

UNIVERSIDADE FEDERAL DO PAMPA

LUCIANE MACIEL ARCE

**DESENVOLVIMENTO DE UM PROTÓTIPO DE CONTROLADOR DE IRRIGAÇÃO
DE BAIXO CUSTO BASEADO NO USO DO ARDUINO**

Alegrete

2025

LUCIANE MACIEL ARCE

**DESENVOLVIMENTO DE UM PROTÓTIPO DE CONTROLADOR DE IRRIGAÇÃO
DE BAIXO CUSTO BASEADO NO USO DO ARDUINO**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Agrícola da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharela em Engenharia Agrícola.

Orientador: Prof. Dra. Fátima Cibele Soares

**Alegrete
2025**

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais).

A451d Arce, Luciane

DESENVOLVIMENTO DE UM PROTÓTIPO DE CONTROLADOR DE
IRRIGAÇÃO DE BAIXO CUSTO BASEADO NO USO DO ARDUINO
/ Luciane Arce.

78 p.

Trabalho de Conclusão de Curso (Graduação) – Universidade
Federal do Pampa, ENGENHARIA AGRÍCOLA, 2025

Orientador: Fátima Cibele Soares.

1. Automação agrícola. 2. Sensor de umidade. 3. Hardware livre. 4.
Gestão hídrica. 5. Prototipagem eletrônica. I. Título.

LUCIANE MACIEL ARCE

DESENVOLVIMENTO DE UM PROTÓTIPO DE CONTROLADOR DE IRRIGAÇÃO DE BAIXO CUSTO BASEADO NO USO DO ARDUINO

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Agrícola da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharela em Engenharia Agrícola.

Trabalho de Conclusão de Curso defendido e aprovado em: 16, julho de 2025.

Banca examinadora:

Profª Drª. Fátima Cibele Soares

Orientadora

(UNIPAMPA)

Profª Drª. Lanes Beatriz Acosta Jaques

(UNIPAMPA)

Eng. Agríc Giulian Rubira Gauterio

(UNIPAMPA)



Assinado eletronicamente por **FATIMA CIBELE SOARES, PROFESSOR DO MAGISTERIO SUPERIOR**, em 18/07/2025, às 09:54, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **GIULIAN RUBIRA GAUTERIO, ENGENHEIRO-AREA**, em 18/07/2025, às 17:05, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **LANES BEATRIZ ACOSTA JAQUES, PROFESSOR DO MAGISTERIO SUPERIOR**, em 25/07/2025, às 08:43, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1787414** e o código CRC **03D1C88E**.

RESUMO

A irrigação desempenha um papel fundamental na eficiência das práticas agrícolas, favorecendo o crescimento correto das culturas e o aumento da produção. Em um cenário mundial, onde a água é um recurso cada vez mais escasso e a demanda por alimentos cresce, o uso de novas tecnologias e a automatização na irrigação se tornam imprescindíveis para melhorar a gestão da água. Desta forma os sistemas automatizados se destacam, por permitirem um controle preciso da quantidade de água aplicada em cada planta, ajustando a irrigação de acordo com as necessidades específicas de cada cultura e as condições ambientais, em tempo real. Estes sistemas não apenas reduzem o desperdício de água, mas também contribuem para o aumento da produtividade e a bem-estar das plantas. Neste contexto, este trabalho apresenta o desenvolvimento e a validação de um protótipo de um controlador de irrigação com baixo custo, baseado no Arduino UNO, para sistemas de irrigação por gotejamento. O controlador opera de forma automatizada, monitorando continuamente a umidade do solo e o nível de água disponível no reservatório. Com base nas condições ideais para três culturas predefinidas, que inicialmente foram; alface, tomate e pepino. Com limitantes de umidade de 70%, 70% e 50%, respectivamente. Os sensores coletam dados de umidade em tempo real, a lógica de controle aciona a válvula solenoide quando a umidade está abaixo do nível indicado como ideal, ao atingir, inicia-se uma pausa de trinta minutos gerenciada pelo módulo RTC (relógio de tempo real), otimizando a absorção de água pelo solo. O painel de controle está equipado com botões push down, sendo três para seleção de cultura e um destinado a ligar e desligar o sistema, mensagens informativas são exibidas no display, como nível de umidade, mensagens, alertas de erro, de falta de água e porcentagem de umidade em tempo real. A fase de implementação revelou desafios práticos, como instabilidades elétricas causada pela alta demanda de corrente dos relés, que foi solucionada com a implementação de um circuito de alimentação isolado e dedicado. O protótipo final demonstrou ser funcional, com todos os indicadores visuais e sonoros respondendo corretamente às simulações de falha. O controlador desenvolvido prova ser uma solução tecnológica e eficiente, com potencial para reduzir o consumo de água e automatizar o manejo da irrigação, promovendo uma agricultura mais sustentável e acessível, especialmente para pequenos produtores.

Palavras-chave: Automação agrícola. Sensor de umidade. Hardware livre. Gestão hídrica. Prototipagem eletrônica.

ABSTRACT

Irrigation plays a fundamental role in the efficiency of agricultural practices, favoring the correct growth of crops and increasing production. However, in a global scenario, where water is an increasingly scarce resource and the demand for food grows due to population growth, the use of new technologies and automation in irrigation become essential to improve water management more efficiently. In this way, automated systems stand out because they allow precise control of the amount of water applied to each plant, adjusting irrigation according to the specific needs of each crop and environmental conditions, in real time. These systems not only reduce water waste, but also contribute to increased productivity and plant well-being. In this context, this work presents the development and validation of a prototype of a low-cost irrigation controller, based on Arduino UNO, for drip irrigation systems. The controller operates in an automated manner, continuously monitoring soil moisture and the level of water available in the reservoir. Managing irrigation based on the ideal conditions of three predefined crops, initially lettuce, tomato and cucumber. With humidity limits of 70%, 70% and 50%, respectively. The sensors collect humidity data in real time, the control logic activates the solenoid valve when humidity is below the level indicated as ideal, when it reaches it, a thirty-minute pause is initiated managed by the RTC module, optimizing water absorption by the soil. The control panel is equipped with push-down buttons, three for crop selection and one for turning the system on and off, informative messages are displayed on the LCD display, such as humidity level, irrigation status, error alerts, lack of water and humidity percentage in real time. The implementation phase revealed practical challenges, such as electrical instabilities caused by the high current demand of the relays, which was solved by implementing an isolated and dedicated power circuit. The final prototype proved to be functional, with all visual and audible indicators responding correctly to failure simulations. The controller developed proves to be a technological and efficient solution, with the potential to reduce water consumption and automate irrigation management, promoting more sustainable and accessible agriculture, especially for small producers.

Keywords: Agricultural automation. Moisture sensor. Open hardware. Water management. Electronic prototyping.

LISTA DE FIGURAS

Figura 01 – Arduino UNO frontal e mini protoboard.....	26
Figura 02 – Sensor de solo e driver.....	27
Figura 03 – Válvula solenoid.....	28
Figura 04 – Modulo de relé.....	29
Figura 05 – RTC DS3231.....	30
Figura 06 – Display LCD.....	31
Figura 07 – Buzzer piezo elétrico.....	31
Figura 08 – Sensor de nível de água.....	32
Figura 09 – Leds.....	33
Figura 10 – Módulo conversor DC-DC buck step down.....	34
Figura 11 – Regulador de tensão 7809.....	34
Figura 12 - Fluxograma da lógica do sistema.....	37
Figura 13 - Inicio do fluxograma.....	38
Figura 14 - Fluxograma de reservatório baixo e erro.....	38
Figura 15 - Fluxograma do sistema de irrigação.....	39
Figura 16 - Diagrama de interligação do Arduino e seus componentes.....	41
Figura 17 - Entrada Jack P4 no Arduino e conector P4 da fonte.....	45
Figura 18 - Diagrama das ligações eletrônicas.....	46
Figura 19 - Ligações da lâmpada com módulo relé.....	47
Figura 20 - Programação do protótipo.....	49
Figura 21 - Script de diagnóstico.....	55
Figura 22 - Mensagem de reservatório baixo.....	57
Figura 23 - Mensagem de erro no sistema.....	57
Figura 24 - Sistema em funcionamento normal.....	58
Figura 25 - Irrigação em funcionamento.....	59
Figura 26 - Pausa pós irrigação.....	60
Figura 27 - Mensagem de boas vindas.....	60
Figura 28 - Seleção de cultura.....	61
Figura 29 - Mensagem de desligamento.....	61
Figura 30 - Visão frontal.....	62
Figura 31 - Gráfico de comparação de preços.....	66

LISTA DE TABELAS

Tabela 01 – Materiais necessários para o controlador de irrigação.....	25
Tabela 02 - Conexões de alimentação e distribuição de energia.....	42
Tabela 03 - Conexões de controle e sensores.....	42
Tabela 04 - Conexões de potência.....	44
Tabela 05 – Análise de custos.....	64

LISTA DE ABREVIATURAS

D2/D13 – Portas digitais do Arduino.

A0/A1 – Portas analógicas do Arduino.

+5V – 5 volts

R - Valor do resistor em ohms.

V - Tensão de alimentação.

V_{led} - Tensão de queda do LED (geralmente em torno de 1,8V para LEDs comuns).

I_{led} - Corrente máxima que o LED pode suportar (geralmente em torno de 20mA para LEDs comuns).

LISTA DE SIGLAS

UNIPAMPA – Universidade Federal do Pampa.
IoT – internet das coisas.
IA – inteligência artificial.
IDE – ambiente integrado de desenvolvimento.
RTC – Relógio de tempo real (Real-Time Clock).
LCD – Liquid Crystal display.
DC – Corrente direta.
SD – Cartão de memória (secure digital).
CS – Chip select.
GND – Terra (ground).
VCC – Alimentação positiva (voltage common collector).
SDA – Serial data.
SCL – Serial clock.
COM – Terminal comum do relé.
NO - Saída Contato Normalmente Aberto.
ETo – Evapotranspiração de referência.
PW - Peso da água evaporada.
PU - Peso úmido.
PS - Peso seco.
12C – Inter – integrated circuit.
UART – Universal asynchronous receiver – transmissor.
USB – Universal Serial Bus.

SUMÁRIO

1 INTRODUÇÃO	15
2 OBJETIVOS	17
2. 1 OBJETIVO GERAL	17
2. 1 OBJETIVOS ESPECÍFICOS	17
3 REVISÃO DE LITERATURA.....	18
3.1 Irrigação	18
3.1.1 Irrigação por gotejamento	20
3.1.2 Manejo de irrigação	21
3.1.3 Controladores no manejo de irrigação	22
3.2 Arduino	23
3.2.1 Plataforma Arduino	25
4. MATERIAIS E MÉTODOS	26
4.1. Materiais utilizados	26
4.2 Arduino UNO	27
4.3 Sensor de umidade de solo	28
4.4 Válvula solenoide	29
4.5 Módulo de relé	29
4.6 RTC DS3231	30

4.7 Display LCD	31
4.8 Buzzer piezo elétrico	32
4.9 Sensor de nível de água	33
4.10 Leds	33
4.11 Modulo conversor DC-DC buck step down	34
4.12 Regulador de tensão	35
5. METODOLOGIA	36
5.1 LÓGICA DO ARDUINO UNO	37
5.2 Funcionamento Simplificado da lógica do sistema	40
5.2.1 Início	40
5.2.2 Iniciando sistema	40
5.2.3 Alerta de Nível de Reservatório	40
5.2.4 Indicação de Erro	41
5.2.5 Monitoramento da Umidade do Solo	41
5.2.6 Pausa Pós-Irrigação	41
5.2.7 Interface com o Usuário	41
5.2.8 Quatro botões no painel	41
5.3 CONEXÕES	42
5.3.1 GUIA DE LIGAÇÕES ELÉTRICAS	43
6. RESULTADOS E DISCUSSÕES	46

6.1.1 CIRCUITO DE CONTROLE DO ARDUINO	46
6.1.2 Circuito de potência das bobinas dos relés	47
6.1.3 Circuito de potência dos atuadores	48
6.2 PROGRAMAÇÃO DO ARDUINO	48
6.3 AVALIAÇÕES DO SISTEMA	55
6.3.1 Avaliação e Calibração dos Sensores	56
6.3.2 Avaliação de Alertas e Indicadores	57
6.3.3 Avaliação do Painel de Controle e Interface do Usuário	62ava
6.4 MODIFICAÇÕES REALIZADAS DO PROJETO ORIGINAL	63
6.5 CAPACIDADE DE ESCALONAMENTO DO PROJETO E MUDANÇAS FUTURAS	64
6.6 ANÁLISE DE CUSTOS	65
7. CONCLUSÃO	69
REFERÊNCIAS	70

1 INTRODUÇÃO

Segundo Roux (1992) historicamente, a irrigação teve início a cerca de 6 mil anos atrás no antigo Egito e Mesopotâmia, com métodos rudimentares de transporte de água, como técnicas de sulcos e canais que, embora eficientes para a época, resultavam em grandes perdas de água. Ao longo do tempo, a irrigação passou por vários processos de desenvolvimento e adaptou-se aos diferentes ambientes, surgindo assim diferentes sistemas que trouxeram maior precisão na distribuição hídrica (REIS, 2015). Para Coelho, et al., (2005) a eficiência da irrigação não apenas assegura o crescimento saudável das plantas, mas também contribui para a sustentabilidade do sistema agrícola como um todo. Já segundo Paz et al., (2000) o uso eficiente da água na irrigação está diretamente relacionado com o sucesso da produção agrícola final.

Neste contexto, o desenvolvimento e a aplicação de tecnologias avançadas de irrigação surgem como uma solução viável para otimizar o uso de água, minimizando desperdícios e maximizando a produtividade. Afirmando o que demonstra Bonfim et al. (2023), destacando que a eficiência na irrigação está diretamente ligada à adoção de tecnologias que permitam o controle mais preciso e automatizado do fornecimento de água, reduzindo perdas e aumentando a produtividade. Dessa forma o uso de sensores para monitorar a umidade do solo, associado com os sistemas de irrigações automatizados, podem levar a uma considerável economia de água e aumento na produtividade das culturas (PEREIRA et al., 2018).

Existem várias formas de automatizar a irrigação, dentre eles o uso de sensores de umidade e sistemas de automação como a plataforma Arduino, que controlam todo o manejo da irrigação e aplicação de nutrientes, fornecendo dados precisos em tempo real, permitindo assim que a irrigação seja acionada somente quando necessário e na quantidade correta (SOUZA et al., 2019; SILVA et al., 2020).

O Arduino surgiu na Itália em 2005, onde seus criadores tinham como principal objetivo, um dispositivo barato, funcional e de fácil programação, mostrando-se desde então uma ferramenta muito utilizável em projetos (SILVA et al. 2014). Esses microcontroladores funcionam como pequenos computadores programáveis, ao qual podem configurar as suas entradas e saídas, dependendo do componente que for conectado as mesmas (SILVA, 2017).

Na irrigação, a automação pode ser aplicada tendo a finalidade de automatizar processos no campo, podendo reduzir o desperdício de água, promovendo um nível adequado de água para cada planta, atendendo assim a suas necessidades fisiológicas diárias (FINHOLDT, 2011).

Deste modo, o presente trabalho visa o desenvolvimento de um controlador de irrigação automatizado, baseado no microcontrolador Arduino. Pretendendo criar uma solução de baixo custo e fácil implementação, capaz de monitorar e ajustar automaticamente o sistema de irrigação conforme o tipo de cultivo. Este controlador permite ao usuário optar por alguns tipos e culturas, pré-definidas pelo sistema, ajustando assim a frequência de irrigação baseado na porcentagem ideal de umidade de solo para cada planta, tendo como resultado uma maior eficiência e racionalidade do uso da água. Assim, o projeto busca não apenas melhorar o gerenciamento de água na agricultura, mas também contribuir para a sustentabilidade e o desenvolvimento de tecnologias mais acessíveis no setor.

2 OBJETIVOS

2.1 Objetivo geral

Desenvolver um protótipo de um sistema de irrigação automatizado, de baixo custo e fácil manuseio. Baseado no microcontrolador Arduino e sensores, que seja capaz de monitorar e controlar a irrigação de forma eficiente e precisa, contribuindo para o uso sustentável da água na agricultura e com funcionalidades adaptadas a pequenas culturas em sistemas de irrigação por gotejamento.

2.1 Objetivos Específicos

- Projetar e montar um sistema físico de irrigação, que permita a medição da umidade do solo em tempo real e o acionamento automático da água conforme os níveis de umidade;
- Desenvolver um código de programação para o microcontrolador Arduino, que englobe funções diversas, permitindo ajuste de parâmetros conforme diferentes tipos de culturas;
- Propor melhorias e alternativas de automação, como ajustes para diferentes tipos de culturas, visando aumentar a adaptabilidade e eficiência do sistema;
- Integrar uma válvula solenoide para o controle automático do fluxo de água;

3 REVISÃO DE LITERATURA

3.1 Irrigação

A irrigação é uma prática muito antiga, essencialmente criada para facilitar a agricultura, com origem em civilizações antigas, como a do Egito antigo e Mesopotâmia, que desenvolveram técnicas para distribuição de água pelos campos, garantindo a produção em regiões com escassez de água (CALZAVARA, 1953). Irrigação trata-se do ato de levar a água de forma artificial, para áreas escolhidas, normalmente proveniente de lagos, reservatórios, rios e poços. Essa prática tem muita importância no agronegócio, devido a constante mudança no ciclo da chuva (DRUMOND, 2022). Com o passar do tempo a irrigação evoluiu, incorporando cada vez mais novas tecnologias e sistemas para otimização do uso da água que tem se tornado um recurso, escasso e valioso (VIEIRA; TELLES; 2001).

Nota-se, que a irrigação é de grande importância na agricultura, além de garantir a quantidade de água suficiente para o crescimento das culturas, permitindo ainda, o controle sobre o crescimento das plantas e, portanto, reduzindo os riscos associados a variações hídricas (LIMA, et al. 1999). Desta forma, a irrigação vem contribuindo significativamente para a estabilidade, continuidade e previsibilidade da produção agrícola (GEISENHOFF et al., 2015). Portanto, os sistemas de irrigação evoluíram ao longo dos anos e novos métodos surgiram, sendo os principais: irrigação superficial, aspersão e a localizada. Segundo Ribeiro et al. (2017), cada sistema possui características específicas, com vantagens e desvantagens, considerando isso o monitoramento dos cultivos e a sua avaliação é de grande importância para a eficiência da irrigação.

Dentre os métodos existentes, a irrigação superficial é a forma mais antiga e simples, onde a água é aplicada diretamente no solo para que por escoamento gravitacional, a lâmina siga de forma contínua e sem ocorrência de erosão, isto se deve graças a sistematização do terreno ou por uniformização da superfície (FRIZZONE, 2017). No método de irrigação superficial, existem diferentes classificações, sendo: faixas, sulcos e inundação. Segundo Costa, et al. (2006), a irrigação por faixas ocorre com a inundação total da área por condução de água, pelo tempo suficiente para a aplicação da quantidade necessária para a irrigação. Já a pôr sulcos é caracterizada por aplicar a água em canais chamados de sulcos,

infiltrando ao longo do perímetro molhado e movimentando-se pela área (SCALOPPI, 2021). A irrigação por inundação, se dá por meio de aplicação de água de forma contínua em tabuleiros, sendo a mais empregada usando-se submersão com água parada ou corrente (MEDEIROS, 1995). Os sistemas para uso da irrigação por superfície geralmente, são de menor custo por área, embora tenha baixo custo inicial, pode apresentar problemas de eficiência devido a perdas por percolação, evaporação e escoamento superficial (TESTEZLAF, 2017).

A aspersão, por sua vez, faz a distribuição da água sob pressão no formato de gotas, como uma chuva artificial sobre a cultura, sendo mais eficiente que a superficial (ANDRADE, 2001). Segundo BISCARO, (2009), existem dois principais tipos de irrigação por aspersão: a convencional e a mecanizada. Para a aspersão convencional utiliza-se de sistemas fixos ou moveis, onde os aspersores ficam distribuídos ao longo de tubulações, sendo seu uso comum em áreas de pequeno e médio porte (MAROUELLI, 2017). Este tipo de irrigação pode ser subdividido em: portáteis, onde os equipamentos são movidos manualmente, semipermanente, onde parte da tubulação é enterrada e autopropelido, onde o emissor é montado sob uma carretinha que se movimenta pela pressão da própria mangueira (MAROUELLI, 1998). A mecanizada por outro lado, é utilizada em grades áreas e envolve sistemas como pivô central e lateral móvel. Sendo o pivô central, composto por uma estrutura que gira em torno de um ponto fixo, enquanto o lateral móvel se desloca em linha reta (BISCAIRO, 2009). Ambos os tipos permitem maior automação e eficiência na irrigação, em contrapartida exigem maior investimento, manutenção e ainda assim também é suscetível a perdas por evaporação e deriva do vento.

A irrigação localizada, inclui métodos como o via gotejamento e a microaspersão, aos quais se destacam pela eficiência no uso da água, aplicando-a diretamente na zona radicular da planta, minimizando as perdas e permitindo melhor gestão da irrigação, mostrando-se ideais para culturas como das hortaliças que tem demanda alta em precisão de água (SILVA, 2005). Entretanto, esses sistemas exigem maior investimento inicial e manutenção mais criteriosa e com frequência (VENTURA et al., 2017).

Dentro dos sistemas localizados, a irrigação por gotejamento tem ganhado destaque, em especialmente na produção de hortaliças como a alface (HAMADA E, 1993). Este método se mostra altamente eficaz devido à sua capacidade de fornecer água e nutrientes diretamente às raízes, promovendo um crescimento mais uniforme

e reduzindo a incidência de doenças foliares, que podem ser agravadas por umidade excessiva na parte aérea das plantas (GOMES; SOUZA, 2002). Em outras palavras, a escolha do sistema de irrigação deve considerar os aspectos específicos de cada cultura, as condições do solo, o clima e os recursos disponíveis.

3.1.1 Irrigação por gotejamento

A técnica de irrigação por gotejamento surgiu em Israel, principalmente devido à escassez de recursos hídricos (MANFRINATO, 1985). Este sistema tem por característica a aplicação da água diretamente na zona radicular da planta, com uso de emissores de água do tipo gotejadores, que são conectadas as linhas laterais, capazes de liberar a pressão disponível e liberar vazões de água de forma lenta e contínua (BERNARDO, et al. 2006), minimizando as perdas por evaporação, escoamento superficial e percolação. Segundo COELHO et al., (2009) por operar com baixa vazão e pressão, resulta em um bom volume de solo molhado, conhecido como bulbo molhado, o que mantém o solo próximo à capacidade de campo, utilizando baixa pressão de serviço e menor demanda de energia.

Este sistema já ocupa uma parte significativa das áreas irrigadas no mundo e segue em crescimento, porém somente para áreas específicas e para culturas economicamente rentáveis (SOUZA; MATSURA, 2004). Esta limitação econômica se dá por seu alto custo inicial em equipamentos, instalação e a manutenção, podendo não ser rentável se não forem aplicadas técnicas de manejo adequadas, que visem a eficiência da água e o aumento da produção final (COELHO, et al. 1999).

Em comparação a outros métodos, a irrigação por gotejamento se mostra superior, visando o aumento qualidade, quantidade e conservação dos recursos hídricos. Segundo GOMES, (1999) em um sistema por gotejamento, quase não se tem perda de água no trajeto do ponto de abastecimento inicial até a saída dos gotejadores, se bem manejado. Para WU, et al. (1983) é possível alcançar elevados níveis de eficiência de água, beirando os 90%, assumindo-se que a variação entre os gotejadores não passe de 20%. A uniformidade na distribuição de água é outro benefício importante, promovendo um crescimento homogêneo das plantas. Segundo MANTOVANI et al. (2007), as vantagens deste sistema incluem ainda: controle rigoroso da lâmina de água aplicada, menos interferência do vento, maior

produtividade em culturas sensíveis, fácil adaptabilidade a diferentes tipos de solos e relevos e, também a economia de mão de obra.

No entanto, também existem algumas desvantagens neste sistema, como: custo de instalação inicial elevado e manutenção constante. De acordo com Keller, et al. (1975), são necessárias avaliações periódicas do sistema, devido ao entupimento dos emissores, geralmente causados por material orgânico, deposição química e partículas minerais, o que geram perdas de produção.

Em síntese, esta técnica de irrigação é altamente eficiente e precisa, especialmente adequada a culturas que demandam a gestão rigorosa dos recursos hídricos. Apesar de seus custos e da necessidade de manutenção, seus benefícios se sobrepõem em termos de economia de água, melhora da produção, do produto e grande capacidade de automação, tornando-a assim uma opção atraente para a agricultura moderna.

3.1.2 Manejo de irrigação

Manejo de irrigação, refere-se ao conjunto de práticas e técnicas utilizadas para controlar a aplicação de água nas plantas, sendo o mais sustentável e eficiente possível. Seu principal objetivo é fornecer a quantidade adequada de água e no momento propício, garantindo assim que a cultura receba os recursos hídricos necessários para o seu desenvolvimento, com o mínimo de desperdício (SOUSA, et al. 2011).

A importância do manejo de irrigação se destaca, pela sua capacidade em otimizar o uso da água, reduzindo o custo de produção e melhorando a produtividade da cultura (RIBEIRO et al., 2017). Além disso, o manejo adequado previne problemas relacionados ao excesso de irrigação, como a lixiviação de nutrientes e o desenvolvimento de doenças radiculares (CUNHA, 2019).

Existem diferentes métodos de manejo de irrigação e sua escolha deve levar em consideração sempre as situações específicas do tipo de cultura escolhida, solo, clima e sistema de irrigação a ser empregado, já para determinar a quantidade de água a ser aplicada, deve-se analisar indicadores da planta, como: fatores atmosféricos e umidade de solo (ANDRADE, 2013). Um dos mais utilizados métodos é o baseado em monitoramento da umidade do solo, onde sensores ou tensiômetros são instalados para medir a quantidade de água no solo, permitindo a percepção do

momento exato de irrigar, podendo ser verificados de maneira indireta ou direta (CRUZ, 2019). Outro manejo baseado no clima, que se fundamenta em dados meteorológicos para calcular a demanda hídrica, por exemplo a evapotranspiração de referência (ET_o) e o tanque classe A (PINHEIRO, 2019).

Cada um dos métodos de manejo, possui suas particularidades que devem ser consideradas para estabelecer a escolha, atentando sempre as características específicas de cada situação agrícola, visando à maximização da eficiência do uso da água e à melhoria da produtividade.

3.1.3 Controladores no manejo de irrigação

Para ALMEIDA (2012), o correto manejo da irrigação tem como objetivo a alta eficiência do uso da água e energia, através da aplicação da quantidade de água adequada a planta e no momento certo, monitorando-a em todo seu ciclo vegetativo. Porém esse monitoramento demanda tempo e esforço, o que torna difícil a decisão de quando irrigar pelos produtores. A automação com a utilização de sensores e controladores tem facilitado essa tomada de decisão o que acarreta em melhorias na eficiência do uso da água e redução dos custos de mão-de-obra (EVANS, 2013). A necessidade do uso racional dos recursos hídricos tem impulsionado também a adesão as novas tecnologias de automação no manejo de irrigação. Assim o uso de controladores na irrigação tem desempenhado um papel fundamental, tornando-se uma ferramenta importante, para garantia da melhora na produtividade agrícola (ARAÚJO JÚNIOR, R. 2017). Diversos estudos exploram o uso de controladores na irrigação, para desenvolver sistemas automatizados que respondem a necessidades específicas das culturas, utilizando sensores e algoritmos para ajustar a irrigação com base em parâmetros como umidade de solo e condições climáticas.

Como o estudo de PEREIRA et al. (2020) que desenvolveu um sistema de automação utilizando o Arduino e um inversor de frequência para manter a pressão de saída constante em sistemas de irrigação por válvulas, conseguindo alcançar uma eficiência energética significativa, com redução de até 42,9% na potência necessária para o funcionamento do sistema, demonstrando assim o potencial da automação na economia de energia e água do sistema. Já no trabalho de ARAÚJO JÚNIOR (2017), que elaborou um controlador automático com base na aplicação de

água em turnos de rega e lâminas variáveis, com comunicação sem fio. O sistema demonstrou uma aplicação mais precisa da água, se ajustando as necessidades específicas da cultura e às condições de solo, indicando melhoria da eficiência de utilização na gestão da irrigação.

A incorporação de tecnologias como wifi, IoT e IA podem ampliar as capacidades dos controladores. Segundo MORAES et al. (2014), a automatização pode ser uma grande ferramenta para controle de sistemas de irrigação, visando à economia de energia elétrica e da água, principalmente quando há necessidade de variações de vazão.

Contudo apesar dos benefícios, segundo ANDRADE (2025) ainda existem alguns desafios como custo inicial de implantação e necessidade de capacitação técnica para operação de algumas tecnologias. Deste modo pesquisas futuras devem focar não somente na tecnologia empregada, mas também no desenvolvimento de controladores mais acessíveis. A adoção de tecnologias, como o Arduino, oferece soluções acessíveis e adaptáveis às diversas realidades do campo, contribuindo para a sustentabilidade e eficiência da agricultura.

3.2 Arduino

O Arduino é uma plataforma de prototipagem eletrônica de hardware e software open-source, ou seja, de código aberto, amplamente utilizada em projetos de IoT, robótica e automação. Criada por Massimo Banzi em 2005 e uma equipe de pesquisadores, sendo David Cuatlielles, Tom Igoe, Gianluca Martino e David Mellis, do Instituto de design de interação de Ivrea, na Itália (EVANS, et al. 2013). Segundo Banzi (2015), o Arduino surgiu com o objetivo de ser uma ferramenta acessível e de fácil utilização, a qual permitisse o desenvolvimento de projetos de automação e de controle sem a necessidade de profundos conhecimentos em eletrônica ou programação. O grande diferencial do Arduino consiste na sua combinação de baixo custo, simplicidade de uso e grande versatilidade, transformando-a em uma plataforma globalmente difundida.

O funcionamento do Arduino baseia-se em um microcontrolador programável, geralmente da família Atmel AVR, com entradas e saídas, digitais e analógicas, que podem ser instruídas a ler sinais de entrada, como sensores e acionar saídas, como LEDs, motores ou válvulas, de acordo com o comportamento previamente

programado, por exemplo, em um sistema de irrigação automática, onde o sensor de umidade de solo pode acionar a bomba hidráulica quando necessário (MARGOLIS, 2011). Dispõe ainda de uma memória flash, pinos de alimentação, cristal oscilador, reguladores de tensão, conexão USB para alimentação e carregamento do código (EVANS, 2013). A programação de linguagem de desenvolvimento é baseada em C/C++ e pode ser elaborada por meio de um IDE próprio, intuitivo, gratuito e que conta ainda com uma vasta biblioteca com dados de vários usuários e tutoriais (CHAVIER, 2013).

As finalidades do Arduino abrangem inúmeras áreas, devido a flexibilidade e baixo custo. No setor agrícola o Arduino, tem sido utilizado principalmente na automação e monitoramento em tempo real, de dados como umidade do solo e temperatura, estimulando o crescimento de práticas mais sustentáveis e eficazes. Segundo PEREIRA, et al. (2020), pesquisas demonstram que o uso do Arduino em sistemas de irrigação automatizados, pode diminuir significativamente o desperdício de água e energia, proporcionando maior eficiência e produtividade. Para Gomes, (2019) o Arduino pode auxiliar a promover o controle automatizado de estufas, em conjunto com sensores de solo, temperatura, umidade, estações meteorológicas e outros acionadores, regulando a ventilação, irrigação e iluminação do local. Estas aplicações do microcontrolador contribuem para a redução de custos operacionais e o aumento da produção, tornando então o Arduino uma ferramenta valiosa e poderosa para a agricultura de precisão (DE SIQUEIRA; FERREIRA, 2023).

Com o avanço das tecnologias embarcadas e a expansão da conectividade, espera-se que futuramente o Arduino continue em um crescimento contínuo com melhorias e integração na indústria 4.0. Deste modo novas pesquisas tornam-se essenciais para explorar todo o potencial do Arduino, principalmente na integração do IoT e IA almejando sempre o aprimoramento da agricultura de precisão (SILVA et al., 2020). A fácil acessibilidade da plataforma a torna ideal para projetos de extensão e inovação no agronegócio, reforçando a importância de investimentos em estudos que ampliem suas aplicações no setor agrícola (SANTOS et al. 2017). Porém a maior importância do Arduino reside na democratização da tecnologia, permitindo que amadores e profissionais desenvolvam soluções inovadoras, sua natureza open-source e a comunidade ativa de desenvolvedores que garantem constantes atualizações e diversificações de suas aplicações, consolidando o Arduino como uma plataforma de grande relevância tecnológica e social.

3.2.1 Plataforma Arduino

O Arduino possui uma plataforma própria de programação, que tem se consolidado como uma das principais ferramentas para a prototipagem eletrônica e desenvolvimento de sistemas, tendo destaque principalmente por sua versatilidade e acessibilidade. Sendo lançada em 2005 como parte do projeto do Arduino, desenvolvida inicialmente para fins educacionais, tornou-se largamente adotada em diversas pesquisas científicas e projetos (BANZI, 2015). Grande parte do sucesso do Arduino deve-se ao seu sistema ser baseado em hardware aberto e um IDE que utiliza linguagem simples e conhecida para a programação dos microcontroladores (MUSCI et al. 2023).

A plataforma permite a escrita, compilação e envio de códigos diretamente para o microcontrolador, por uma interface gráfica simplificada, funcional e acessível. O layout da IDE é composta por área de edição, console de mensagens, botões de verificação, upload, novo, abrir, monitor serial e salvar, possibilitando a maior interação do usuário ao dispositivo conectado (VIANA, 2020).

A linguagem de programação é uma derivação do C/C++, com adaptações feitas especificamente para simplificar a interação com pinos no microcontrolador e demais componentes do sistema (MCROBERTS, 2011). Basicamente a estrutura de programação é composta pelas funções “setup()”, que fica responsável pela configuração inicial dos componentes, e “loop()”, que se repete continuamente os comandos do sistema. Existem também a possibilidade de acesso a bibliotecas que expandem as funcionalidades básicas da linguagem, tornando possível o controle de diversos tipos de módulos (DA PORCIÚNCULA et al. 2018).

A principal importância da plataforma Arduino de programação, está em sua capacidade de incorporar, baixo custo, facilidade de uso e potencial de inovação, promovendo o desenvolvimento de soluções acessíveis que atendem a todas as demandas da indústria tecnológica.

4. MATERIAIS E MÉTODOS

Este tópico mostra de forma detalhada os componentes eletrônicos utilizados e a metodologia adotada para o desenvolvimento do projeto.

4.1. Materiais utilizados

A seleção de materiais foi principalmente focada em baixo custo, acessibilidade no mercado e funcionabilidade, de modo a cumprir o objetivo de desenvolver uma solução acessível. A lista completa de componentes e as quantidades que foram necessárias está detalhada na Tabela 01 abaixo.

Tabela 01 – Materiais necessários para o controlador de irrigação.

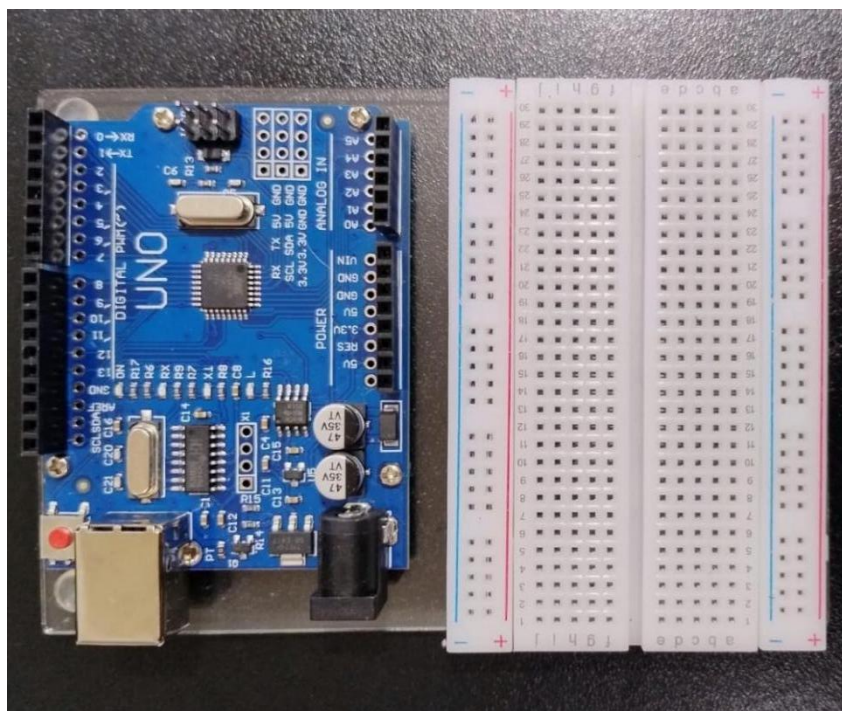
Componente	Quantidade
Arduino Uno	1
Sensor de umidade de solo	1
Sensor de nível de água	1
Módulo de Relé (2 canais)	1
Display LCD 16x2 12C	1
RTC DS3231	1
Válvula solenoide 12V	1
Resistores 220Ω	6
Driver LM-393	2
Botões (push-buttons)	4 unidades
Modulo conversor DC - DC buck step donw	1
Leds de 5mm	5
Fios	50
Quadro elétrico	1
Buzzer piezo elétrico	1
Lâmpada 12V (LED)	1
Fonte de alimentação 12V, 2A	1
Regulador de tensão (LM7805)	1
Mini-protoboard	1

4.2 Arduino UNO

O componente central do trabalho é a placa, modelo utilizado é a Arduino UNO, um microcontrolador da família ATmega329P, que atua como a unidade central de processamento. Sendo responsável por executar o algoritmo de controle, ler os dados dos sensores e comandar os atuadores. Executando a lógica de irrigação programada e respondendo as condições de solo e parâmetros que poderão ser definidos pelo próprio usuário.

Foi utilizada junto ao arduino, uma mini-protoboard também conhecida como matriz de contatos, para facilitar a organização e garantir a confiabilidade das múltiplas conexões elétricas do protótipo. Este componente é comumente utilizada em montagens temporárias, foi empregada na montagem final como um barramento de distribuição de energia, centralizando as linhas de alimentação 5V e GND. A placa Arduino UNO possui poucos pinos de alimentação, que seriam insuficientes para a quantidade de componentes do sistema. Assim através da protoboard, uma única saída de 5V do Arduino consegue distribuir de maneira organizada para todos os periféricos. Logo abaixo podemos ver uma imagem do Arduino UNO e a mini-protoboard utilizada em uma visão frontal.

Figura 01 – Arduino UNO frontal e mini-protoboard.



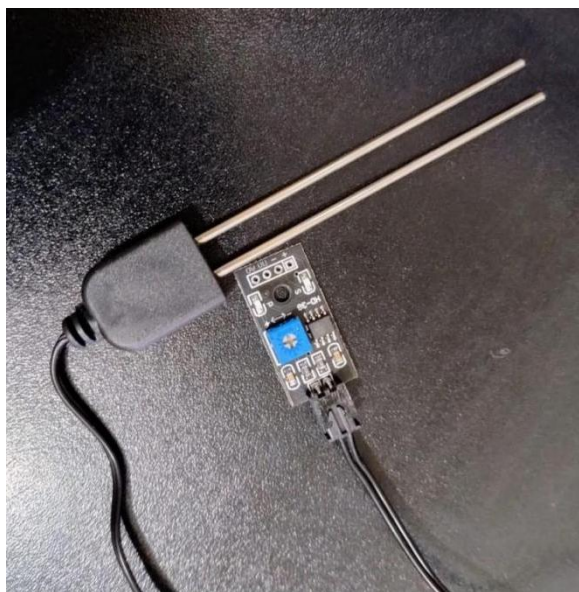
Fonte: Autor.

4.3 Sensor de umidade de solo

A finalidade do sensor de umidade de solo é medir a quantidade de água presente no solo. Sendo um dispositivo amplamente utilizado na agricultura de precisão por garantir uma irrigação adequada e eficiente. Este tipo de sensor opera por meio de sondas metálicas que, ao serem inseridas no solo, medem a resistência elétrica. A umidade do solo altera essa resistência, e esse sinal é recebido pelo microcontrolador Arduino UNO. Quanto maior a umidade presente no solo, menor a resistência entre as sondas e maior o sinal de saída.

Para o monitoramento destas variáveis de umidade de solo, foi utilizado um sensor de solo, modelo HD-38, por ter hastes anticorrosivas, o que aumenta a vida útil do sensor, tornando o projeto mais durável. Ele tem por característica também, tensão de operação de 3,3 a 12V, saídas digitais, saídas analógicas e dimensões de 36 x 15 x 7 mm. Desta forma no projeto o sensor de umidade do solo, será o responsável por fornecer dados sobre o nível de umidade do solo ao Arduino e baseado nestes dados o microcontrolador decidirá se é necessário acionar a irrigação. Na figura 02 logo abaixo é possível ver uma imagem do sensor que foi utilizado.

Figura 02 – Sensor de solo e driver.



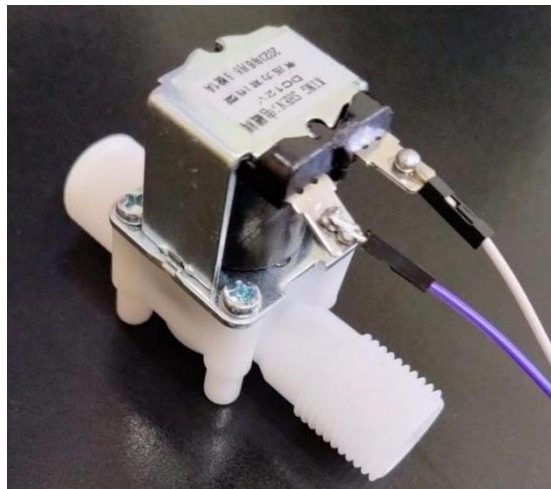
Fonte: Autor.

4.4 Válvula solenoide

A válvula solenoide foi desenvolvida para controlar o fluxo de fluidos de uma forma automática, podendo ser utilizada em sistemas de encanamento, automação industrial e dispositivos de irrigação. Esta válvula funciona com um eletroímã que, ao ser energizado, abre ou fecha o fluxo de água. Quando a corrente elétrica passa pela bobina, é formado um campo magnético que faz com que o pistão se movimente, regulando assim o fluxo.

Para o projeto será utilizada uma válvula solenoide 12V com tubulação $\frac{3}{4}$, ela terá função de controlar o fluxo de água. Com ativação pelo módulo relé sob o comando do Arduino, permitindo a irrigação do solo quando o nível de umidade estiver abaixo do valor pré-programado pelo sistema, na figura 03 é possível ver uma imagem da uma válvula solenoide qual foi utilizada no trabalho.

Figura 03 – Válvula solenoide.



Fonte: Autor.

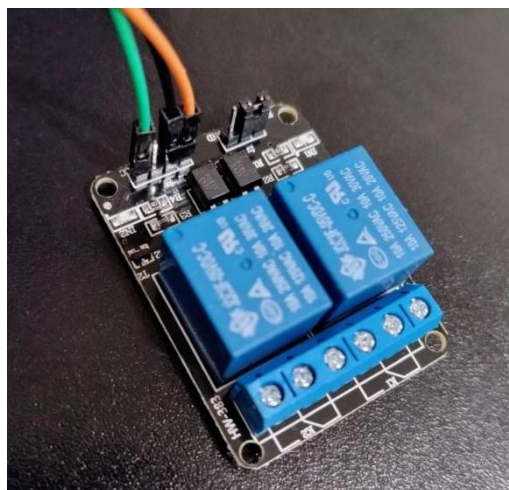
4.5 Módulo de relé

O módulo relé é um dispositivo eletromecânico que funciona como um interruptor controlado de forma eletrônica, onde uma pequena corrente acionada pelo microcontrolador ativa um eletroímã que abre e fecha o circuito de alta tensão. O módulo é composto por um transistor que amplifica o sinal do Arduino, a bobina cria o campo magnético e um contato comutador que permite o controle de dispositivos conectados à rede elétrica.

No projeto o módulo relé 5V, com um canal será utilizado, para controlar a válvula solenoide e a lâmpada de emergência. Ele atua como uma ponte segura entre o circuito de controle de baixa tensão do Arduino e os atuadores de alta potência, que são a válvula solenoide e a lâmpada de alerta. Baseando o seu funcionamento no princípio de isolamento galvânico, onde um sinal lógico de 5V, enviado pela porta digital do Arduino, ativa um optoacoplador interno. Este por sua vez, permite que um circuito de alimentação de 5V dedicado energize a bobina do relé. A energização da bobina gera um campo magnético que aciona um interruptor mecânico interno, conectando o terminal COM ao terminal NO.

No contexto deste trabalho o terminal COM de cada canal do relé é conectado diretamente a fonte principal de 12V. Quando o Arduino decide iniciar a irrigação ou o sistema de alerta, o relé corresponde acionando a energia de 12V que flui do COM para o NO, alimentando a válvula solenoide ou a lâmpada. Assim o modulo relé cumpre o papel de permitir que o microcontrolador mesmo com sua corrente limitada, controle as cargas de potência significativamente maiores, garantindo a integridade e a segurança do sistema. Abaixo pode-se ver na figura 04 o modulo relé que foi utilizado no prototipo, sendo do tipo modulo relé de dois canais.

Figura 04 – Modulo de relé.



Fonte: Autor.

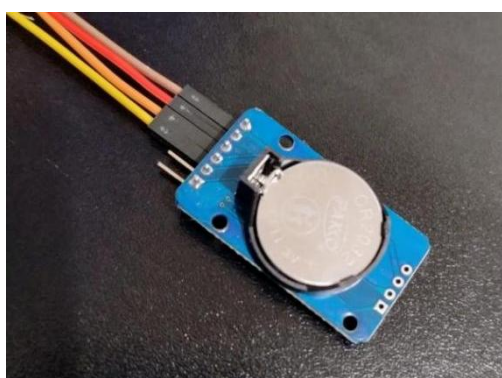
4.6 RTC DS3231

Os relógios de tempo real (RTC) foram desenvolvidos para manter uma contagem precisa do tempo, mesmo quando o sistema está desligado devido a uma

bateria acoplada ao componente. Muito utilizados em dispositivos como computadores, sistemas de alarme e controles industriais. Este controlador usa um oscilador de cristal que mantém o tempo corrente, no Arduino o RTC se comunica através da interface 12C, atualizando o microcontrolador com a hora e datas atuais.

No sistema de irrigação automatizado, o RTC permite a programação monitorar o tempo de pausa após cada ciclo de irrigação programado em trinta minutos de pausa após cada irrigação, garantindo que o solo tenha tempo para absorver a água antes de uma nova ativação do sistema e também tem a função de iniciar o timer de desligamento do sistema em caso de erros, para proteger o sistema de sobrecarregamentos em caso de erro e o desperdício de energia, para que o sistema não fique muito tempo ligado sem estar funcionando corretamente. Na figura 05 mostra uma imagem do RTC DS3231, frente.

Figura 05 – RTC DS3231.

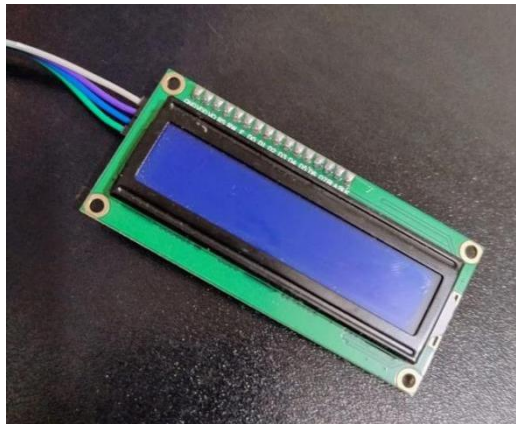


Fonte: Autor.

4.7 Display LCD

O LCD utiliza cristais líquidos que alteram sua opacidade quando uma corrente elétrica é aplicada. No display utilizado no projeto, a comunicação será feita através de um barramento 12C, que permite a transferência de dados de maneira simplificada e eficiente. No projeto a função do display será exibir as informações sobre o estado do sistema, bem como os níveis de umidade do solo, o tipo de cultivo selecionado e o status da irrigação. O modelo será display LCD 16 x 2 12C, tendo em vista facilitar a interação do usuário com o sistema e fornecer uma visão clara das condições monitoradas. Para melhor visualização é possível ver logo abaixo na figura 06 do display LCD utilizado.

Figura 06 – Display LCD



Fonte: Autor.

4.8 Buzzer piezo elétrico

Buzzer ou transdutores piezoelétricos, foram desenvolvidos para fornecer alertas sonoros em dispositivos eletrônicos. Ele emite um som quando uma corrente elétrica é aplicada. O modelo piezo eletrônico que será utilizado, funciona através de deformação de um cristal piezoelétrico que gera vibrações audíveis ao ser energizado. E terá a atuação em situações de alerta, quando o nível do reservatório estiver baixo ou ocorrendo algum defeito no sistema. Fornecendo uma indicação sonora para que o usuário possa notar com facilidade o quando ocorre um problema, complementando os alertas visuais. Na figura 07 podemos ver o buzzer piezo elétrico utilizado no projeto.

Figura 07 – Buzzer piezo elétrico.

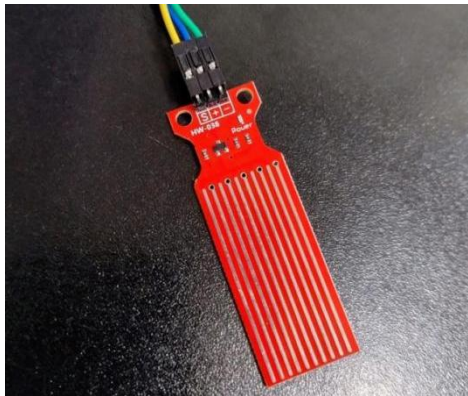


Fonte: Autor.

4.9 Sensor de nível de água

Este tipo de sensor surgiu pela necessidade de monitorar o volume de líquido em reservatórios e tanques industriais. Funcionando geralmente com base em flutuação ou na condução elétrica entre eletrodos. No projeto o sensor modelo EASY PLUG RJ11, atua fazendo a verificação do nível da água no reservatório, se estiver abaixo do pré-definido pelo sistema, um sinal de alerta é enviado do sensor ao Arduino, fazendo assim com que ele acione, o led vermelho, lâmpada e o buzzer, além de impedir a ativação da irrigação, com intuito de evitar danos ao sistema, por operar vazio. Em seguida na figura 08, observa-se uma imagem do sensor de nível de água.

Figura 08 – Sensor de nível de água.



Fonte: Autor.

4.10 Leds

Os LEDs são dispositivos de polaridade, ou seja, com lado positivo (ânodo) e lado negativo (cátodo). Para o funcionamento do LED é necessário que a corrente elétrica passe no sentido correto, do positivo para o negativo. Pode-se fazer isto conectando o ânodo a porta digital do Arduino e o cátodo a um resistor limitador de corrente. Para escolher o resistor limitador de corrente adequado foi utilizada a fórmula abaixo (GEA, 2023).

$$R = \frac{(V - V_{led})}{I_{led}}$$

Onde:

- R é o valor do resistor em ohms.
- V é a tensão de alimentação.
- V_{led} é a tensão de queda do LED (geralmente em torno de 1,8V para LEDs comuns).
- I_{led} é a corrente máxima que o LED pode suportar (geralmente em torno de 20mA para LEDs comuns).

Chegando a um resistor de 220Ω , no projeto os LEDs fornecem indicações visuais sobre os status do sistema, sendo cada cor uma condição específica:

- Amarelo: erro no sistema.
- Branco: sistema de irrigação funcionando.
- Verde: funcionamento normal do sistema (irrigação desligada).
- Azul: pausa pós-irrigação;
- Vermelho: baixo nível de água no reservatório.

A diferenciação por cores foi escolhida com a intenção, de colaborar com o fácil monitoramento do sistema pelo usuário. Segue uma imagem dos leds utilizados.

Figura 09 – Leds.



Fonte: Autor.

4.11 Módulo conversor DC-DC buck step down

Este componente é um conversor de chaveamento de alta eficiência, projetado para reduzir uma tensão de entrada maior para uma tensão de saída menor e estável, com alta capacidade de corrente. No projeto, o módulo é alimentado com 12V da fonte principal e foi ajustado para fornecer uma saída regulada de 5V. A saída de 5V é usada exclusivamente para alimentar as bobinas do módulo

relé, através do pino RY-VCC, ao qual o jumper de acoplamento foi removido, criando assim um circuito de alimentação para o relé totalmente isolado da alimentação lógica do Arduino de 5V. Na figura 10 mostra o modulo conversor utilizado na montagem do prototipo.

Figura 10 - Módulo Conversor DC-DC buck step down.

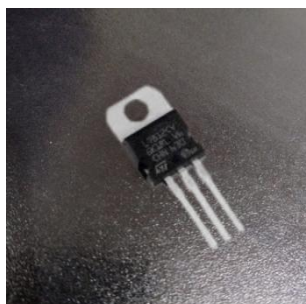


Fonte: Autor.

4.12 Regulador de tensão

Este componente possui três terminais e foi projetado para receber uma tensão de entrada variável e fornecer uma tensão de saída fixa e constante. No circuito de alerta, o regulador 7809 foi posicionado em série, após a saída NO na entrada da lâmpada para o relé. Deste modo quando o Arduino aciona o relé a tensão de 12V passa primeiro pelo regulador, que reduz para 9V antes de alimentar a lâmpada. Na figura 11 pode-se ver a imagem do regulador de tensão 7809 que foi utilizado no projeto.

Figura 11 - Regulador de tensão 7809.



Fonte: Autor.

5. METODOLOGIA

A metodologia de desenvolvimento do protótipo do controlador de irrigação foi construída sob a arquitetura de malha fechada, onde o estado do ambiente, especificamente a umidade do solo é continuamente monitorada para influenciar as ações de controle sobre o sistema de irrigação. O Arduino UNO atua como a unidade central de processamento, o núcleo que interliga todos os componentes e executa a lógica de automação.

A lógica operacional do sistema prioriza a segurança e a eficiência, ao ser ativada pelo usuário através do botão liga/desliga, localizado acima do painel em preto, o controlador não inicia imediatamente suas operações, ele primeiro executa uma rotina de verificação crítica, testando a comunicação com periféricos essenciais com o módulo RTC e lendo o sensor de nível de água. Caso o sistema detecte uma falha de hardware ou nível de água baixo no reservatório, o sistema impede a continuação da sequência de inicialização e entra imediatamente no modo de alerta correspondente, acionando os indicadores visuais e sonoros para notificar o usuário da necessidade de intervenção. Em caso de “erro” o sistema entra em um timer de trinta minutos regressivo para o desligamento, este sistema foi implementado para evitar queima de componentes e também para promover a eficiência energética, desligando-se automaticamente e evitando o equipamento ficar por longos períodos ligado sem o correto funcionamento, o LED amarelo ascende junto a mensagem de erro para uma melhor visualização do usuário. Em caso de reservatório baixo o LED vermelho, a lâmpada e o buzzer são ativados, este é o sistema mais crítico e tem prioridade hierárquica dentro do sistema, pois o sistema não pode operar sem água.

Apenas se todas as verificações de segurança forem bem-sucedidas, o sistema prossegue para tela sequencial de boas-vindas e então passa para o modo de espera, sinalizado pelo acendimento do LED verde, neste estado o sistema fica na espera da interação do usuário para a parametrização da irrigação, que é realizada através de três botões dedicados à seleção de cultura. O projeto conta inicialmente com três culturas; alface (70%), tomate (70%) e pepino (50%), cujos valores representam os limites de umidade de solo ideais para a cultura definidos com base na literatura. Segundo FILGUEIRA (2008) a alface se desenvolve bem sob umidade de 60 e 80%, acima pode haver acúmulo de água o que favorece a proliferação de fungos e abaixo disto favorece a perda de água e murchamento das

folhas. Deste modo seguindo a literatura acima a umidade da alface será definida como a média entre as umidades, adotando então o valor de 70%. Da mesma forma para o pepino, segundo BALBINO. et. al (1991) para solos de textura media o nível de umidade no solo deve ser entre 40 e 60% para que não ocorra estresse hídrico na cultura. Assim para a cultura do pepino a média para que ocorra o acionamento será definido em 50%. Já a cultura do tomate demonstra resultados melhores em umidades em torno de 70%, tendo uma demanda de água de 600 mm e com exigência de irrigações frequentes. Portanto o valor para o projeto será pré-definido em 70% de umidade do solo. Assim que é selecionada uma cultura os parâmetros de “umidade alvo”, que guia todas as tomadas de decisões subsequentes da irrigação, são definidos. Uma vez estipulados os parâmetros o controlador entra em seu ciclo de operação principal, monitorando continuamente a umidade do solo. A leitura analógica do sensor é convertida pelo Arduino em um valor percentual, que é comparado em tempo real com a umidade alvo. Se a umidade medida for inferior ao limite ideal, o Arduino envia um sinal ao módulo relé, que atua como um interruptor eletronicamente isolado, acionando a válvula solenoide de 12V, iniciando o fluxo de água, durante este processo o LED laranja é ativado e o display LCD informa que a irrigação está em andamento, exibindo também a porcentagem de umidade em tempo real. A irrigação prossegue de forma contínua até que o nível de umidade do solo atinja ou ultrapasse a meta definida.

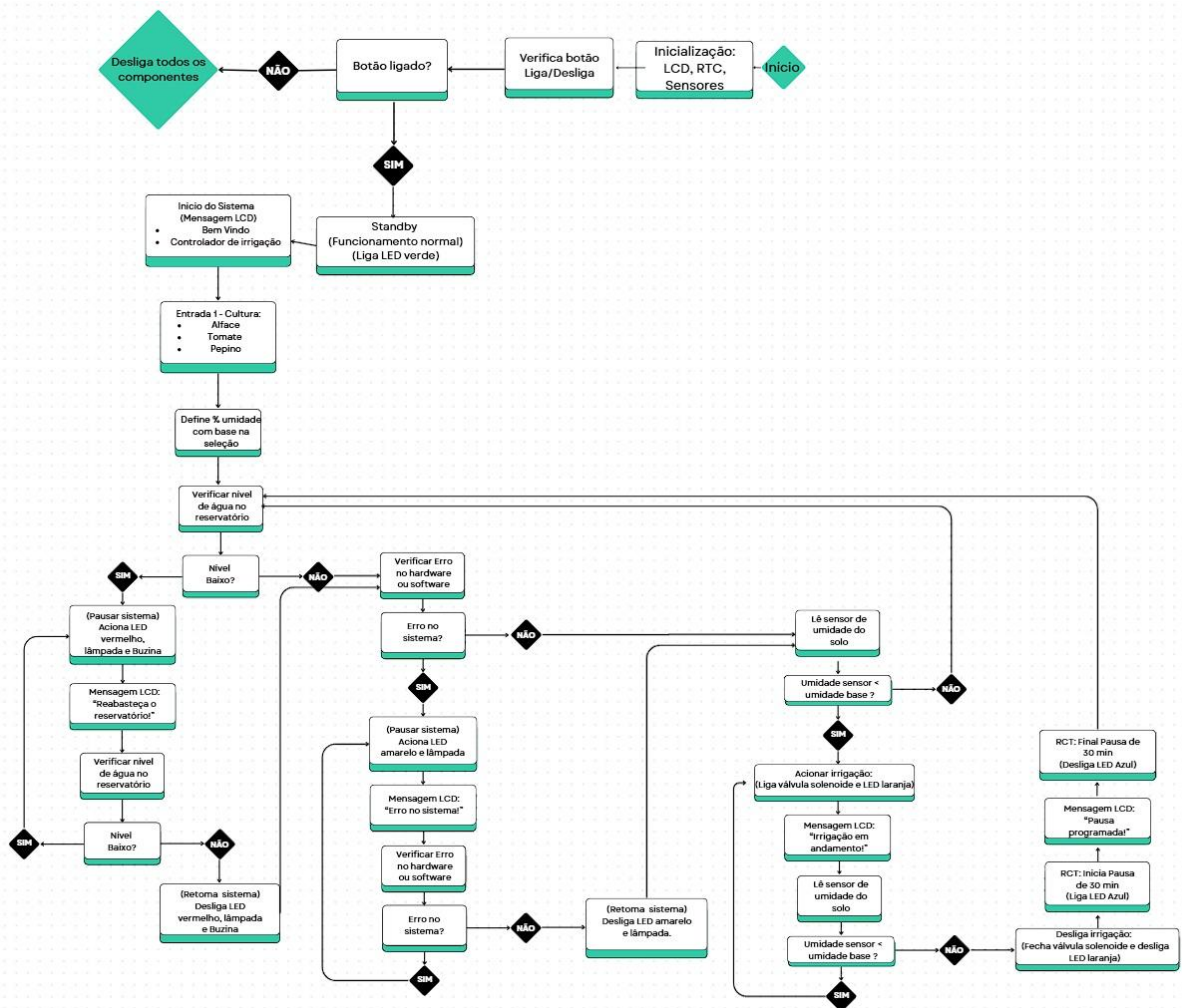
Após atingir a umidade ideal novamente o Arduino para o sinal para o relé, fechando assim a válvula e cessando a irrigação. Logo após o término da rega, o controlador entra em um período de pausa de trinta minutos, visando evitar o acionamento excessivo do sistema e permitindo a correta absorção da água pelo solo. Este intervalo é gerenciado pelo módulo RTC e o estado de pausa é sinalizado pelo acionamento do LED azul e uma mensagem com a contagem regressiva do tempo para sair do estado de pausa. Após o término da pausa, o sistema retorna ao ciclo de monitoramento ativo, pronto para iniciar o ciclo novamente se necessário, garantindo assim um manejo hídrico eficiente e autônomo.

5.1 Lógica do Arduino UNO

Exemplificando de forma simplificada como será a lógica do Arduino UNO e a base de suas tomadas de decisões. Na Figura 12 demonstra em formato de

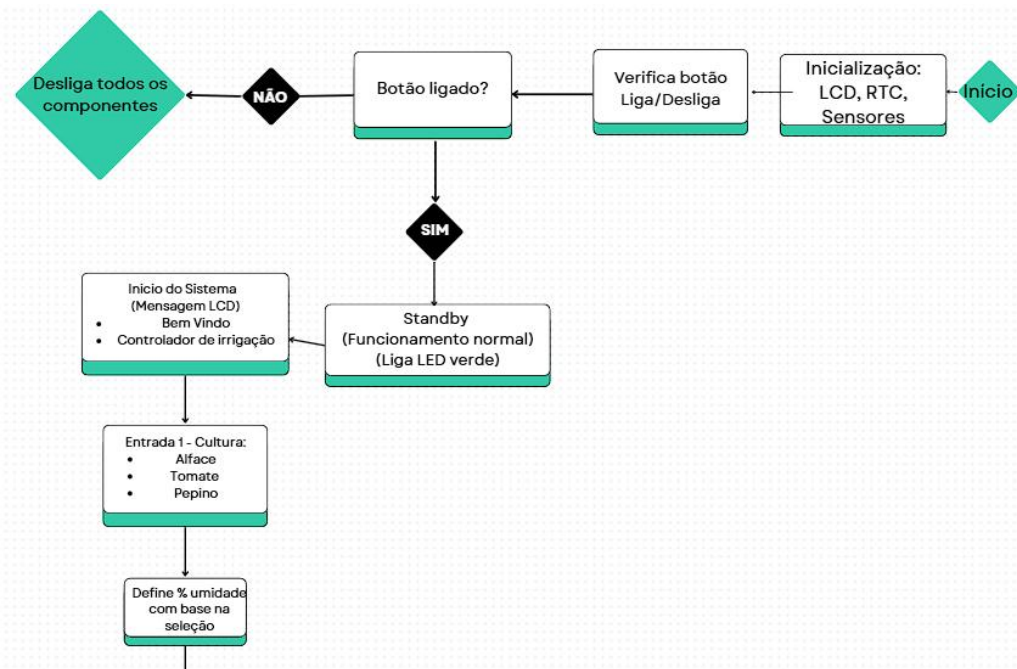
fluxograma o seguimento da estruturação lógica que o sistema usará, iniciando quando o botão liga/desliga é pressionado, verificando nível de reservatório, erros e somente após iniciando a lógica de irrigação. Para melhor visualização do fluxograma na Figura 13 foram seccionados em; fluxo inicial, seguido pela figura 14 o fluxograma de reservatório baixo/erro e na figura 16 o loop de irrigação.

Figura 12 – Fluxograma da lógica do sistema.



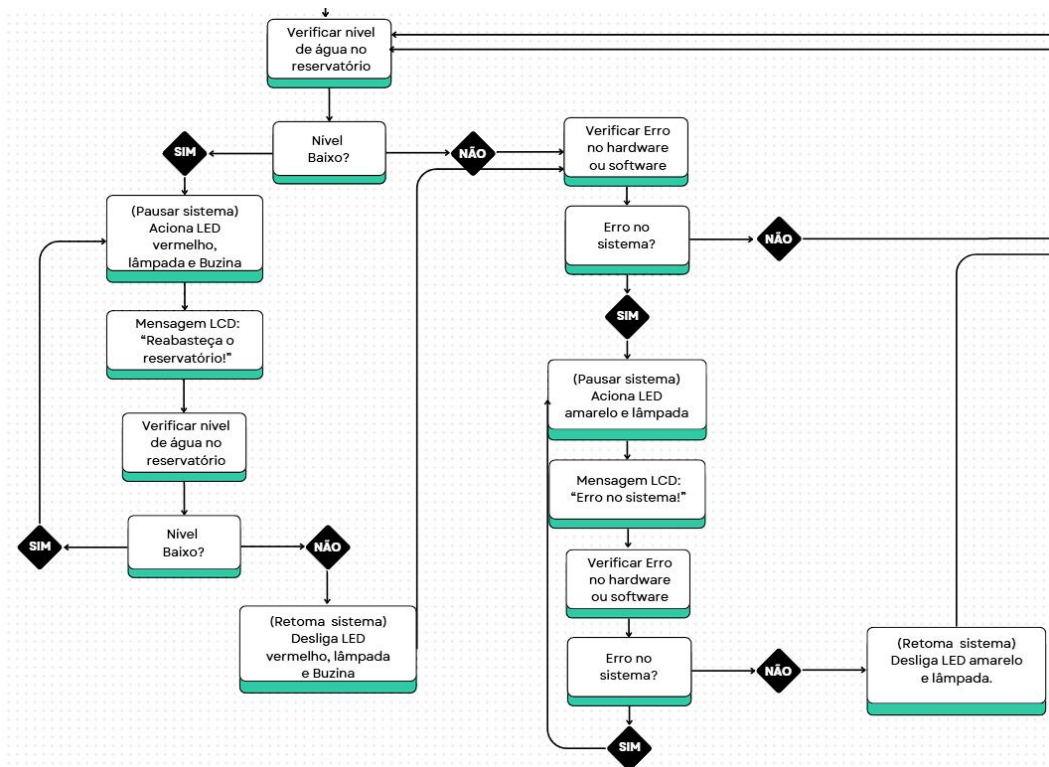
Fonte: Autor.

Figura 13 - Início do fluxograma.



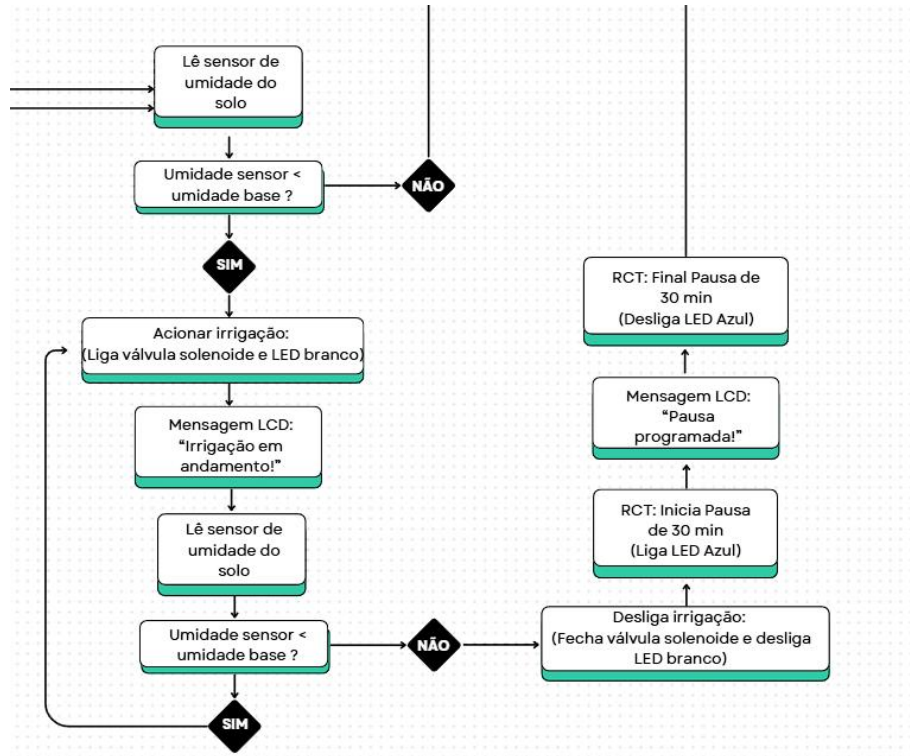
Fonte: Autor.

Figura 14 - Fluxograma de reservatório baixo e erro.



Fonte: Autor.

Figura 15 - Fluxograma do sistema de irrigação.



Fonte: Autor.

5.2 Funcionamento Simplificado da lógica do sistema

5.2.1 Início

- Inicia:
 - RCT.
 - LCD.
 - Sensores.

5.2.2 Iniciando sistema

- Verifica se o botão (liga/desliga) está pressionado.

5.2.3 Alerta de Nível de Reservatório

- Sensor de nível monitora a água no reservatório.

- Em caso de nível baixo, o LED vermelho, a lâmpada de emergência e o buzzer são ativados.

5.2.4 Indicação de Erro

- Em caso de falhas, o LED amarelo e a lâmpada de emergência são acionados.

5.2.5 Monitoramento da Umidade do Solo

- Sensor de umidade calibra os valores de 0 = (100%) a 700 = (0%) com base na leitura elétrica.
- Umidade ideal por cultura:
 - Alface: 70%
 - Tomate: 70%
 - Pepino: 50%
- O controlador aciona a irrigação se o valor da umidade estiver abaixo do ideal.

5.2.6 Pausa Pós-Irrigação

- Após atingir o nível de umidade desejado, o sistema entra em pausa por 30 minutos, controlado por um módulo RTC.

5.2.7 Interface com o Usuário

- Display LCD 16x2 apresenta
 - Boas-vindas ao ligar.
 - Status do sistema (irrigando, pausa, erro, nível baixo de água).
 - Umidade do solo em tempo real.

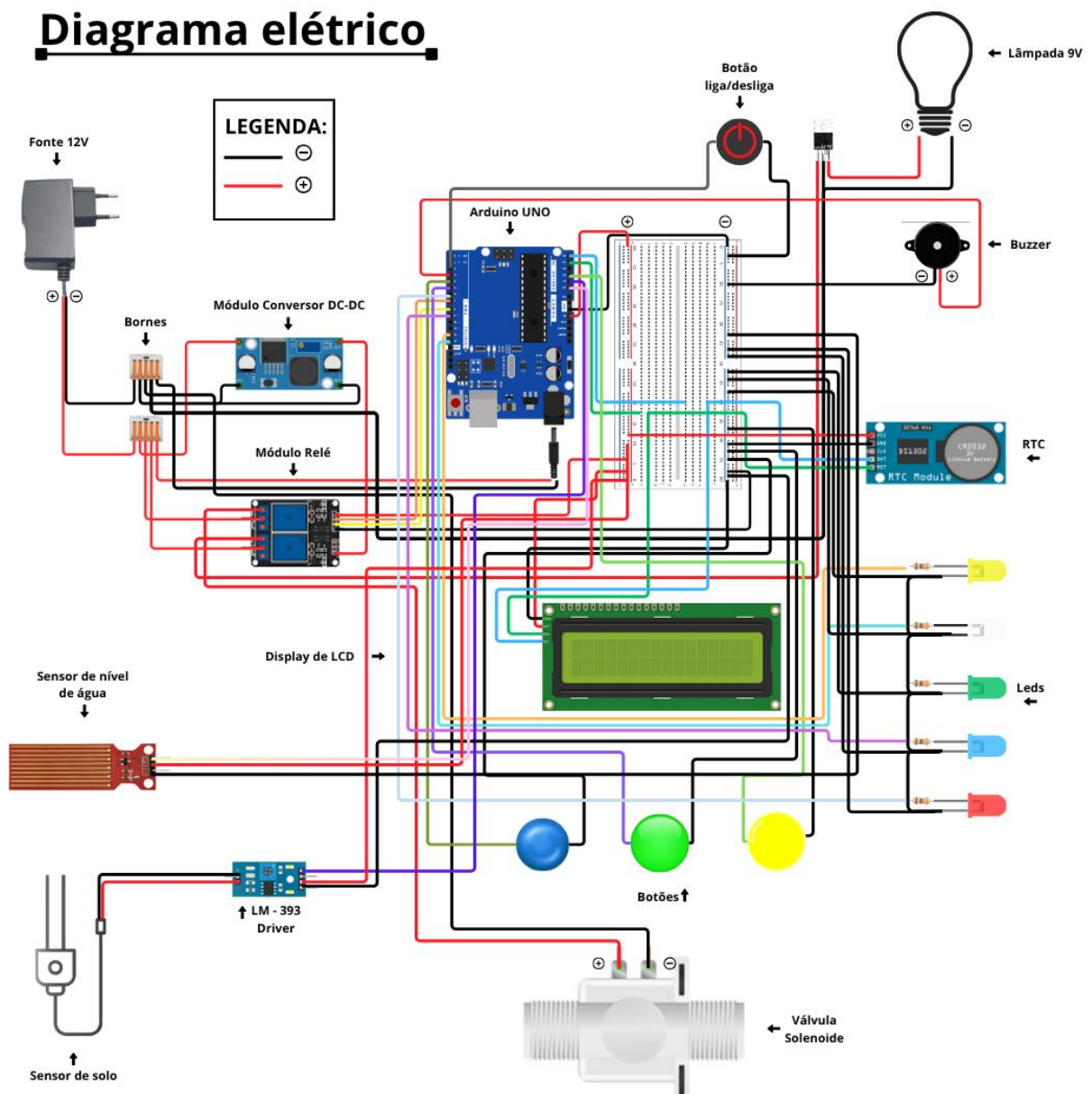
5.2.8 Quatro botões no painel

- Três para seleção de cultura (alface, tomate, pepino).
- Um para ligar/desligar o sistema.

5.3 Conexões

Aqui neste tópico é descrito como serão as principais conexões dos componentes do projeto e do Arduino UNO. E na figura 11 é possível visualizar como ficam as ligações dos componentes com o Arduino.

Figura 16 – Diagrama de interligação do Arduino e seus componentes



Fonte: Autor.

5.3.1 Guia de ligações elétricas

Abaixo estão as descrições simplificadas das principais conexões que foram realizadas entre os componentes e o Arduino. Para garantir o funcionamento correto e a integridade do sistema, o Arduino será alimentado por uma fonte externa de 12V DC, e todos os componentes estarão conectados conforme destacado abaixo. Na tabela 01 estão descritas as conexões de alimentação para a correta distribuição da energia, na Tabela 02 estão destacadas as conexões de controle e sensores no circuito de baixa tensão e Tabela 03 mostra as conexões de potência.

Tabela 02 - Conexões de alimentação e distribuição de energia.

Componente	Conexão de origem	Conexão de destino	Observações
Fonte de alimentação 12V	Positivo +	Borne +	Ponto de partida de toda a energia positiva.
	Negativo -	Borne -	Ponto de partida de toda energia negativa.
Arduino UNO	Borne +	Plug P4	Conectado a entrada P4 fêmea do Arduino.
	Borne -	Plug P4	
Módulo conversor buck DC-DC	Borne +	Entrada IN+ do conversor	Alimenta o conversor com 12V.
	Borne -	Entrada IN- do conversor	
Módulo relé	Saída OUT+ (5V) do conversor	Pino RY-VCC do relé	O jumper entre VCC e RY-VCC foi removido
	Saída OUT- (GND) do conversor	Borne -	Conectado ao GND comum.

Tabela 03 - Conexões de controle e sensores.

Componente	Pino do componente	Pino do Arduino	Observações
	GND	GND	

Display LCD	VCC	5V		
	SDA	A4		
	SCL	A5		
Módulo RTC	GND	GND		
	VCC	5V		
	SDA	A4		
	SCL	A5		
Sensor de nível de água	VCC	5V		
	GND	GND		
	S	A0		
Sensor de umidade de solo	VCC	5V		
	GND	GND		
	AOUT	A1		
Módulo relé (lógica)	VCC	5V	Alimenta somente a parte lógica do relé.	
	GND	GND		
	IN1	D7		
	IN2	D8		
Botão Liga/Desliga	-	GND		
	+	D2		
Botão Alface	-	GND		
	+	D4		
Botão pepino	-	GND		
	+	A2		
Botão tomate	-	GND		
	+	D5		
LED Verde	Ânodo +	A3		Ânodo + dos LED's com resistor de 220Ω
	Catodo -	GND		
LED Branco	Ânodo +	D11		
	Catodo -	GND		
LED Azul	Ânodo +	D9		
	Catodo -	GND		
LED Vermelho	Ânodo +	D6		

	Catodo -	GND	
LED Amarelo	Ânodo +	D10	
	Catodo -	GND	
Buzzer	Positivo +	D3	
	Negativo -	GND	

Tabela 04 - Conexões de potência.

Componente	Pino do componente	Conexão de origem	Observações
Válvula solenoide	Positivo +	NO do canal 1 do relé	COM do canal 1 recebe 12V da fonte.
	Negativo -	Borne -	
Regulador de tensão 7809	Pino de entrada Input	NO do canal 2 do relé	
	Pino do meio Ground	Borne -	
Lâmpada 9V	Positivo +	Pino de saída output do 7809	
	Negativo -	Borne -	

6. RESULTADOS E DISCUSSÕES

Neste tópico serão apresentados os resultados obtidos com o desenvolvimento do protótipo do controlador de irrigação automatizado, detalhando as etapas de montagem, os problemas encontrados durante a montagem e implementação do sistema, as soluções adotadas e a análise do sistema final. Ainda veremos as modificações realizadas em relação ao escopo original, análise de custos e potenciais escalonamentos do projeto.

6.1 Etapas de desenvolvimento do protótipo

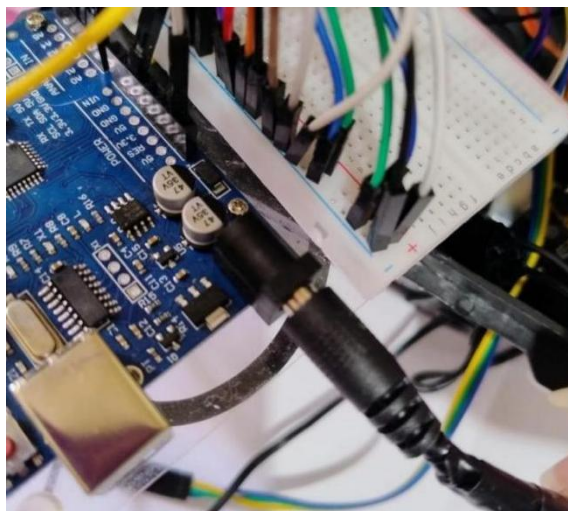
Primeiramente foi dada a construção do projeto em duas fases: a montagem mecânica e a integração eletrônica. Na fase mecânica, todos os componentes foram dispostos e fixados em um quadro elétrico residencial padrão, visando a organização, proteção dos circuitos e a facilidade de acesso para manutenção. O layout foi planejado para posicionar LDC e os botões de controle na parte frontal do painel, garantindo uma interface de usuário intuitiva e acessível.

A fase de integração eletrônica, foi a mais crítica e seguiu um estudo mais aprofundado para garantir a estabilidade e segurança do sistema. O sistema de alimentação foi o ponto central. Tentando-se métodos mais simples inicialmente e então optou-se por uma distribuição de energia mais robusta a partir de uma única fonte de 12V e 2A. Sendo a energia distribuída em paralelo para três diferentes circuitos.

6.1.1 Circuito de controle do arduino

A fonte de 12V alimenta a placa de forma direta pela entrada Vin, sendo está uma entrada Jack P4 fêmea, a qual vem acoplada ao arduino. Como pode ser visto na figura 17, o regulador de tensão interno do microcontrolador tem a função de gerar os 5V necessários para o Arduino e para a parte da alimentação lógica de periféricos de baixo consumo, como por exemplo o display LCD, o módulo RTC, etc...

Figura 17 - Entrada Jack P4 no arduino e conector P4 da fonte

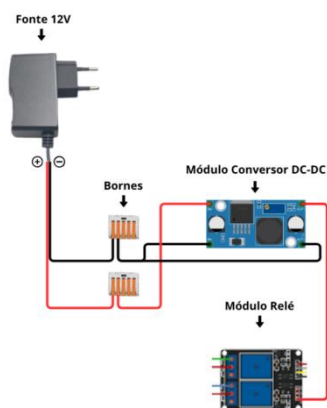


Fonte: Autor.

6.1.2 Circuito de potência das bobinas dos relés

Durante a montagem e experimentação do projeto, foram possíveis notar, instabilidades no sistema, ocorrendo uma sobrecarga nos componentes, para solucionar este problema foi introduzindo um modulo conversor DC - DC. Este componente tem a função de receber os 12V da fonte principal e convertê-los em 5V estáveis, dedicados exclusivamente a alimentar as bobins eletromecânicas do módulo relé, via pino RY-VCC, com o jumper de acoplamento removido. Abaixo podemos ver a imagem do modulo conversor e na figura 18 podemos ver as ligações eletrônicas dos componentes.

Figura 18 - Diagrama das ligações eletrônicas.

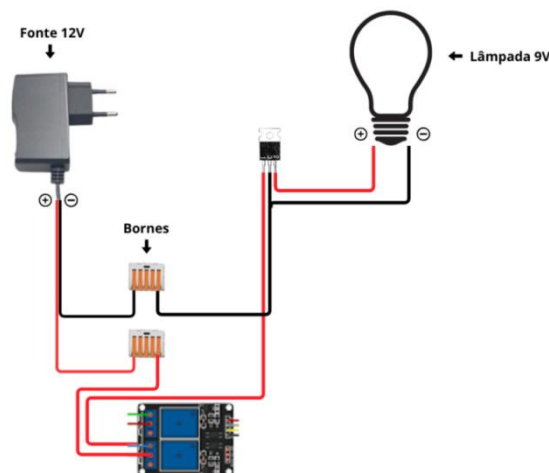


Fonte: Autor.

6.1.3 Circuito de potência dos atuadores

A energia que vem da fonte também é direcionada de maneira independente, para os terminais COM do módulo relé. Um dos canais controla a válvula solenoide de 12V e o outro canal controla a lâmpada de alerta 9V, pode-se notar que as tensões são diferentes, assim para adequar a tensão, foi inserido um regulador de tensão linear 7809, que fica posicionada entre o terminal NO do relé e o terminal positivo da lâmpada. Como demonstrado na figura 19. Todas as conexões de alimentação e de sinal foram realizadas conforme o diagrama esquemático, Figura 16, com a utilização de bornes para a distribuição de energia e garantindo que todos os componentes compartilhassem um ponto de GND comum para assegurar a integridade de todos os sinais elétricos.

Figura 19 - Ligações da lâmpada com o módulo relé.



Fonte: Autor.

6.2 Programação do Arduino

A programação do controlador de irrigação foi desenvolvida em linguagem C++, utilizando a interface de programação Arduino IDE, que é a oficial da plataforma. A arquitetura do software foi projetada para ser, não-bloqueante e orientada a eventos, garantindo que o sistema de respostas rápidas e seja capaz de

gerenciar múltiplas tarefas de monitoramento e controle ao mesmo tempo. Para a interação com os componentes, foram utilizadas bibliotecas padrão, como:

- Wire.h - para comunicação do barramento 12C.
- LiquidCrystal_I2C.h - para o controle do display.
- RTCLib.h - para gestão precisa do tempo com o módulo RTC.

O centro da lógica de controle foi implementado através de uma máquina de estados finitos (FSM), que se trata de um modelo computacional abstrato que representa um sistema com número finito de estados. Esta abordagem foi a escolhida por sua capacidade de gerenciar de forma clara e organizada as diferentes fases operacionais do sistema, evitando a complexidade e a instabilidade de um código linear. Foram definidos múltiplos estados, cada um representando uma condição de funcionamento específico do controlador:

- OFF - desligado.
- STARTING - sequência de inicialização.
- STANDBY - seleção de cultura.
- RUNNING - monitoramento e irrigação ativa.
- PAUSED - pausa pós-irrigação.
- LOW_WATER_MODE - alerta de reservatório baixo.
- ERROR_COUNTDOWN - alerta de falha de hardware com desligamento temporizado.

A estrutura fundamental do código reside nas funções:

- setup()
- loop()

Conforme o padrão da plataforma Arduino. Na função setup(), são realizadas as configurações iniciais: os pinos digitais e analógicas são configuradas como entrada INPUT_PULLUP ou saída OUTPUT, as bibliotecas dos componentes são inicializados e o sistema é colocado no estado inicial OFF. A função loop(), atua como o motor principal do sistema, executando o ciclo contínuo. Sua lógica foi

projetada como uma estrutura de verificação prioritária, a cada ciclo o programa avalia as condições críticas de erro e alerta de nível de reservatório primeiro. Se uma dessas condições for detectada o estado do sistema é imediatamente alterado para o modo de alerta correspondente, sobrepondo-se a qualquer operação normal. Somente em funcionamento normal, sem condições de falhas, a lógica prossegue para a máquina de estados, onde uma estrutura, switch-case executa as tarefas específicas do estado operacional que seja o atual.

Para modularizar o código e garantir sua clareza, foram criadas funções auxiliares. A função `atualizarDisplay()` centraliza toda a comunicação com o display LCD, exibindo mensagens contextuais e dados em tempo real, como a porcentagem de umidade, de acordo com o estado vigente do sistema. A função `ligarAlarmes()` é responsável por acionar a combinação correta de atuadores (LEDs, lâmpada e buzzer) conforme o tipo de alerta, garantindo uma resposta consistente e clara para o usuário. A gestão da pausa pós-irrigação de 30 minutos é realizada de forma assíncrona, utilizando o módulo RTC. Ao invés de travar o processador com um longo `delay()`, o sistema registra o horário final da pausa e, a cada ciclo do `loop()`, compara o tempo atual com o tempo final, liberando o processador para continuar monitorando outras condições. De forma análoga, a lógica de recuperação de erro foi implementada para que, mesmo em estado de alerta, o sistema continue a verificar se a condição de falha foi sanada, permitindo um retorno automático à operação normal sem a necessidade de reinicialização manual. Na figura 20 pode ser visto a imagem da programação completa utilizada no protótipo.

Figura 20 - Programação do protótipo.

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <RTClib.h>
const int PINO_SENSOR_AGUA = A0; const int PINO_SENSOR_SOLO = A1;
const int PINO_BOTAO_LIGA_DESLIGA = 2; const int PINO_BOTAO_ALFACE = 4;
const int PINO_BOTAO_TOMATE = 5; const int PINO_BOTAO_PEPINO = A2;
const int PINO_RELE_VALVULA = 7; const int PINO_RELE_LAMPADA = 8;
const int PINO_BUZZER = 3; const int PINO_LED_VERMELHO = 6;
const int PINO_LED_AZUL = 9; const int PINO_LED_AMARELO = 10;
const int PINO_LED_LARANJA = 11; const int PINO_LED_VERDE = A3;

// --- Configurações do Sistema ---
const int SENSOR_SOLO_SECO = 1080; const int SENSOR_SOLO_UMIDO = 0;
// Lógica do sensor de água: Alarme quando a leitura for MENOR que o limite.
const int LIMIAR_NIVEL_AGUA_BAIIXO = 300;
const int UMIDADE_ALFACE = 70; const int UMIDADE_TOMATE = 70; const int UMIDADE_PEPINO = 50;
const long DURACAO_PAUSA_MINUTOS = 30;
const long DURACAO_ERRO_MINUTOS = 30;

// --- Objetos e Variáveis Globais ---
LiquidCrystal_I2C lcd(0x27, 16, 2);
RTC_DS3231 rtc;
enum SystemState { OFF, STARTING, STANDBY, RUNNING, PAUSED, ERROR_COUNTDOWN, LOW_WATER_MODE };
SystemState estadoSistema = OFF;
bool estadoBotaoLigaAnterior = HIGH;
unsigned long ultimoDebounce = 0; const long intervaloDebounce = 200;
int umidadeBaseSelecionada = 0; String culturaSelecionada = "";
int umidadeAtualPorcento = 0;
DateTime tempoFimPausa; DateTime tempoFimErro;
bool irrigandoAgora = false;
unsigned long timestampAlerta = 0;

// --- Declaração das Funções (Protótipos) ---
void desligarTudo();
void atualizarDisplay();
void ligarAlarmes(char tipo);

// --- SETUP ---
void setup() {
  Serial.begin(9600); Serial.println("\n\n--- INICIANDO CONTROLADOR V15.1 ---");
  pinMode(PINO_RELE_VALVULA, OUTPUT); pinMode(PINO_RELE_LAMPADA, OUTPUT); pinMode(PINO_BUZZER, OUTPUT);
  pinMode(PINO_LED_VERMELHO, OUTPUT); pinMode(PINO_LED_AZUL, OUTPUT); pinMode(PINO_LED_AMARELO, OUTPUT);
  pinMode(PINO_LED_LARANJA, OUTPUT); pinMode(PINO_LED_VERDE, OUTPUT);
  pinMode(PINO_BOTAO_LIGA_DESLIGA, INPUT_PULLUP); pinMode(PINO_BOTAO_ALFACE, INPUT_PULLUP);
  pinMode(PINO_BOTAO_TOMATE, INPUT_PULLUP); pinMode(PINO_BOTAO_PEPINO, INPUT_PULLUP);
  pinMode(PINO_SENSOR_AGUA, INPUT); pinMode(PINO_SENSOR_SOLO, INPUT);

```

```

Wire.begin(); lcd.init(); rtc.begin();
desligarTudo(); estadoSistema = OFF;
}

// --- LOOP PRINCIPAL ---
void loop() {
  bool estadoBotaoLigaAtual = digitalRead(PINO_BOTAO_LIGA_DESLIGA);
  if (estadoBotaoLigaAtual == LOW && estadoBotaoLigaAnterior == HIGH) {
    if (millis() - ultimoDebounce >= intervaloDebounce) {
      ultimoDebounce = millis();
      if (estadoSistema == OFF) {
        estadoSistema = STARTING;
      } else {
        lcd.clear(); lcd.backlight(); lcd.print("Ate logo!");
        delay(1500); estadoSistema = OFF;
      }
    }
  }
}

estadoBotaoLigaAnterior = estadoBotaoLigaAtual;

if (estadoSistema == OFF) { desligarTudo(); return; }

// --- ESTRUTURA DE CONTROLE PRIORITÁRIA ---
Wire.beginTransmission(0x68);
if (Wire.endTransmission() != 0) {
  if (estadoSistema != ERROR_COUNTDOWN) {
    tempoFimErro = rtc.now() + TimeSpan(0, 0, DURACAO_ERRO_MINUTOS, 0);
  }
  estadoSistema = ERROR_COUNTDOWN;
}

if (estadoSistema != ERROR_COUNTDOWN) {
  if (analogRead(PINO_SENSOR_AGUA) < LIMIAR_NIVEL_AGUA_BAIIXO) {
    if (estadoSistema != LOW_WATER_MODE) {
      timestampAlerta = millis();
    }
    estadoSistema = LOW_WATER_MODE;
  } else {
    if (estadoSistema == LOW_WATER_MODE) {
      estadoSistema = STANDBY;
    }
  }
}
}

```

```

switch (estadoSistema) {
case STARTING:
  ligarAlarmes('N');
  if (analogRead(PINO_SENSOR_AGUA) < LIMIAR_NIVEL_AGUA_BAIXO) { estadoSistema = LOW_WATER_MODE; timestampAlerta = millis(); break; }
  Wire.beginTransmission(0x68);
  if (Wire.endTransmission() != 0) { estadoSistema = ERROR_COUNTDOWN; tempoFimErro = rtc.now() + TimeSpan(0, 0, DURACAO_ERRO_MINUTOS, 0); break; }

  lcd.backlight(); lcd.clear(); lcd.print("Bem vindo"); delay(2000);
  lcd.clear(); lcd.print("Controlador de"); lcd.setCursor(0, 1); lcd.print("irrigacao"); delay(2000);
  estadoSistema = STANDBY;
  break;

case STANDBY:
  ligarAlarmes('N'); digitalWrite(PINO_LED_VERDE, HIGH);
  if (digitalRead(PINO_BOTAO_ALFACE) == LOW) {
    umidadeBaseSelecionada = UMIDADE_ALFACE; lcd.clear(); lcd.print("Alface: " + String(umidadeBaseSelecionada) + "%"); delay(1500);
    estadoSistema = RUNNING;
  } else if (digitalRead(PINO_BOTAO_TOMATE) == LOW) {
    umidadeBaseSelecionada = UMIDADE_TOMATE; lcd.clear(); lcd.print("Tomate: " + String(umidadeBaseSelecionada) + "%"); delay(1500);
    estadoSistema = RUNNING;
  } else if (digitalRead(PINO_BOTAO_PEPINO) == LOW) {
    umidadeBaseSelecionada = UMIDADE_PEPINO; lcd.clear(); lcd.print("Pepino: " + String(umidadeBaseSelecionada) + "%"); delay(1500);
    estadoSistema = RUNNING;
  }
  break;

case RUNNING:
  umidadeAtualPorcento = map(analogRead(PINO_SENSOR_SOLO), SENSOR_SOLO_SECO, SENSOR_SOLO_UMIDO, 0, 100);
  umidadeAtualPorcento = constrain(umidadeAtualPorcento, 0, 100);
  digitalWrite(PINO_LED_AZUL, LOW);
  if (umidadeAtualPorcento < umidadeBaseSelecionada) {
    irrigandoAgora = true;
    digitalWrite(PINO_RELE_VALVULA, HIGH); digitalWrite(PINO_LED_LARANJA, HIGH);
  } else {
    irrigandoAgora = false;
    digitalWrite(PINO_RELE_VALVULA, LOW); digitalWrite(PINO_LED_LARANJA, LOW);
    estadoSistema = PAUSED;
    tempoFimPausa = rtc.now() + TimeSpan(0, 0, DURACAO_PAUSA_MINUTOS, 0);
  }
  break;

case PAUSED:
  irrigandoAgora = false;
  digitalWrite(PINO_LED_LARANJA, LOW); digitalWrite(PINO_RELE_VALVULA, LOW);
  digitalWrite(PINO_LED_AZUL, HIGH);
  if (rtc.now() >= tempoFimPausa) { estadoSistema = RUNNING; }
  break;
}

```

```

    case PAUSED:
        irrigandoAgora = false;
        digitalWrite(PINO_LED_LARANJA, LOW); digitalWrite(PINO_RELE_VALVULA, LOW);
        digitalWrite(PINO_LED_AZUL, HIGH);
        if (rtc.now() >= tempoFimPausa) { estadoSistema = RUNNING; }
        break;

    case ERROR_COUNTDOWN:
        ligarAlarmes('E');
        Wire.beginTransmission(0x68);
        if (Wire.endTransmission() == 0) {
            estadoSistema = STANDBY;
        } else if (rtc.now() >= tempoFimErro) {
            estadoSistema = OFF;
        }
        break;

    case LOW_WATER_MODE:
        ligarAlarmes('A');
        break;
}

atualizarDisplay();
delay(200);
}

// --- IMPLEMENTAÇÃO DAS FUNÇÕES AUXILIARES ---

void desligarTudo() {
    digitalWrite(PINO_RELE_VALVULA, LOW);
    digitalWrite(PINO_RELE_LAMPADA, LOW);
    digitalWrite(PINO_BUZZER, LOW);
    digitalWrite(PINO_LED_VERMELHO, LOW);
    digitalWrite(PINO_LED_AZUL, LOW);
    digitalWrite(PINO_LED_AMARELO, LOW);
    digitalWrite(PINO_LED_LARANJA, LOW);
    digitalWrite(PINO_LED_VERDE, LOW);
    lcd.clear();
    lcd.noBacklight();
    culturaSelecionada = "";
    umidadeBaseSelecionada = 0;
    irrigandoAgora = false;
}

```

```

void ligarAlarmes(char tipo) {
    digitalWrite(PINO_LED_VERDE, LOW);
    digitalWrite(PINO_LED_LARANJA, LOW);
    digitalWrite(PINO_LED_AZUL, LOW);
    digitalWrite(PINO_LED_VERMELHO, LOW);
    digitalWrite(PINO_LED_AMARELO, LOW);
    digitalWrite(PINO_RELE_LAMPADA, LOW);
    digitalWrite(PINO_BUZZER, LOW);

    if (tipo == 'A') { // Alarme de Água
        digitalWrite(PINO_RELE_VALVULA, LOW); irrigandoAgora = false;
        digitalWrite(PINO_LED_VERMELHO, HIGH);
        digitalWrite(PINO_RELE_LAMPADA, HIGH);
        digitalWrite(PINO_BUZZER, HIGH);
    } else if (tipo == 'E') { // Alarme de Erro
        digitalWrite(PINO_RELE_VALVULA, LOW); irrigandoAgora = false;
        digitalWrite(PINO_LED_AMARELO, HIGH);
        digitalWrite(PINO_RELE_LAMPADA, HIGH);
        digitalWrite(PINO_BUZZER, HIGH);
    }
}

void atualizarDisplay() {
    lcd.clear();
    lcd.backlight();
    char bufferUmidade[5];
    sprintf(bufferUmidade, "%d%%", umidadeAtualPorcento);

    switch (estadoSistema) {
        case OFF: lcd.noBacklight(); lcd.clear(); break;
        case STANDBY: lcd.setCursor(0, 0); lcd.print("Selecione a cultura"); break;
        case RUNNING:
            if (irrigandoAgora) {
                lcd.setCursor(0, 0); lcd.print("Irrigando...");
                lcd.setCursor(0, 1); lcd.print("Umidade: " + String(bufferUmidade));
            } else {
                lcd.setCursor(0, 0); lcd.print("Sistema em func.");
                lcd.setCursor(0, 1); lcd.print("normal " + String(bufferUmidade));
            }
            break;
        case PAUSED: {
            lcd.setCursor(0, 0); lcd.print("Pausa programada!");
            TimeSpan restante = tempoFimPausa - rtc.now();
            char bufferPausa[9]; sprintf(bufferPausa, "Prox.: %02d:%02d", restante.minutes(), restante.seconds());
            lcd.setCursor(0, 1); lcd.print(bufferPausa);
            break;
        }

        case ERROR_COUNTDOWN: {
            lcd.setCursor(0, 0); lcd.print("Erro no sistema!");
            TimeSpan restante = tempoFimErro - rtc.now();
            char bufferTempo[9]; sprintf(bufferTempo, "%02d:%02d", restante.minutes(), restante.seconds());
            lcd.setCursor(0, 1); lcd.print("Desligando em " + String(bufferTempo));
            break;
        }
        case LOW_WATER_MODE:
            if (millis() - timestampAlerta < 4000) {
                lcd.setCursor(0, 0); lcd.print("Reservatorio");
                lcd.setCursor(0, 1); lcd.print("baixo!");
            } else {
                lcd.clear(); lcd.setCursor(0, 0); lcd.print("Reabasteca!");
                if (millis() - timestampAlerta >= 8000) {
                    timestampAlerta = millis();
                }
            }
            break;
    }
}

```

Fonte: Autor.

6.3 Avaliações do Sistema

Para validação do projeto, foram realizadas avaliações funcionais com o objetivo de verificar a precisão do protótipo montado. Estas avaliações foram realizadas em ambiente controlado, focando principalmente no desempenho do

hardware, na precisão dos sensores após a calibração e na resposta do sistema aos diferentes cenários de alerta e na usabilidade da interface do usuário. Cada teste foi projetado para garantir que o sistema atenda aos requisitos propostos.

6.3.1 Avaliação e Calibração dos Sensores

Esta avaliação teve como objetivo a verificação da precisão dos sensores de umidade do solo e de nível de água, sendo uma etapa fundamental para que o sistema tome a correta tomada de decisão da hora de irrigar.

Para o sensor de umidade do solo, a calibração foi realizada de forma empírica para estabelecer os pontos de referência correspondentes a 0% e 100% de umidade. Utilizando um script de diagnóstico no Arduino (figura 21), foram capturados os valores analógicos brutos do sensor em duas condições, com o sensor em solo totalmente seco (representando 0% solo seco) e com o sensor em solo saturado de água (representando 100% solo saturado). Os testes práticos demonstraram que o valor para solo seco era de aproximadamente 1080, enquanto para solo saturado era próximo a 0. Com estes dados, a função `map()` no software foi ajustada para converter a faixa de leitura do sensor em uma escala percentual de 0 a 100%, garantindo que a umidade exibida ao usuário fosse um reflexo real da condição do solo.

De forma similar, o sensor de nível de água foi avaliado simulando condições de "reservatório cheio" (sensor submerso na água) e "reservatório vazio" (sensor exposto ao ar totalmente fora da água). Foi observado que o componente retornava uma leitura analógica baixa, próxima a 0-5, quando seco e uma leitura mais alta, próxima a 240 - 245 quando submerso. Este resultado levou a um ajuste mais preciso na lógica do software. Foi estabelecido um limite para o alarme (`LIMIAR_NIVEL_AGUA_BAIKO`) no valor de 100, e a condição de comparação foi definida como leitura < 100 para acionar o alerta. Este método garantiu que o sistema detectasse a condição de nível baixo, de forma confiável.

Figura 21 - Script de diagnóstico.

```
calibracaoensores.ino
1  const int PINO_SENSOR_PARA_TESTAR = A0;
2
3  void setup() {
4      Serial.begin(9600);
5      Serial.println("\n\n--- INICIANDO SCRIPT DE DIAGNOSTICO ---");
6      Serial.print("Lendo valores do pino: ");
7      Serial.println(PINO_SENSOR_PARA_TESTAR);
8      Serial.println("-----");
9  }
10
11 void loop() {
12     int leituraDoSensor = analogRead(PINO_SENSOR_PARA_TESTAR);
13
14     Serial.print("Leitura atual: ");
15     Serial.println(leituraDoSensor);
16
17     delay(1000);
18 }
19
```

Fonte: Autor.

6.3.2 Avaliação de Alertas e Indicadores

Os indicadores visuais, LEDs e lâmpada, e sonoros, buzzer, foram submetidos a testes simulados para verificar a resposta do sistema a condições operacionais e de falha. A simulação de nível baixo de água, foi realizada ao remover o sensor do reservatório durante a operação, resultando na ativação imediata e simultânea do LED vermelho, da lâmpada de 9V e do buzzer, enquanto o display LCD exibia a mensagem de alerta correspondente, validando a eficácia do sistema de alarme (Figura 22). A falha de hardware foi induzida desconectando o módulo RTC, o que fez o controlador entrar em modo de erro, acionando o LED amarelo e os demais alarmes e iniciando a contagem regressiva de segurança (figura 23). Adicionalmente, foi verificado que os LEDs verde, branco e azul indicaram corretamente os estados de "sistema funcionamento normal" (Figura 24), "irrigando" (Figura 25) e "pausa pós-irrigação" (Figura 26), também foram avaliadas a clareza das mensagens exibidas no LCD, incluindo mensagens de boas vindas (Figura 26), seleção de cultura (Figura 27) e mensagem de desligamento (Figura 28), fornecendo um feedback visual claro e imediato. A confiabilidade dos alertas foi, portanto, confirmada.

Figura 22 - Mensagem de reservatório baixo.



Fonte: Autor.

Figura 23 - Mensagem de erro no sistema.



Fonte: Autor.

Figura 24 - Sistema em funcionamento normal.



Fonte: Autor.

Figura 25 - Irrigação em funcionamento.



Fonte: Autor.

Figura 26 - Pausa pós irrigação.



Fonte: Autor.

Figura 27 - Mensagem de boas vindas.



Fonte: Autor.

Figura 28 - Seleção de cultura.



Fonte: Autor.

Figura 29 - Mensagem de desligamento.

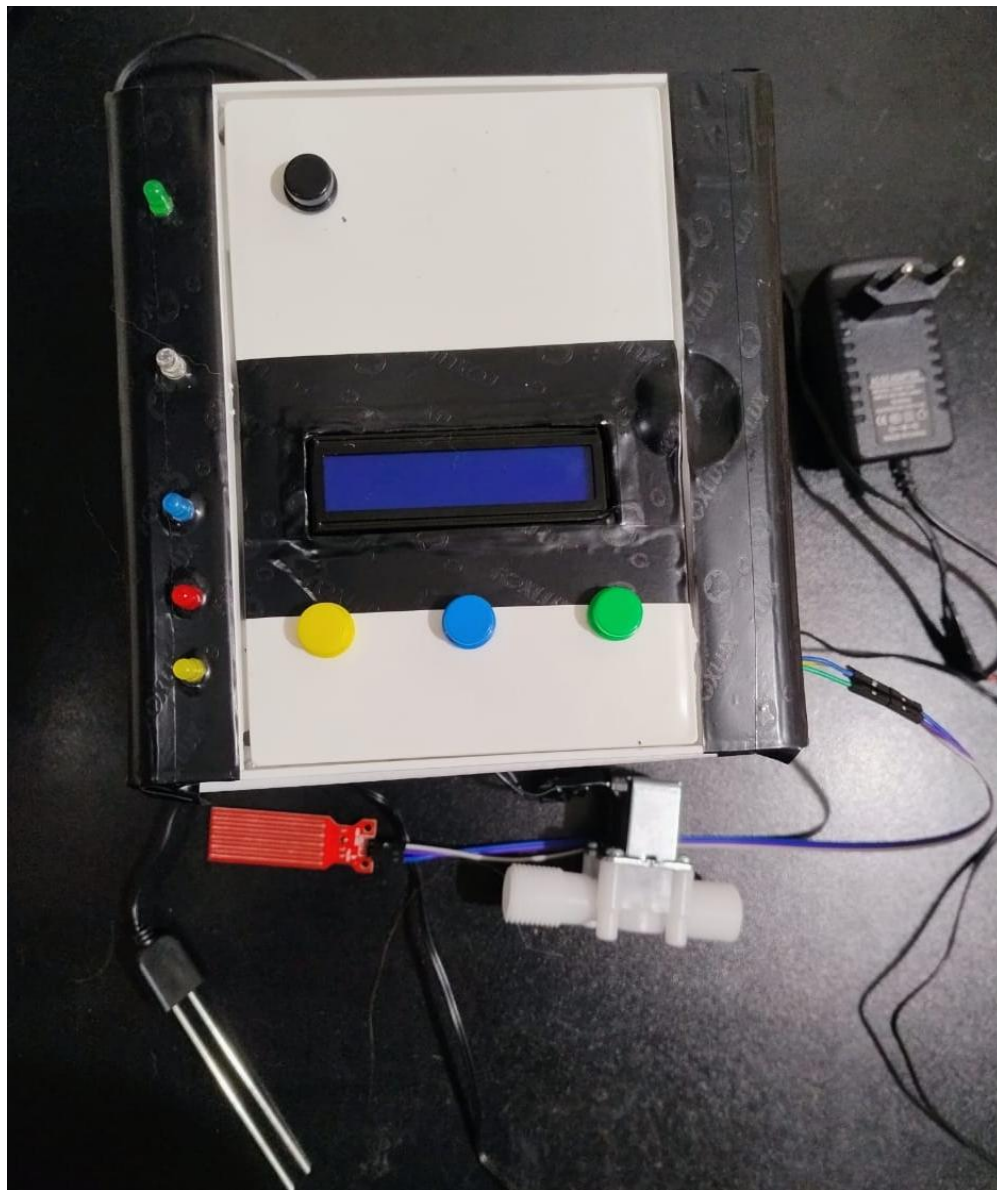


Fonte: Autor.

6.3.3 Avaliação do Painel de Controle e Interface do Usuário

A interface foi avaliada através de testes de usabilidade do painel de controle e da verificação das mensagens exibidas no display LCD. Os quatro botões de pressão foram operados repetidamente para confirmar sua funcionalidade. O botão "Liga/Desliga" controlou o estado geral do sistema de forma segura, enquanto os botões de seleção de cultura permitiram a alteração do parâmetro de umidade alvo, com o sistema confirmando a seleção no LCD, demonstrando que a interface é intuitiva e funcional. A validação do display LCD confirmou que todas as mensagens de status, umidade em tempo real e os alertas críticos são exibidos de forma clara e precisa, em conformidade com o que foi projetado, garantindo uma comunicação eficaz com o usuário e cumprindo os objetivos desta avaliação. Figura 30 visão frontal do protótipo.

Figura 30 - Imagem frontal do protótipo.



Fonte: Autor.

6.4 Modificações realizadas do projeto original

A versão final do protótipo representa uma evolução significativa em relação ao plano conceitual inicial, as modificações foram motivadas pela necessidade de superar os desafios práticos de implementação e aumentar a eficácia geral do sistema.

A principal e mais impactante modificação foi a arquitetura de alimentação elétrica. O plano inicial descrevia genericamente o uso de uma fonte 12V, sem o

detalhamento da distribuição de energia para os diferentes componentes, que tinham requisitos de tensão variados (12V, 9V, 5V). A implementação final resultou em um sistema de múltiplas tensões derivado de uma única fonte, utilizando componentes dedicados para gerar as tensões necessárias de forma estável. Essa abordagem nova garante a proteção da placa arduino contra sobrecargas.

O software também teve evoluções, o esboço de código original era linear e não previa a complexidade da gestão de estados de falha. O sistema encontrava a falha e não sabia quando sair desse estado. A versão final implementa à máquina estados de hierarquica. Também foram adicionadas funções como desligamento automático de segurança após trinta minutos de erro e a recuperação inteligente após a resolução de uma falha, estas atualizações não estavam no escopo original, mas foram adicionadas para tornar o controlador de irrigação verdadeiramente autônomo e seguro.

A calibração dos sensores foi outra área de modificação, inicialmente partiu-se de premissas teóricas sobre a faixa de operação dos sensores. A prática demonstrou que cada sensor possui um comportamento único. O processo de desenvolvimento forçou a criação de rotinas de diagnóstico para uma calibração empírica, resultando em limiares e lógicas de comparação ajustados à realidade do hardware utilizado, o que acarretou no aumento da precisão e a confiabilidade do sistema.

6.5 Capacidade de escalonamento do projeto e mudanças futuras

Embora tenha sido projetado e testado como um projeto de pequena escala, a arquitetura fundamental do controlador possui potencial de escalonamento para aplicações em áreas maiores, com adaptações.

O microcontrolador e a lógica de programação são como o “cérebro” do sistema, que é perfeitamente capaz de gerenciar áreas maiores, a escalabilidade do projeto reside principalmente na modificação do circuito de potência. Para o controle de bombas de irrigação de maior porte, que operam em 110V ou 220V AC, seria necessário substituir o módulo de relé de 5V por relés de estado sólido (SSRs) ou contadores de maior capacidade. Estes dispositivos são projetados para controlar altas tensões e correntes, mas ainda podem ser acionados pelos mesmos sinais de

5V enviados pelo Arduino, mantendo o princípio de isolamento entre o controle e a potência.

Já para gerenciamento de múltiplas zonas de irrigação independentes, o projeto teria de ser expandido, utilizando uma placa com mais portas, como o arduino mega ou ate mesmo expansores de portas projetados para o arduino. Isso permitiria a conexão de múltiplos sensores de umidade e o controle de múltiplas válvulas solenoides se necessário, cada uma em um canal de relé diferente. A lógica do software precisaria ser expandida também para loops que verificassem cada zona individualmente, mas a estrutura base de estados permaneceria válida.

A alimentação dos relés de 5V teria de ter uma capacidade de corrente maior para acionar múltiplos relés simultaneamente se necessário. Por fim o conceito lógico do projeto é altamente escalonável, a adaptação para áreas maiores é somente uma questão de dimensionar corretamente os componentes da interface de potência.

6.6 Análise de custos

Conforme a Tabela 05 do projeto, o custo estimado dos componentes principais seria de aproximadamente R\$ 362,21.

Tabela 05 – Análise de custos.

Item	Quantidade	Valor Unitário (R\$)	Subtotal (R\$)
Arduino Uno (UNO)	1	55,00	55,00
Sensor de umidade anticorrosivo	1	20,00	20,00
Módulo RTC DS3231	1	18,00	18,00
Sensor de nível (boia ou simples)	1	10,00	10,00
Válvula solenoide 12V	1	45,00	45,00
Módulo Relé 12V (2 canais)	1	18,00	18,00
Display LCD 16x2 I2C	1	25,00	25,00
LEDs coloridos (kit com 5)	1	5,00	5,00
Buzzer Piezoelétrico	1	5,00	5,00
Botões (push-buttons)	4	1,00	4,00

Lâmpada LED 9V	1	10,00	10,00
Fonte 12V 2A	1	30,00	30,00
Protoboard e jumpers	1 kit	25,00	25,00
Quadro elétrico residencial	1	50,00	50,00
Módulo conversor DC-DC	1	15,00	15,00
Regulador de tensão	1	10,36	10,36
Resistores	18	0,15	2,70
TOTAL	38		~362,21

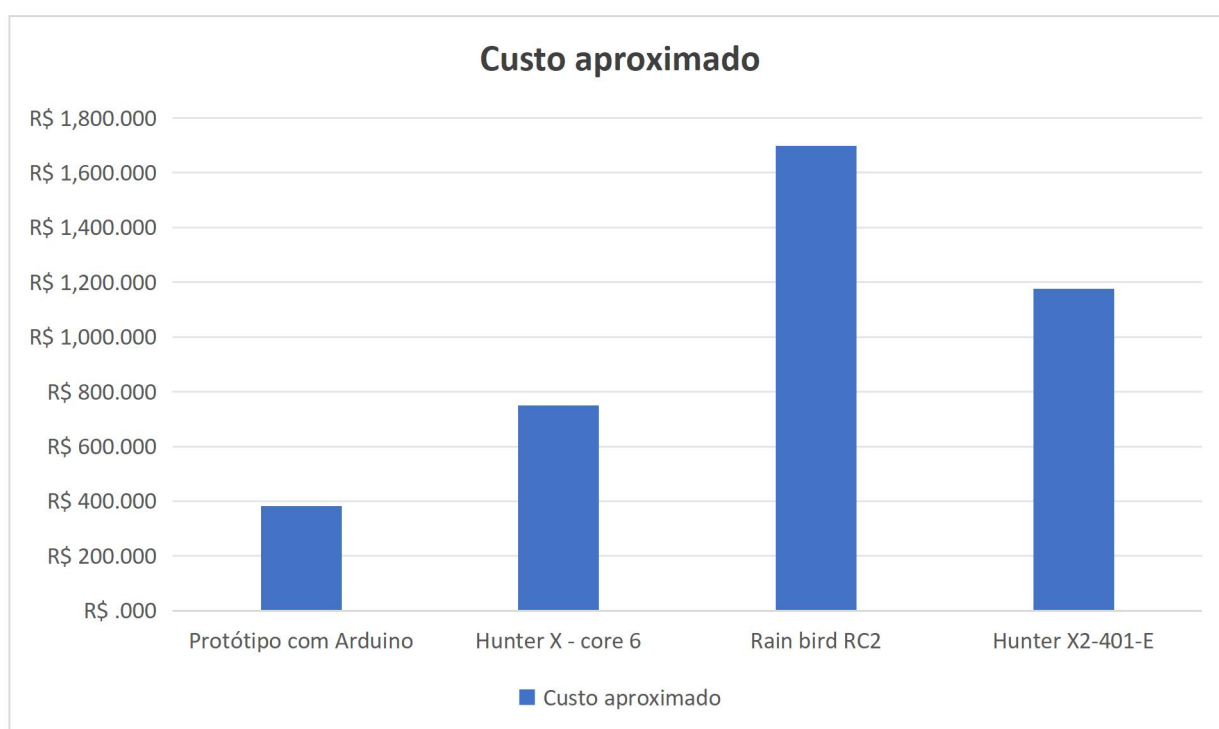
Considerando os valores de acréscimo e decréscimo, o custo total final do prototipo foi recalculado para aproximadamente R\$ 381,47. Apesar do aumento de cerca de 5%, o valor final ainda se enquadra na categoria de baixo custo, especialmente quando comparado a soluções comerciais de automação de irrigação. O investimento adicional nos módulos de conversão de tensão é um requisito fundamental para a confiabilidade e longevidade do projeto.

Para contextualizar a viabilidade econômica do projeto, realizou-se um levantamento de preços de controladores de irrigação comerciais disponíveis no mercado brasileiro. Os modelos selecionados para comparação foram:

- **Hunter X-Core 6** : Geralmente não possui Wi-Fi e opera com base em programação de tempo, mas permite a conexão de um sensor de chuva para interromper a rega. Os preços variam, em média, de R\$ 750 a R\$ 950.
- **Rain Bird ESP-TM2**: É um controlador baseado em tempo que, em algumas versões, está pronto para receber um módulo Wi-Fi (vendido separadamente por um custo adicional). Os valores ficam tipicamente na faixa de R\$ 800 a R\$ 1.000.
- **Controladores com Wi-Fi**: Modelos como o Hunter X2 com o módulo Wi-Fi já integrado oferecem controle via aplicativo e ajustes inteligentes baseados na meteorologia local via internet, porém ajustam somente os horários de irrigação definidos pelo usuário, não tomam a decisão direta de quando irrigar. O custo desses sistemas geralmente começa em torno de R\$ 1050 e pode facilmente ultrapassar os R\$ 2000.

Constatou-se que os sistemas comerciais com funcionalidades minimamente comparáveis — como a capacidade de resposta a sensores e o controle de pelo menos duas zonas — possuem um custo que varia significativamente, partindo de R\$ 750,00 para modelos de entrada e podendo ultrapassar R\$ 2.000,00 para controladores com interfaces mais avançadas. A comparação direta revela que o protótipo desenvolvido representa uma economia de, no mínimo, 40% em relação à alternativa comercial mais simples e uma redução superior a 70% em comparação com sistemas mais robustos, como podemos notar na Figura 31 que demonstra o gráfico da comparação de custo do protótipo proposto e soluções comerciais existentes no mercado. Vale ainda ressaltar que as opções

Figura 31 - Gráfico de comparação de preços.



Fontes: MercadoLivre.com, casadairrigação.com e drirrigação.com.

Este resultado valida o projeto como uma solução tecnológica acessível e funcional, representando uma alternativa viável frente às soluções comerciais de

automação de irrigação. O baixo investimento inicial, aliado à potencial economia de água e mão de obra, confere ao protótipo um alto potencial de retorno e ampla aplicabilidade prática, especialmente para pequenos agricultores, hortas comunitárias e projetos educacionais.

7. CONCLUSÃO

O desenvolvimento do protótipo de controlador de irrigação atingiu com sucesso seus objetivos principais, resultando em um sistema funcional, de baixo custo e com uma lógica de operação inteligente e segura. A construção deste projeto desde a concepção inicial até a versão final, foi um processo que reforçou princípios fundamentais da engenharia.

As adversidades encontradas, em especial a instabilidade elétrica, foram superadas com a aplicação de soluções técnicas padrão da indústria, como o uso do conversor dedicado, que agregou um nível de confiabilidade ao projeto que não estava previsto anteriormente. A versão final do controlador é capaz de gerenciar a irrigação de forma autônoma, adaptando-se a diferentes culturas, fornecendo uma interface clara ao usuário e os modos de segurança para proteger o sistema.

A análise final de custos, situou o custo total do protótipo em aproximadamente R\$ 392,21, confirmando a viabilidade do projeto como uma solução de baixo custo. Em termos de desempenho, o sistema opera com um tempo médio de resposta de 200 milissegundos por ciclo de verificação, garantindo um monitoramento praticamente em tempo real das condições do solo e do reservatório. A lógica de controle demonstrou ser precisa, gerenciando os parâmetros de umidade para as três culturas pré-definidas e acionando corretamente a válvula solenoide de 12V. A implementação de funcionalidades de segurança e eficiência foi validada, executando de forma confiável o ciclo de pausa pós-irrigação de 30 minutos controlado pelo módulo RTC.

O projeto demonstrou ser uma alternativa viável e acessível às soluções comerciais, com grande potencial de aplicação para pequenos produtores e como ferramenta educacional. As modificações e soluções implementadas não apenas resolveram os problemas práticos, mas também enriqueceram o projeto, transformando-o em um exemplo completo do ciclo de desenvolvimento de um produto de automação.

REFERÊNCIAS

ALMEIDA, A. C. dos S. **Desenvolvimento de um controlador de irrigação acionado mecanicamente pela tensão da água no solo**. 2012. Tese (Doutorado) – Escola Superior de Agricultura Luiz de Queiroz, Universidade de São Paulo, Piracicaba, 2012. Disponível em: <http://www.teses.usp.br/teses/disponiveis/11/11143/tde-04012013-103112/>. Acesso em: 06 jan. 2025.

ANDRADE, C. L. T. **Seleção do sistema de irrigação**. Sete Lagoas: Embrapa Milho e Sorgo, 2001. 20 p. (Circular Técnica, 14). Disponível em: <https://www.infoteca.cnptia.embrapa.br/bitstream/doc/485302/1/Circ14.pdf>. Acesso em: 10 jul. 2024.

ANDRADE, I. P. S.; **Manejo da Irrigação na cultura da Figueira (Ficus carica L.) utilizando o balanço de água no solo**. 2013. Dissertação (Mestrado em Ciência do Solo) - Instituto de Agronomia Departamento de Solo, Universidade Federal Rural do Rio de Janeiro, p. 56. Rio de Janeiro, RJ, 2013.

ANDRADE, M. G. de.; VILAS BOAS, M. A.; SIQUEIRA, J. A. C.; SATO, M.; DIETER, J.; HERMES, E.; MERCANTE, E.; **Uniformity microsprinkler irrigation system using statistical quality control**. Ciência Rural, v. 47, n. 4, 2017.

ANDRADE, M. de M. **Inovações tecnológicas no agronegócio: impactos econômicos e jurídicos da inteligência artificial**. Revista FT, v. 29, n. 144, mar. 2025. Disponível em: <https://revistaft.com.br/inovacoes-tecnologicas-no-agronegocio-impactos-economicos-e-juridicos-da-inteligencia-artificial/>. Acesso em: 3 maio 2025.

ARAÚJO JÚNIOR, R. de A. **Controlador automático de irrigação, baseado na aplicação de água em turnos de rega e lâminas variáveis**. 2017. Tese de doutorado (Doutorado em ciência do solo) – Universidade Estadual Paulista (Unesp). Jaboticabal. 2017.

BALBINO, J. M. de S.; COSTA, H.; VENTURA, J. A.; FORNAZIER, M. J.; CASTRO, L. L. F. de; PREZOTTI, L. C.; SOUZA, J. L. de. **Cultura do pepino salada**. Vitória, ES: EMCAPA, p. 34, 1991.

BANZI, M.; SHILOH, M. **Primeiros Passos com o Arduino–2ª Edição: A plataforma de prototipagem eletrônica open source**. Novatec Editora, 2015.

BERNARDO, S. et al.; **Manual de irrigação**. Salassier Bernardo, Antonio Alves Soares, Everardo Chartuni Mantovani. 8. ed. Viçosa: Ed. UFV, 2006. 600 p.

BISCAIRO, G. A.; **Sistemas de irrigação por aspersão**. Editora da UFGD. Dourados - MS, p. 134, ISBN 978-85-61228-35-4. 2009.

BONFIM, F. de F.; TURIN, F. A. M.; BONFIM, R. F. **Irrigação automatizada para cultivo de morango**. 2023. Trabalho de Conclusão de Curso (Curso Técnico em Automação Industrial) – Escola Técnica Philadelpho Gouvêa Netto, São José do Rio Preto, 2023. Disponível em: <https://ric.cps.sp.gov.br/handle/123456789/14469>. Acesso em: 5 abril 2025.

CALZAVARA, B. B.; **História e importância da irrigação**. Embrapa Amazônia Oriental - Outras publicações científicas (ALICE), 1953.

CHAVIER, L. F. **Programação para arduino-primeiros passos**. Conceitos iniciais de programação para Arduino, 2013. Disponível em: <https://professor.luzerna.ifc.edu.br/marcelo-cendron/wp-content/uploads/sites/40/2017/03/Programa%C3%A7%C3%A3o-para-Arduino-Primeiros-Passos-Conceitos-iniciais-de-programa%C3%A7%C3%A3o-para-Arduino-Projeto-de-eletr%C3%B4nica-modular-com-Arduino-Circuitar.pdf>. Acesso em: 18 fev. 2025.

COELHO, E. F.; OR, D.; **Root distribution and water uptake patterns of corn under surface and subsurface drip irrigation**. Plant and Soil, v. 206, n. 02, p. 123 - 136, 1999.

COELHO, E. F.; FILHO, M. A. C.; OLIVEIRA, S. L.; **Agricultura irrigada: eficiência de irrigação e de uso de água**. Revista Bahia Agricultura, v. 7, n. 1, 2005.

COELHO, E. F.; SILVA, L. D. B. da; MAROUELLI, W. A.; COSTA, R. N. T. da; SOARES, J. M. **Distribuição de água no solo aplicado por gotejamento enterrado e superficial**. Revista Brasileira de Engenharia Agrícola e Ambiental, Campina Grande, v. 13, n. 6, p. 839–847, 2009. Disponível em: <https://www.scielo.br/j/rbeaa/a/qqS6qH3mTvLXWNzs8Vnq67p/>. Acesso em: 4 março 2025.

COSTA, R. N. T.; SOUZA, F. de.; **Irrigação por superfície**. Gestão Sustentável no Baixo Jaguaribe, Ceará. Fortaleza: Embrapa Agroindústria Tropical, p. 261-288, 2006.

CRUZ, J. P. H. **Crescimento e eficiência no uso da água de cafeeiro submetido a estratégias de manejo da irrigação**. 2019. Dissertação (Mestrado em Agronomia) - Universidade Federal de Uberlândia, Monte Carmelo, 2019. Disponível em: <https://repositorio.ufu.br/handle/123456789/26078>. Acesso em: 05 agost. 2024.

CUNHA, M. M.; **Desenvolvimento de um sistema embarcado para realização de manejo de irrigação**. 2019. Dissertação (Mestrado em engenharia agrícola) - Universidade Federal de Sergipe, São Cristóvão, SE, 2019. Disponível em URL: <<http://ri.ufs.br/jspui/handle/riufs/10937>>. Acesso em: 28 julho. 2024.

DE SIQUEIRA, A. S.; FERREIRA, A. F. **Sistema de irrigação automatizada para estufas de cultivo de tomate utilizando a plataforma arduino**. Revista Ibero-Americana de Humanidades, Ciências e Educação, v. 9, n. 2, p. 1121-1129, 2023.

DA PORCIÚNCULA, C. B.; BESKOW, S.; MARCON, D. S.; NOBRE, J. C. **Constrained Application Protocol (CoAP) no Arduino UNO R3: Uma Análise Prática**. In: **WORKSHOP PREPARATÓRIO DO IETF**, 2018, Porto Alegre. Anais. Porto Alegre: Sociedade Brasileira de Computação, 2018. Disponível em: <https://www.mendeley.com/catalogue/3387c056-c1d6-3f0b-be8d-03465ea7b888/>. Acesso em: 6 fev. 2025.

DRUMOND, Luís César Dias et al. **Irrigação de pastagens**. Agricultura irrigada no Brasil: ciência e tecnologia. Tradução. Piracicaba: ESALQ, 2022. p. 397. Disponível em: <<https://doi.org/10.11606/9786587391236>>. Acesso em: 24 ago. 2024.

EVANS, M.; NOBLE, J.; HOCHENBAUM, J.: **Arduino em ação**. Novatec Editora, 2013. Disponível em: < <https://books.google.com.br/books?hl=pt-BR&lr=&id=tig0CgAAQBAJ&oi=fnd&pg=PA5&dq=partes+do+arduino&ots=mTjthbZxXJ&sig=gzX6DlnhEjDVVzeETbotIE3XuSg#v=onepage&q=partes%20do%20arduino&f=false>>. Acesso em: 23 de janeiro de 2025.

FILGUEIRA, F. A. R.; **Novo manual de olericultura: agrotecnologia moderna na produção e comercialização de hortaliças**. Viçosa: UFV, 2008. 421p.

FINHOLDT, Gustavo et al. **Desenvolvimento de uma estação meteorológica automática para manejo de irrigação**. Revista Engenharia na Agricultura. v. 19, n. 4, p. 313-328, 2011.

FRIZZONE, J. A.; **Métodos de irrigação**. Departamento de engenharia de biosistemas USP. ESALQ / USP 2017.

GEA, G. F. M.; RIBEIRO, J. S. **Estufa modular com sistema automatizado de controle**. 2023. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Eletrônica Industrial) – Faculdade de Tecnologia de São Paulo, 50 f. São Paulo, 2023. Orientador: Prof. Dr. Mauricio Antônio Deffert. Disponível em: <https://ric.cps.sp.gov.br/handle/123456789/14536>. Acesso em: 4 jan. 2025.

GEISENHOFF, L. O.; OLIVEIRA, F. C.; BISCARO, G. A.; ALMEIDA, A. C. S.; SCHWERZ F.; **Produtividade do brócolis-de-cabeça sob diferentes sistemas de irrigação**. Engenharia Agrícola, Jaboticabal, v. 35, n. 5, p. 863 - 874, 2015.

GOMES, A.: **Controle automatizado de estufas com Arduino**. Revista engenharia Agrícola, 2019.

GOMES, E. P.; SOUSA, A. P.; **Produtividade da alface (*Lactuca sativa* L.) em função dos valores de lâminas de água aplicados por gotejamento superficial e subsuperficial.** Irriga, v. 7, n. 1, p. 35 - 41, 2002.

GOMES, H. P.; **Engenharia de Irrigação: Hidráulica dos Sistemas Pressurizados, Aspersão e Gotejamento.** ed. 3, p. 412, UFPB, Universidade Federal da Paraíba. Campina Grande, PB, 1999.

HAMADA, E.; **Desenvolvimento e produtividade da alface (*Lactuca sativa* L) submetida a diferentes lâminas de água através da irrigação por gotejamento.** 1993. Dissertação (Mestrado em água e solo) - Universidade Estadual de Campinas, Campinas, 103 f. 1993.

JÚNIOR, S. L. S.; SILVA, R. A. **Automação e Instrumentação Industrial com Arduino.** 1. ed. São Paulo: Editora Saraiva, 2017.

KELLER, J.; KARMELI, D.; **Trickle irrigation design.** Glendora: Rain Bird Sprinkler Manufacturing Corporation, S. 1, p. 133, 1975.

LIMA, J. E. F. W.; FERREIRA, R. S.; Alves; C.; **O uso da irrigação no Brasil.** In: O estado das águas no Brasil. Agência Nacional de Energia Elétrica. 1999.

MANFRINATO, H. A.; **A irrigação por gotejamento: influência sobre as relações solo - água.** Irrigação e Tecnologia Moderna, n. 22, p. 21 - 25, 1985.

MANTOVANI, E. C.; BERNARDO, S.; PALARETTI, L. F.; **Irrigação: princípios e métodos.** Editora UFV2, Ed. 2, p. 358. Viçosa, MG, 2007.

MARQUELLI, W. A.; SILVA, H. R.; SILVA, W. L. C.; **Irrigação por aspersão em hortaliças: qualidade da água, aspectos do sistema e método prático de manejo.** ed. 4, revista atual. e ampl. – Brasília, DF: Embrapa, 2017.

MARQUELLI, W. A.; SILVA, H. R.; **Seleção de sistemas de irrigação para hortaliças**. Circular técnica 11, ISSN 1415 - 3033. EMBRAPA, Samambaia Norte, Brasília, DF, 1998.

MARGOLIS, M. **Arduino Cookbook**. O'Reilly, 2011.

MICROBERTS, Michael. **Arduino Básico**. 1. ed. São Paulo: Novatec, 2011.

MEDEIROS, R. D.; **Efeitos do manejo de água e de sistemas de controle de plantas daninhas em arroz (*Oryza sativa* L.) irrigado**. 1995. Dissertação (Mestrado em Agronomia) - Escola Superior de Agricultura "Luiz de Queiroz", Universidade de São Paulo, p. 80, São Paulo, 1995.

MORAES, MARIA J. et al. **Automação em sistemas de irrigação tipo pivô central para economia de energia elétrica**. Engenharia Agrícola, Jaboticabal, v. 34, n. 6, p. 1075 – 1088, nov./dez. 2014. Disponível em: <https://www.scielo.br/j/eagri/a/Q9ymfq9QX9n5cscBcTXsshx/?format=pdf&lang=pt>. Acesso em: 12 jan. 2025.

MUSCI, Marcelo.; ANTOLIN, Gisele Duarte Caboclo.; VISTA, Renata Couto.; ANTOLIN, Mauricio Quelhas.; **Introdução ao Arduino**. Revista Contemporânea, v. 3, n. 10, ISSN 2447-0961. 2023.

PAZ, V. P. S.; TEODORO, R. E. F.; MENDONÇA, F. C.; **Recursos hídricos, agricultura irrigada e meio ambiente**. Revista brasileira de engenharia Agrícola e Ambiental. Campina Grande, PB – universidade v. 4, n. 3, p. 465 - 473, 2000.

PEREIRA, F. A. L.; et al. **Automação de precisão utilizando arduino e inversor de frequência aplicado a sistemas de irrigação por válvulas**. IRRIGA, v. 25, n. 1, p. 27–37, 2020.

PEREIRA, M. J.; FILHO, J. S.; CRUZ, J. M. G.; RIBEIRO, J. R.; CARVALHO, A. A. P.; **Proposta de desenvolvimento de sistema de automação de irrigação por microaspersão utilizando hardware livre**. Escola potiguar de computação e suas

aplicações. Angicos. Porto Alegre: Sociedade Brasileira de Computação, p. 84-93, 2018.

PINHEIRO, A. A.; **Manejo da irrigação utilizando umidade do solo e dados climáticos em lavoura de cafeeiro conilon**. 2019. Dissertação (Mestrado engenharia agrícola) - Universidade Federal do Espírito Santo, ALEGRE, ES. 2019. Disponível em URL: < <http://repositorio.ufes.br/handle/10/10964>> Acesso em: 30 jul. 2024.

REIS, J. S.; **Sistema de controle aplicado à automação de irrigação agrícola**. Trabalho de conclusão de curso (graduação em engenharia de controle e automação) – Universidade Tecnológica Federal do Paraná, Curitiba, 2015.

RIBEIRO, R. B.; FILGUEIRAS, R.; RAMOS, M. C. A.; ALMEIDA, L.; GENEROSO, T. M. et al. **Variabilidade espaço-temporal da condição da vegetação na agricultura irrigada por meio de imagens sentinel**. Revista Brasileira de Agricultura Irrigada. v.11, nº.6, p. 1884 – 1893, Fortaleza, CE. 2017. Disponível em: <https://doi.org/10.7127/rbai.v11n600648>. Acesso em: 05 jul. 2024.

ROUX, G.; **Ancient Iraq**. 3. ed. London: Penguin Books, 1992. Disponível em: https://books.google.com.br/books?hl=pt-BR&lr=&id=klZX8B_RzzYC&oi=fnd&pg=PT5&dq=Ancient+Iraq&ots=K7Nm2HjphM&sig=x2y-DQuvi-rBCTf7HgH7a2UTLI4. Acesso em: 17 jan. 2025.

SANTOS, P. V. S; ARAÚJO, M. A. de. **A importância da inovação aplicada ao agronegócio: uma revisão**. Revista Latino-Americana de inovação e engenharia de produção, v. 5, n. 7, p. 31-47, 2017.

SCALOPPI, E. J. Paolinelli, A.; Dourado Neto, D.; Mantovani, E.C.; **Irrigação de baixo custo por sulcos**. Cap. 10. (Organizadores). Diferentes abordagens da agricultura irrigada no Brasil, Livro 2 – Técnica e cultura, Editora USP, p. 175 - 194, 2021.

SILVA, A. O. et al. **Irrigation in the age of agriculture 4.0: management, monitoring and precision**. Revista Ciência Agronômica, Fortaleza, v. 51, p. 1-17, 2020.

SILVA, J. L. de S. et al. **Plataforma Arduino integrado ao PLX-DAQ: Análise e aprimoramento de sensores com ênfase no LM35**. In: ESCOLA REGIONAL DE COMPUTAÇÃO BAHIA, ALAGOAS E SERGIPE (ERBASE), 14. Feira de Santana. Anais. Feira de Santana: ERBASE, BA, 2014.

SILVA, L. et al.: **Monitoramento climático com Arduino e IoT**. In: CONGRESSO BRASILEIRO DE AGRICULTURA DE PRECISÃO, 2005. Anais, 2005.

SILVA, A. R.; OLIVEIRA, D. S.; COSTA, J. R.; **Impacto do manejo da irrigação na produtividade da alface**. Irrigação e Tecnologia, v. 23, n. 1, p. 45 - 52, 2017.

SOUSA, V. F. de; MAROUELLI, W. A.; COELHO, E. F.; PINTO, J. M.; COELHO FILHO, M. A.; **Manejo da água de irrigação**. In: Irrigação e fertirrigação em fruteiras e hortaliças, ed. 1, cap. 5, p. 158 - 232. Brasília, DF, Embrapa Informação Tecnológica, 2011.

SOUZA, C. F.; MATSURA, E. E.; **Distribuição da água no solo para o dimensionamento da irrigação por gotejamento**. Revista Brasileira de Engenharia Agrícola e Ambiental, Campina Grande, v. 8, n. 1, p. 7 - 15, 2004.

SOUZA, C. F.; CONCHESQUI, M. E. S.; SILVA, M. B. **Semiautomatic irrigation management in tomato**. Engenharia Agrícola, v. 39, n. special issue, p. 118-125, 2019.

TESTEZLAF, R.; **Irrigação: métodos, sistemas e aplicações**. Faculdade de engenharia agrícola, UNICAMP. Campinas, SP, 2017. Disponível em: https://www2.feis.unesp.br/irrigacao/pdf/testezlaf_irrigacao_metodos_sistemas_aplicacoes_2017.pdf. Acesso em: 28 fev. 2025.