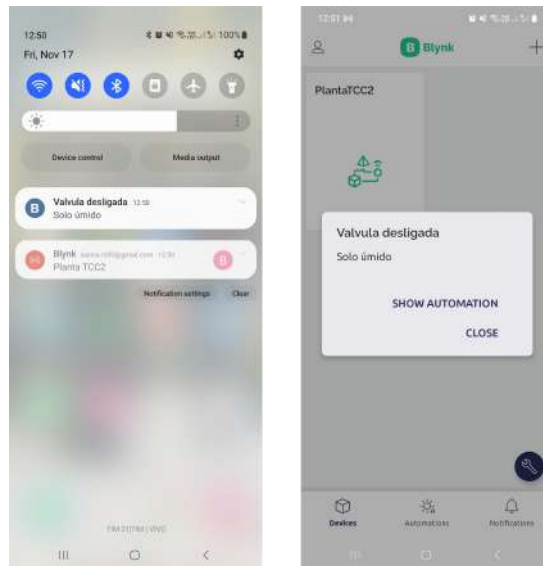
Para realizar o teste o sensor de umidade do solo foi inserido primeiro em um solo úmido e a válvula estava desligada, logo em seguida foram ligadas as automações e o sensor foi retirado do vaso de planta e registrou-se um valor abaixo de 30% de umidade o que fez com que a válvula fosse ligada e o Blynk enviasse uma notificação e um e-mail avisando, como pode ser visto na figura 32. Logo após as notificações o sensor foi inserido novamente no solo úmido e assim como esperado, a válvula foi desligada e foram enviadas a notificação e o e-mail informado o ocorrido, conforme a figura 33.

Figura 32 – Notificações relacionadas à válvula ligada



Fonte: Autor

Figura 33 – Notificações relacionadas à válvula desligada



Fonte: Autor

O teste descrito acima pode também ser observado através dos gráficos no Blynk Console da figura 34. Lembrando que quanto menor for o valor do sensor, mais úmido o solo estará.

Figura 34 – Interface Blynk Console após o teste de umidade do solo



Fonte: Autor

Outro teste realizado foi a comparação dos valores de medição de temperatura do sensor DHT11, figura 35, e da temperatura indicada por um site de medições climáticas, figura

36. A comparação foi feita durante algumas horas, porém as imagens mostram apenas um período de 2 horas e 40 minutos, sendo que a medição do sensor apresentou diferenças entre a medição do site que eram esperadas devido a alguns fatores, como:

- O DHT11 é um sensor de baixo custo e não é tão preciso quanto outros sensores mais sofisticados.
- A localização física do sensor DHT11 pode influenciar suas leituras. No caso ele se encontra na sombra.
- Os sites de clima geralmente usam dados de várias estações meteorológicas e modelos climáticos para fornecer uma média da temperatura em uma área mais ampla.
- O clima pode variar significativamente em pequenas distâncias, especialmente em áreas urbanas com muitos prédios e concreto, ou em regiões com geografias variadas.

Figura 35 – Medição de temperatura no Blynk Console



Fonte: Autor

Figura 36 – Medição de temperatura hora a hora

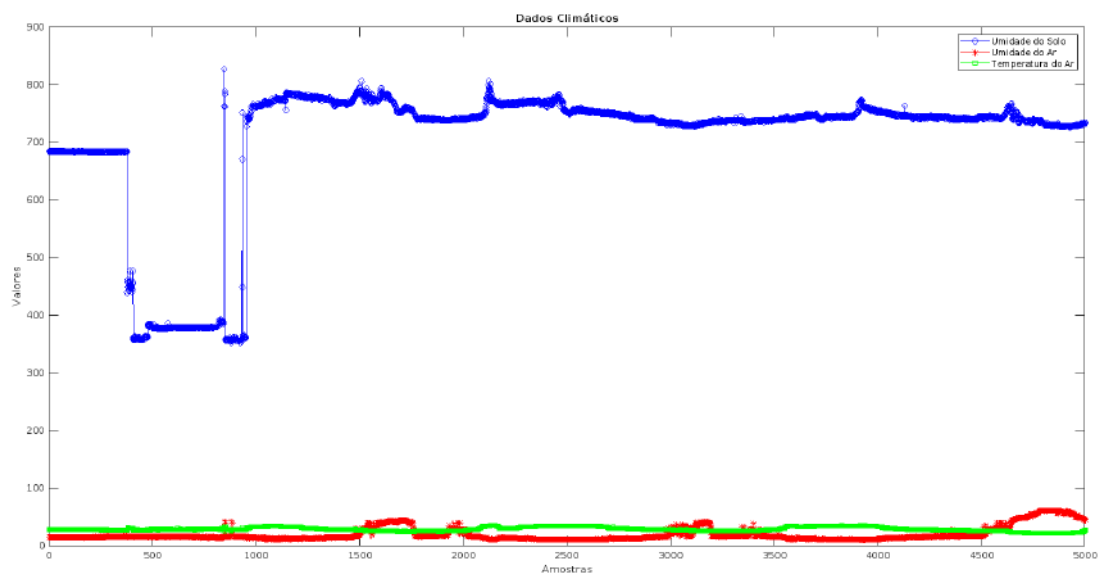


Fonte: (CHANNEL, 2023)

5.3 Resultados

A utilização do MATLAB para visualizar e interpretar dados de sensores ambientais mostrou-se eficaz. O primeiro gráfico gerado, observado na figura 37, sintetizou as leituras de todos os sensores, proporcionando uma visão agregada das condições ambientais. Através desta visualização, é possível observar a interação entre a umidade do solo e as variáveis atmosféricas, como a umidade e a temperatura do ar. A capacidade do MATLAB de manipular e representar grandes conjuntos de dados permite a geração de gráficos detalhados que facilitam o reconhecimento de padrões e a compreensão de fenômenos complexos. Este gráfico não apenas confirmou a funcionalidade dos sensores mas também enfatizou a capacidade do MATLAB de atuar como uma ferramenta robusta para análises ambientais.

Figura 37 – Gráfico com os valores de todos os sensores



Fonte: Autor

Foi criado também um segundo gráfico focado nos dados do sensor DHT11, especializado em medições de umidade e temperatura do ar, exemplificou a precisão com que o MATLAB pode tratar dados específicos. A representação visual das variações temporais de umidade e temperatura forneceu *insights* valiosos sobre as flutuações diárias e sazonais. Esse gráfico pode ser visualizado na figura 38 juntamente com o restante dos gráficos e leituras na página do ThingSpeak. A flexibilidade do MATLAB em adaptar-se às necessidades do usuário, demonstraram a versatilidade do MATLAB não apenas como uma ferramenta de visualização, mas também como um meio eficiente para realizar análises exploratórias e deduzir relações significativas a partir de dados ambientais.

Figura 38 – Página do ThingSpeak com os gráficos gerados



Fonte: Autor

6 Conclusão

“A ciência é uma maneira de pensar muito mais do que um conjunto de conhecimentos.”

Carl Sagan

O objetivo de criação do protótipo proposto neste trabalho foi alcançado. Ao considerar os objetivos específicos do projeto, como o dimensionamento de componentes eletrônicos, a construção de protótipos de baixo custo e a elaboração de um sistema de comunicação eficiente entre smartphone e dispositivo NodeMCU ESP8266, foi possível criar uma solução prática e eficiente. A capacidade de realizar a irrigação de maneiras variadas (manual, programada ou através de automações) oferece ao usuário flexibilidade e controle, enquanto a coleta e exibição de dados em um *dashboard* proporciona *insights* valiosos sobre a eficácia dos diferentes métodos de irrigação.

A integração das plataformas Blynk e ThingSpeak com o Arduino representa um avanço significativo na acessibilidade e na experiência do usuário. Essa combinação de tecnologias oferece uma interface mais amigável do que a programação direta no Arduino, tornando o controle da irrigação mais acessível a usuários sem conhecimentos técnicos avançados.

Contudo, foram enfrentados desafios relacionados às limitações da versão gratuita do Blynk, como a restrição no armazenamento de dados históricos e a ausência de ferramentas para exportação e análise detalhada dos dados. Para contornar essas limitações, a utilização conjunta do Blynk com o ThingSpeak mostrou-se uma solução eficaz, permitindo o armazenamento prolongado de dados e a possibilidade de análise aprofundada através da criação de gráficos detalhados e utilização de ferramentas como o MATLAB.

O histórico de irrigação acessível ao usuário desempenha um papel crucial, permitindo comparações e ajustes com base no desenvolvimento das plantas, tornando o cuidado com as plantas uma tarefa mais gerenciável e menos sujeita ao esquecimento ou à falta de tempo. Essa funcionalidade não só atende às necessidades dos usuários modernos, mas também oferece uma oportunidade educacional para compreender melhor os requisitos de cuidado para diferentes tipos de plantas.

O protótipo desenvolvido neste trabalho vai além da simples automação residencial de cuidados com plantas. Com algumas expansões, ele demonstra um potencial significativo para aplicações em contextos mais amplos, como em pequenas propriedades agrícolas e laboratórios de testes. Para pequenos agricultores, essa tecnologia pode representar uma oportunidade de otimizar a irrigação e o monitoramento de culturas, contribuindo para a eficiência na utilização de recursos e na gestão do tempo. Já em ambientes de laboratório, o protótipo pode ser usado para estudos precisos e controlados sobre as necessidades hídricas de diferentes espécies vegetais, fornecendo dados valiosos para a pesquisa agrônômica e botânica.

6.1 Trabalhos futuros

Primeiramente, propõe-se a implementação de um design mais sofisticado, que incluiria a impressão do circuito em uma placa de circuito impresso (PCI) customizada, promovendo uma melhor organização. Adicionalmente, contempla-se a integração de uma carcaça robusta, possivelmente fabricada em acrílico ou outro material de alta durabilidade, para proteger o dispositivo contra deterioração ambiental, incluindo danos causados por intempéries e outros fatores externos.

Um aspecto crucial deste aperfeiçoamento é a inclusão de uma interface de usuário mais intuitiva e acessível, como a adição de uma chave ou botão ergonômico. Isso permitirá a ativação e desativação manual da válvula, assegurando a continuidade da irrigação mesmo em cenários de falhas de conectividade de rede.

A incorporação de uma lâmpada de espectro completo, que imita a luz solar, poderia ser particularmente benéfica para ambientes internos onde a luz natural é limitada. Esta melhoria no sistema poderia permitir um controle mais preciso sobre as condições de crescimento das plantas, levando a resultados ainda mais otimizados.

Por fim, vislumbra-se a expansão do projeto para controlar um número maior de válvulas, o que exigirá a integração de um módulo com capacidades aprimoradas, incluindo um número maior de entradas analógicas para acomodar sensores adicionais de umidade do solo. Este avanço permitirá uma gestão mais abrangente e eficaz da irrigação em uma escala ampliada.

Referências

- ACCARDI, A.; DODONOV, E. Automação residencial: elementos básicos, arquiteturas, setores, aplicações e protocolos. *Revista TIS*, v. 1, n. 2, 2012. Citado na página 11.
- ADAFRUIT. *DHT11, DHT22 and AM2302 Sensors*. 2019. <<https://www.mouser.com/datasheet/2/737/dht-932870.pdf>>. (Acessado em 06/21/2022). Citado na página 23.
- ADOLPHS, R. S. Projeto: sistema de irrigação automático supervisionado para hortas. 2020. Citado nas páginas 24 e 25.
- ANDRADE, C. Seleção do sistema de irrigação. *Embrapa Milho e Sorgo. Circular técnica*, Sete Lagoas: Embrapa Milho e Sorgo., 2001. Citado na página 14.
- ANDRADE, C.; BRITO, R. Métodos de irrigação e quimigação. *Embrapa Milho e Sorgo-Circular Técnica (INFOTECA-E)*, Sete Lagoas: Embrapa Milho e Sorgo, 2006., 2006. Citado na página 14.
- BARBOSA, J. W. Sistema de irrigação automatizado utilizando a plataforma arduino. *Trabalho de conclusão de curso, Fundação Educacional do Município de Assis-FEMA-Assis*, 2013. Citado na página 24.
- BARREIRO, A. P. et al. Análise de crescimento de plantas de manjerição tratadas com reguladores vegetais. *Bragantia*, SciELO Brasil, v. 65, p. 563–567, 2006. Citado na página 11.
- BISCARO, G. A. Sistemas de irrigação localizada. Universidade Federal da Grande Dourados, 2014. Citado nas páginas 11, 12 e 15.
- BLYNK. *Blynk Documentation*. 2023. <<https://docs.blynk.io/en/>>. (Acessado em 02/10/2023). Citado na página 18.
- BLYNK. *Blynk Pricing*. 2023. <<https://blynk.io/pricing>>. (Acessado em 02/10/2023). Citado nas páginas 20 e 21.
- CARVALHO, A. V. P. d. et al. Projeto social em uma comunidade carente na cidade de Recife (pe): Horta caseira e seus benefícios. *REVISTA DE TRABALHOS ACADÊMICOS-UNIVERSO RECIFE*, v. 5, n. 2, 2019. Citado na página 11.
- CASTRO, G. *Leitura da Umidade do Solo*. 2023. <<https://www.robocore.net/tutoriais/leitura-umidade-solo>>. (Acessado em 02/10/2023). Citado nas páginas 22 e 32.
- CASTRO, L. S. Tratamento de sementes e déficit hídrico em plantas de manjerição: abordagens fisiológica e bioquímica. Universidade Federal de Viçosa, 2019. Citado na página 11.
- CHANNEL, T. W. 2023. <<https://weather.com/pt-AO/clima/horaria//4b637d03fe55f1fa8d3e89185c12a5a779f45727f164f8c74ee71565d9a20c9d>>. (Acessado em 18/11/2023). Citado na página 48.
- COELHO, E.; SIMOES, W. Onde posicionar sensores de umidade e de tensão de água do solo próximo da planta para um manejo mais eficiente da água de irrigação. *Embrapa Mandioca e Fruticultura-Circular Técnica (INFOTECA-E)*, Cruz das Almas, BA: Embrapa Mandioca e Fruticultura, 2015., 2015. Citado na página 11.

COMPONENTS101. *NodeMCU ESP8266 Pinout, Specifications, Features & Datasheet*. 2020. <<https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>>. (Acessado em 06/19/2022). Citado na página 17.

CONTRIBUTORS, M. *Uma visão geral do HTTP*. 2023. <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP>>. (Acessado em 02/10/2023). Citado na página 17.

ELETROGATE. *Módulo WiFi ESP8266 NodeMcu ESP-12E - Eletrogate | Arduino, Robótica, IoT, Apostilas e Kits*. 2022. <https://www.eletrogate.com/modulo-wifi-esp8266-nodemcu-esp-12e?utm_source=Site&utm_medium=GoogleMerchant&utm_campaign=GoogleMerchant&gclid=Cj0KCQjwntCVBhDdARIsAMEwACm-IBj4-0O0PVqP1cEzjGQlgb10_yQiD8BGCSLio5aScY0nH7L4fw8aAqpgEALw_wcB>. (Acessado em 06/23/2022). Citado na página 23.

FILHO, M. F. *Internet das coisas. Unisul Virtual*, 2016. Citado na página 15.

GERHARDT, T. E.; SILVEIRA, D. T. *Métodos de pesquisa*. [S.l.]: Plageder, 2009. 31–42 p. Citado na página 27.

GIOMO, D. *Desenvolvimento de um sistema de irrigação automatizado de baixo custo*. Universidade Estadual do Oeste do Paraná, 2019. Citado nas páginas 25 e 26.

MAGRANI, E. *A internet das coisas*. [S.l.]: Editora FGV, 2018. Citado na página 15.

OLIVEIRA, S. de. *Internet das coisas com ESP8266, Arduino e Raspberry Pi*. [S.l.]: Novatec Editora, 2017. Citado nas páginas 16 e 23.

REDHAT. *Webhook*. 2022. <<https://www.redhat.com/pt-br/topics/automation/what-is-a-webhook>>. (Acessado em 02/10/2023). Citado na página 18.

SANTOS, L. M. F. dos. *Irrigação em hortas escolares ou comunitárias. ALIMENTAÇÃO SAUDÁVEL E SUSTENTABILIDADE AMBIENTAL NAS ESCOLAS DO PARANÁ*, p. 217, 2014. Citado na página 11.

SERRANO, T. *Introdução ao Blynk App*. 2018. <<https://embarcados.com.br/introducao-ao-blynk-app/#:~:text=O%20App%20Blynk%20%C3%A9%20um,que%20interagem%20com%20o%20hardware>>. (Acessado em 02/10/2023). Citado nas páginas 18, 19, 20 e 22.

THINGSPEAK. *Learn More About ThingSpeak*. 2023. <https://thingspeak.com/pages/learn_more>. (Acessado em 02/10/2023). Citado na página 17.

ZANATTA, B. *Casa “inteligente” é cada vez mais realidade*. O Estado de S.Paulo, 2021. Disponível em: <http://www.aureside.org.br/_pdf/oesp_170121.pdf>. Citado na página 11.

APÊNDICE A – Código desenvolvido no Arduino IDE

```

1 #define BLYNK_TEMPLATE_ID      "TMPL2RzjYZko8"
2 #define BLYNK_TEMPLATE_NAME    "PlantaTCC2"
3 #define BLYNK_AUTH_TOKEN       "UrvsSoKUJgnIZfe1z6dYd4vEFUXpbXnUA"
4 #include <ESP8266WiFi.h>
5 #include <BlynkSimpleEsp8266.h>
6 #include "DHT.h"
7
8 #define DHTTYPE DHT11 //Define o tipo de sensor DHT vai ser utilizado
9
10 //defines - mapeamento de pinos do NodeMCU
11 #define D0      16
12 #define D1      5
13 #define D2      4
14 #define D3      0
15 #define D4      2
16 #define D5      14
17 #define D6      12
18 #define D7      13
19 #define D8      15
20 #define D9      3
21 #define D10     1
22
23 #define VPIN_UMIDADESOLO        V0
24 #define VPIN_UMIDADEAR          V1
25 #define VPIN_TEMPERATURA        V2
26 #define VPIN_VALVULA            V3
27 #define VPIN_WEBHOOK            V4
28
29 char ssid[] = "2.4G-CLARO-01";
30 char pass[] = "11111111";
31 BlynkTimer timer;
32
33 const int DHTPin = 2;          // Pino utilizado pelo DHT11
34 const int valvula = 4;         // Pino utilizado pela valvula
35 const int pinSensor = A0;     // Pino utilizado pelo sensor
36 DHT dht(DHTPin, DHTTYPE);
37
38
39 BLYNK_CONNECTED() {
40   Blynk.syncVirtual(VPIN_UMIDADESOLO);
41   Blynk.syncVirtual(VPIN_UMIDADEAR);

```

```
42 Blynk.syncVirtual(VPIN_TEMPERATURA);
43 Blynk.syncVirtual(VPIN_VALVULA);
44 Blynk.syncVirtual(VPIN_WEBHOOK);
45 }
46
47 BLYNK_WRITE(VPIN_VALVULA)
48 {
49     int button = param.asInt();
50     if (button == 1)
51     {
52         digitalWrite(valvula, HIGH);
53     } else {
54         digitalWrite(valvula, LOW);
55     }
56 }
57
58 void setup() {
59     Serial.begin(115200);
60     pinMode(valvula, OUTPUT);
61     Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
62     dht.begin(); // Leitura do sensor DHT11
63     timer.setInterval(60000L, sensors);
64 }
65
66 //Funcao para leitura dos valores dos sensores e envio para o Blynk e o
67 //ThingSpeak
68 void sensors() {
69     int umidadeSolo = analogRead(pinSensor);
70     int temperaturaAmbiente = dht.readTemperature();
71     int umidadeAr = dht.readHumidity();
72     Blynk.virtualWrite(VPIN_UMIDADESOLO, umidadeSolo);
73     if (isnan(temperaturaAmbiente) || isnan(umidadeAr)) {
74         //Serial.println("Failed to read from DHT sensor!");
75         return;
76     } else {
77         Blynk.virtualWrite(VPIN_TEMPERATURA, temperaturaAmbiente);
78         Blynk.virtualWrite(VPIN_UMIDADEAR, umidadeAr);
79     }
80     String tempAr = String(temperaturaAmbiente);
81     String umiAr = String(umidadeAr);
82     String umiSolo = String(umidadeSolo);
83     String webhook = "&field1="+tempAr+"&field2="+umiAr+"&field3="+umiSolo
84         ;
85     Blynk.virtualWrite(V4, webhook); //webhook que enviar os valores de
86     // todos os sensores para o ThingSpeak
87 }
88
89 void loop() {
```

```
87   Blynk.run();  
88   timer.run();  
89 }
```

Listing A.1 – Código Arduino