

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO



PROJETO DE IRRIGAÇÃO SEMIAUTOMÁTICO

HAKAWÃ LUIZ BERNARDI

GOIÂNIA
2025

HAKAWÃ LUIZ BERNARDI

PROJETO DE IRRIGAÇÃO SEMIAUTOMÁTICO

Trabalho de Conclusão de Curso apresentado à Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para obtenção do título de Bacharelado em Engenharia de Computação.

Orientador (a):

Prof. Me. André Luiz Alves.

Banca examinadora:

Prof. Me. Daniel Corrêa da Silva.

Prof.^a Dra. Martha Nascimento Castro.

GOIÂNIA

2025

HAKAWÃ LUIZ BERNARDI

PROJETO DE IRRIGAÇÃO SEMIAUTOMÁTICO

Trabalho de Conclusão de Curso aprovado em sua forma final pela Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, para obtenção do título de Bacharelado em Engenharia de Computação, em 02 / 12 / 2025.

Orientador (a): Prof. Me. André Luiz Alves

Prof. Me. Daniel Corrêa da Silva

Prof.^a Dra. Martha Nascimento Castro

GOIÂNIA

2025

RESUMO

Diante do cenário atual de crescente escassez hídrica e da necessidade de adotar práticas mais sustentáveis, este trabalho apresenta o desenvolvimento de um sistema de irrigação semiautomática utilizando a plataforma Arduino UNO, visando a otimização do uso da água e a praticidade no cuidado com plantas. A proposta foi estruturada com base nos princípios de Sistemas Embarcados e na aplicação do modelo em cascata, que guiou todas as etapas do projeto, desde o levantamento de requisitos até a implementação e testes. O sistema integra sensores de umidade do solo e de nível de água no reservatório, que monitoram em tempo real as condições necessárias para o acionamento automático da irrigação, reduzindo desperdícios e promovendo um uso mais consciente da água. O projeto também oferece a opção de irrigação manual, aumentando sua flexibilidade e funcionalidade. Com isso, busca-se fornecer uma solução acessível, eficiente e sustentável, aplicável tanto a hortas domésticas quanto a pequenos cultivos urbanos.

PALAVRAS-CHAVE: Irrigação Semiautomática. Automação. Sistemas Embarcados. Arduino. Sustentabilidade. Otimização Hídrica.

LISTA DE ABREVIATURAS E SIGLAS

CPU	<i>Central Processing Unit</i> (Unidade Central de Processamento)
GPU	<i>Graphics Processing Unit</i> (Unidade de Processamento Gráfico)
IoT	<i>Internet of Things</i> (Internet das Coisas)
I2C	<i>Inter-Integrated Circuit</i> (Circuito Inter-Integrado)
LCD	<i>Liquid Crystal Display</i> (Visor de Cristal Líquido)
LED	<i>Light Emitting Diode</i> (Diodo Emissor de Luz)
RF	Requisito Funcional
RNF	Requisito Não Funcional

LISTA DE FIGURAS

Figura 1 – Modelo Cascata.....	13
Figura 2 – Irrigação por Gotejamento.....	17
Figura 3 – Irrigação por Microaspersão.....	18
Figura 4 – Irrigação Subterrânea.....	18
Figura 5 – Diagrama de Contexto.....	23
Figura 6 – Componentes do Projeto.....	25
Figura 7 – Estrutura Inicial.....	27
Figura 8 – Estrutura <i>Setup()</i>	28
Figura 9 – Inicialização do Sistema.....	29
Figura 10 – Função <i>medirDistancia()</i>	29
Figura 11 – Função <i>exibirLCD()</i>	30
Figura 12 – Mensagem de Aviso.....	31
Figura 13 – Função <i>controlarBomba()</i>	32
Figura 14 – Bomba de Irrigação.....	32
Figura 15 – Função <i>loop()</i>	33
Figura 16 – Pinagem da Bomba D'água.....	34
Figura 17 – Pinagem do LCD.....	34
Figura 18 – Pinagem Sensor de Distância.....	35
Figura 19 – Vista Panorâmica.....	35

LISTA DE QUADROS

QUADRO 1 – Requisitos Funcionais.....	21
QUADRO 2 – Requisitos não Funcionais.....	22

SUMÁRIO

1. INTRODUÇÃO	9
2. SISTEMAS EMBARCADOS	12
2.1. Metodologia de Desenvolvimento Adotada	12
2.2. Vantagens do Modelo Cascata	15
3. ESTUDO DE CASO.....	16
3.1. Cenário do Problema	16
3.2. Levantamento de Soluções Existentes	17
4. Proposta de Solução	20
4.1. Requisitos.....	20
4.1.1. Requisitos Funcionais	21
4.1.2. Requisitos não Funcionais	22
4.2. Projeto de Solução.....	23
4.2.1 Diagrama de Contexto	23
4.2.2 Componentes Utilizados	24
4.3. Construção	25
4.3.1. Codificação.....	25
4.3.2. Testes.....	26
4.3.3. Desenvolvimento do Código.....	27
4.3.4. Estrutura Inicial.....	27
4.3.5. Estrutura Setup.....	28
4.3.6. Função Medir Distância	29
4.3.7. Exibição no LCD e Alerta de Nível de Água	30
4.3.8. Controle da Bomba de Irrigação.....	32
4.3.9. Irrigação Manual.....	33
5. Considerações finais.....	36
5.1 Dificuldades Encontradas	36

1. INTRODUÇÃO

A crescente preocupação com a escassez de recursos hídricos e a necessidade de otimizar o uso da água na agricultura têm impulsionado a adoção de sistemas de irrigação mais eficientes. Entre essas tecnologias, destaca-se a irrigação semiautomática, que alia a eficiência no consumo de água à praticidade no manejo das plantas.

A irrigação semiautomática permite um manejo prático ao utilizar sensores de umidade do solo, que monitoram a presença de água e acionam a irrigação conforme a necessidade. Dessa forma, evita-se tanto o desperdício de água quanto a subirrigação, garantindo um ambiente adequado para o desenvolvimento das plantas (Silva *et al.*, 2021). Além disso, essa tecnologia reduz a necessidade de intervenção manual, tornando-se uma solução viável para quem deseja manter plantas saudáveis sem preocupações constantes com a rega (Santos & Oliveira, 2020).

Pesquisas apontam que o uso de sensores de umidade pode reduzir significativamente o consumo de água na irrigação, uma vez que o sistema só é acionado quando o solo atinge um nível crítico de secura (Almeida & Souza, 2019). Esse método permite uma distribuição mais inteligente dos recursos hídricos, beneficiando tanto pequenos produtores quanto pessoas que cultivam plantas em ambientes domésticos.

Diante deste contexto, este projeto visa responder a seguinte questão de pesquisa: **Seria possível criar e desenvolver um projeto semiautomático visando a redução do desperdício de água?**

O objetivo geral deste trabalho é:

- Desenvolver um projeto de irrigação inteligente visando reduzir o consumo de água, promovendo um uso mais eficiente dos recursos hídricos.

Para alcançar esse propósito, busca-se:

- Desenvolver um código que ofereça suporte ao sistema, permitindo que a irrigação seja ativada automaticamente sempre que for detectada a ausência de umidade no solo;
- Identificar possíveis insuficiências e desafios que possam surgir no uso dessa tecnologia no dia a dia;
- Projetar, construir, testar e validar um protótipo funcional que comprove a eficácia da proposta.

Este trabalho se justifica pela necessidade de soluções tecnológicas que possam auxiliar na redução de escassez hídrica. A aplicação de Sistemas Embarcados no desenvolvimento do projeto visa não apenas fornecer uma resposta a esse desafio, mas também contribuir para a evolução da área.

O capítulo 2 abordará os conceitos fundamentais sobre sistemas embarcados, destacando sua importância no desenvolvimento de soluções tecnológicas aplicadas à automação, como é o caso deste projeto de irrigação semiautomática. Este capítulo discute como a integração entre hardware e software permite criar sistemas específicos, eficientes e de baixo consumo energético, essenciais para aplicações sustentáveis.

Dando continuidade, o capítulo 3 apresenta o estudo de caso que deu origem ao desenvolvimento do sistema de irrigação semiautomática proposto. Neste capítulo, será detalhado o cenário do problema e as motivações que levaram à criação da solução, destacando os desafios relacionados ao desperdício de água e à dificuldade de manutenção de plantas no dia a dia. Além disso, são analisadas algumas soluções existentes no mercado, ressaltando suas vantagens e limitações. Por fim, o capítulo detalha como a aplicação da automação, por meio do uso do Arduino e sensores de umidade, pode oferecer uma alternativa eficiente, econômica e de fácil implementação para pequenos cultivos e jardins domésticos.

Diante disso, o capítulo 4 descreve uma proposta que tem como objetivo principal oferecer uma solução eficiente, acessível e sustentável, capaz de monitorar a umidade do solo e controlar a irrigação de maneira automatizada, evitando desperdícios de água e reduzindo a necessidade de intervenção humana.

O capítulo 5 apresentou as considerações finais do projeto de irrigação semiautomático com Arduino UNO, destacando a viabilidade de sua aplicação em

pequenos cultivos e ambientes domésticos. O sistema desenvolvido demonstrou eficiência na redução do desperdício de água, acionando a irrigação apenas quando necessário, além de oferecer flexibilidade por meio do acionamento manual. Também foram ressaltadas as contribuições educacionais e sociais do projeto, bem como possibilidades futuras de expansão, como a integração com IoT e o uso de energia solar. Por fim, foram discutidas as principais dificuldades enfrentadas durante o desenvolvimento, como problemas de mau contato nos componentes e a calibração do sensor de umidade, que exigiram testes práticos e ajustes constantes para garantir a confiabilidade do sistema.

Nas Referências, são apresentadas todas as fontes de pesquisa que serviram de embasamento teórico, técnico e metodológico para o desenvolvimento deste trabalho. Esse conjunto de materiais garante a credibilidade e a consistência do trabalho realizado.

2. SISTEMAS EMBARCADOS

A área de Sistemas Embarcados se dedica ao desenvolvimento de soluções computacionais integradas a hardware, visando controle, automação e otimização de processos. Ela abrange atividades como análise de requisitos, projeto, implementação e testes. Conforme salientado por Wolf (2019), os sistemas embarcados são projetados para desempenhar funções específicas dentro de restrições de tempo real, consumo de energia e processamento eficiente.

Segundo Vahid & Givargis (2011), um sistema embarcado é composto por hardware e software projetados para uma função específica, operando de maneira autônoma ou interagindo com outros sistemas. Esses sistemas são essenciais em diversas aplicações, incluindo automação industrial, dispositivos médicos e sistemas de controle, como ocorre na automação da irrigação.

A complexidade dos sistemas embarcados, conforme destacado por Wolf (2019), resulta da integração entre software e hardware, exigindo uma abordagem multidisciplinar para garantir desempenho e eficiência energética. Esses sistemas são fundamentais para o desenvolvimento de tecnologias inteligentes e sustentáveis.

Um sistema embarcado personalizado, de acordo com Vahid e Givargis (2011, p. 8), refere-se a "soluções desenvolvidas especificamente para um determinado propósito, otimizadas em termos de hardware e software para atender às necessidades de um projeto específico".

No contexto deste trabalho, o sistema de irrigação inteligente se encaixa nessa definição, pois foi projetado para monitorar a umidade do solo e controlar a irrigação de maneira eficiente, reduzindo desperdícios e otimizando o uso da água.

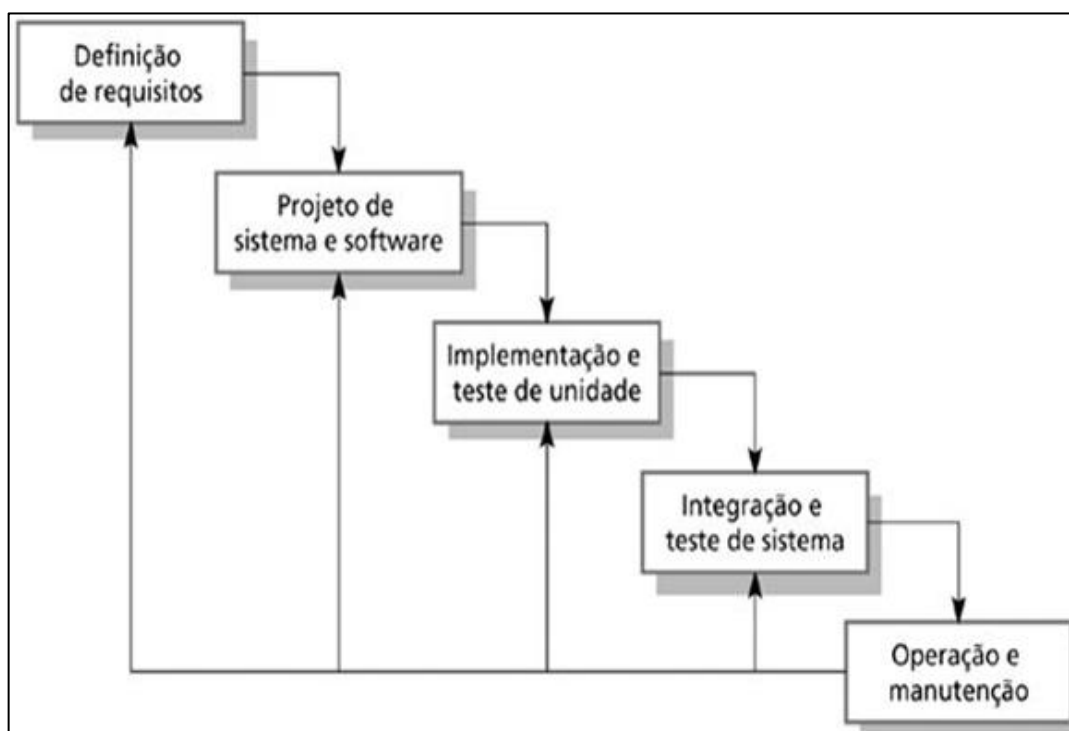
2.1. Metodologia de Desenvolvimento Adotada

O desenvolvimento do projeto de irrigação semiautomática é orientado pelo modelo em cascata, um dos métodos mais tradicionais no desenvolvimento de sistemas. Como explica Sommerville (2019), este modelo segue um fluxo linear e sequencial, onde cada fase deve ser concluída antes do início da próxima. Esse

modelo é adequado para projetos bem estruturados, com requisitos definidos desde o início, tornando-se ideal para a implementação de um sistema embarcado.

A figura 1 ilustra o modelo em cascata.

Figura 1 – Modelo Cascata



Fonte: Adaptado pelo autor com base em Sommerville, 2011.

O modelo é dividido em cinco etapas principais: levantamento de requisitos, projeto, implementação, verificação e manutenção. Cada uma dessas fases foi aplicada diretamente ao desenvolvimento do sistema de irrigação inteligente, conforme detalhado a seguir.

Na etapa de levantamento de requisitos, foram definidos os objetivos e as funcionalidades essenciais do sistema. O principal requisito identificado foi criar uma forma de irrigação que economizasse água e reduzisse a necessidade de intervenção humana.

Durante essa fase, determinou-se a necessidade de:

- Utilizar um sensor de umidade do solo para detectar quando o solo está seco e acionar a irrigação automaticamente;
- Empregar um sensor de distância para medir o nível de água no reservatório;
- Exibir informações e alertas em um display LCD;

- Integrar um botão de acionamento manual;
- Controlar a bomba d'água por meio de um módulo relé.

Segundo Pressman & Maxim (2016), a clareza na definição dos requisitos é essencial para evitar retrabalho e falhas em fases posteriores, além de servir como guia para todo o desenvolvimento do sistema.

Com os requisitos definidos, iniciou-se a fase de projeto, na qual foram desenvolvidos a organização lógica do código e a definição da interação entre os componentes.

O projeto elétrico foi elaborado de modo que cada módulo se comunicasse de forma eficiente com o Arduino UNO. Além disso, foram considerados aspectos como alimentação elétrica segura, compatibilidade de tensão e disposição física dos componentes na protoboard, de modo a minimizar ruídos e falhas de conexão.

No âmbito do software, foi planejada a lógica de controle: a leitura dos sensores, o acionamento do relé, a exibição no LCD e o comportamento do LED indicador. Essa fase garantiu que todas as partes do sistema estivessem integradas e prontas para a codificação.

Na etapa de implementação, o sistema foi construído fisicamente e o código foi desenvolvido no ambiente Arduino IDE.

As bibliotecas necessárias foram importadas e os pinos foram configurados. Em seguida, o código foi implementado por partes, durante essa fase, ajustes foram realizados para calibrar os sensores, garantindo que as leituras fossem confiáveis. O sensor de umidade foi testado em diferentes tipos de solo para identificar o valor ideal de referência para acionar a irrigação.

Segundo Sommerville (2019), a fase de implementação deve seguir rigorosamente o planejamento definido, assegurando que cada módulo funcione conforme o esperado e que as etapas anteriores sejam fielmente traduzidas em código e hardware.

A etapa de verificação teve como objetivo confirmar se o sistema funcionava conforme os requisitos estabelecidos. Foram realizados testes unitários e testes integrados, avaliando o funcionamento conjunto de todos os componentes.

Durante os testes, observou-se que o sistema era capaz de ligar e desligar a bomba automaticamente conforme a umidade do solo variava. O LCD exibiu corretamente as mensagens de alerta, enquanto o LED indicador funcionou como sinal visual de aviso em caso de baixo nível no reservatório ou falhas de leitura.

Após a validação funcional, o sistema entrou na fase de manutenção, que envolveu observações do desempenho em uso real e possíveis melhorias futuras.

Durante essa fase, identificou-se a necessidade de revisar o posicionamento dos sensores para aumentar a precisão das medições. Também foi discutida a possibilidade de expansão do projeto para incluir recursos de conectividade IoT, permitindo o monitoramento remoto dos níveis de umidade e água através de um aplicativo ou interface web.

Segundo Pressman & Maxim (2016), a manutenção é uma etapa contínua que assegura a evolução do sistema e a incorporação de novas tecnologias conforme as necessidades se ampliam. Dessa forma, o projeto permanece atualizável e adaptável a contextos mais complexos, como pequenas propriedades agrícolas ou estufas automatizadas.

2.2. Vantagens do Modelo Cascata

De acordo com Sommerville (2019) o modelo cascata oferece uma série de vantagens, incluindo:

- **Organização e Estruturação:** Como cada fase deve ser concluída antes da próxima, o desenvolvimento ocorre de forma ordenada;
- **Facilidade na Documentação:** Todas as etapas são bem registradas, facilitando futuras manutenções e melhorias no sistema;
- **Maior Controle do Projeto:** O cronograma e os custos podem ser mais bem planejados.

3. ESTUDO DE CASO

A irrigação semiautomática tem como principal objetivo a otimização do consumo de água. Diante do cenário atual de escassez hídrica e da crescente necessidade por soluções sustentáveis, o presente estudo de caso propõe o desenvolvimento de um sistema de irrigação semiautomática utilizando Arduino UNO, com foco no consumo consciente de água e na praticidade no cuidado com plantas, tanto em ambientes residenciais quanto em pequenos cultivos urbanos.

A escolha por um sistema semiautomático se baseia na necessidade de equilibrar automação e simplicidade, oferecendo ao usuário um controle eficiente sobre a irrigação sem depender exclusivamente de intervenções manuais ou de sistemas complexos. Este modelo se mostra especialmente vantajoso para pessoas que possuem hortas domésticas, jardins ou plantas ornamentais, proporcionando autonomia no manejo hídrico e contribuindo para práticas mais sustentáveis.

3.1. Cenário do Problema

O uso indiscriminado de água em práticas simples, como a irrigação manual, frequentemente gera desperdícios consideráveis. Muitas vezes, usuários irrigam plantas sem considerar o real nível de umidade do solo, o que pode resultar tanto no desperdício de água quanto na deterioração da própria planta, que sofre com excesso ou escassez de água.

Somado a isso, a rotina cada vez mais corrida das pessoas dificulta a manutenção constante das plantas. Faltas na irrigação podem prejudicar a saúde das plantas, principalmente em períodos de estiagem ou quando o usuário se ausenta por alguns dias.

Diante desse contexto, surge a necessidade de um sistema que, de forma simples e econômica, seja capaz de identificar quando a planta realmente necessita de água, acionando a irrigação automaticamente e informando ao usuário sobre o status do sistema.

3.2. Levantamento de Soluções Existentes

Existem outras técnicas de irrigação que também buscam reduzir o consumo de água, tais como:

- Irrigação por gotejamento: Consiste na aplicação lenta e localizada da água diretamente na raiz da planta. Este sistema pode reduzir a perda de água por evaporação e garantir um uso mais eficiente dos recursos hídricos (Ferreira, 2020). A Figura 2 ilustra uma irrigação por gotejamento.

Figura 2 – Irrigação por Gotejamento



Fonte: Penacchi (2022)

- Irrigação por microaspersão: Utiliza emissores que liberam pequenos jatos de água próximos às plantas. Essa técnica melhora a eficiência no uso da água e minimiza o escoamento superficial (Silva, 2018). A figura 3 ilustra uma irrigação por microaspersão.

Figura 3 – Irrigação por Microaspersão



Fonte: Fillipi (2020)

- Irrigação subterrânea: Distribui a água diretamente no sistema radicular das plantas, evitando perdas por evaporação e otimizando o consumo hídrico (Gomes, 2017). A figura 4 ilustra uma irrigação subterrânea.

Figura 4 – Irrigação Subterrânea



Fonte: Cremasco (2021)

Apesar da eficiência desses métodos, eles geralmente dependem de sistemas maiores, com custos mais elevados, ou de projetos mais complexos, muitas vezes inviáveis para pequenos espaços ou usuários leigos em automação.

Nesse contexto, o sistema de irrigação com Arduino surge como uma alternativa intermediária, que combina automação, baixo custo e simplicidade operacional. Por meio da integração entre sensores, atuadores e monitoramento em tempo real, o sistema garante que a irrigação ocorra apenas quando necessária, reduzindo o desperdício de água e o esforço humano.

Além disso, o estudo evidencia o potencial de expansão do projeto para fins educacionais e de pesquisa, já que sua simplicidade de montagem e programação permite a aplicação em escolas técnicas e universidades. Dessa forma, o sistema contribui não apenas para o uso racional da água, mas também para a formação de novos profissionais capacitados na área de automação e sustentabilidade.

4. PROPOSTA DE SOLUÇÃO

O desenvolvimento de um sistema de irrigação inteligente com Arduino UNO visa otimizar o uso da água e facilitar o cuidado com as plantas de forma eficiente e acessível. De acordo com Pressman (2016), a automação de processos permite maior controle e precisão, reduzindo desperdícios e aumentando a confiabilidade do sistema.

Sob essa perspectiva, o projeto propõe um sistema que monitora a umidade do solo por meio de sensores e exibe informações no display LCD, indicando se há necessidade de abastecer o reservatório. A integração desses componentes possibilita um controle eficiente da irrigação, garantindo que a planta receba água somente quando necessário.

A arquitetura do sistema considera a escolha de componentes adequados, como sensores e atuadores, garantindo compatibilidade e eficiência energética. Além disso, técnicas de controle e automação são aplicadas para permitir o funcionamento de forma otimizada, evitando desperdícios e garantindo a distribuição adequada da água.

4.1. Requisitos

Os requisitos são as características ou funcionalidades que um sistema de controle e automação deve atender para garantir um funcionamento eficiente e autônomo. Eles são fundamentais para o sucesso de qualquer projeto, pois estabelecem uma base sólida para o desenvolvimento e implementação do sistema. Sommerville (2011).

No contexto deste trabalho, os requisitos são essenciais para garantir que o sistema de irrigação semiautomático funcione corretamente, atenda às necessidades dos usuários e evite desperdício de água e energia. Além disso, os requisitos promovem:

- a) Comunicação: Definem com clareza as funcionalidades do sistema, permitindo que todos os envolvidos compreendam seu funcionamento, como a ativação da irrigação com base na umidade do solo e a exibição de informações no LCD;
- b) Planejamento: Fornecem informações essenciais para o desenvolvimento do sistema, incluindo o escopo, os sensores e atuadores necessários, além do controle do nível de água no reservatório;
- c) Desenvolvimento: Direcionam a implementação do sistema, garantindo que o Arduino UNO atue corretamente na leitura da umidade do solo, no acionamento da bomba de irrigação e na exibição do nível de água no reservatório via LCD;
- d) Testes: Servem de base para a verificação e validação do sistema, permitindo avaliar se o controle da irrigação está sendo realizado corretamente e se as informações exibidas no LCD refletem com precisão o nível de água e a necessidade de reabastecimento.

4.1.1. Requisitos Funcionais

Essa seção especifica os funcionais essenciais do projeto de irrigação, visando um melhor entendimento.

Conforme destacado por Sommerville (2019), é crucial pontuar que os requisitos funcionais não se limitam apenas a descrever o que o projeto deve realizar, mas também podem explicar o que o projeto não deve executar. Em essência, requisitos funcionais definem as capacidades e funcionalidades específicas que o projeto deve oferecer.

No Quadro 1 há a ilustração de Requisitos funcionais.

QUADRO 1 – Requisitos Funcionais.

Identificação	Descrição
RF001	O sistema deve medir a umidade do solo utilizando um sensor adequado.
RF002	O sistema deve ativar a bomba de água quando a umidade do solo estiver abaixo de um limite pré-definido.

RF003	O sistema deve desativar a bomba de água quando a umidade atingir um nível adequado.
RF004	O sistema deve exibir no LCD o nível de água no reservatório
RF005	O sistema deve alertar no LCD quando o nível de água reservatório estiver baixo e precisar ser reabastecido.

Fonte: Autor (2025)

4.1.2. Requisitos não Funcionais

Esta seção destaca os requisitos não funcionais. Estes requisitos sob a perspectiva de Sommerville (2019), referem-se a critérios que não estão diretamente ligados às funcionalidades específicas do sistema, mas têm impacto significativo em sua qualidade e desempenho.

No Quadro 2 há a ilustração dos Requisitos não funcionais.

QUADRO 2 – Requisitos não Funcionais.

Identificação	Descrição
RNF001	O sistema deve utilizar um Arduino UNO como microcontrolador principal.
RNF002	O sensor de umidade deve apresentar precisão mínima de $\pm 5\%$ para leituras confiáveis.
RNF003	A bomba de água deve ser ativada por um relé, garantindo segurança elétrica.
RNF004	O LCD deve ser legível em ambientes externos e de baixa luminosidade

Fonte: Autor (2025)

4.2. Projeto de Solução

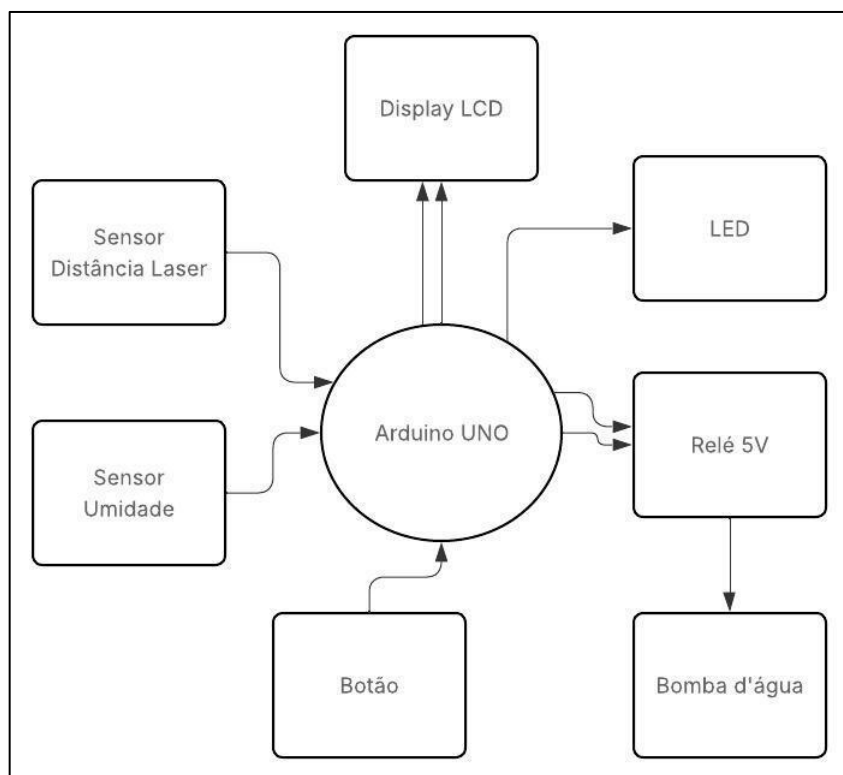
O projeto de solução proposto consiste no desenvolvimento de um sistema de irrigação semiautomático baseado no microcontrolador Arduino UNO, capaz de monitorar variáveis ambientais essenciais e acionar o processo de irrigação de maneira inteligente. A ideia central é fornecer um método eficiente e de baixo custo para o controle hídrico, utilizando sensores, atuadores e um módulo de exibição em LCD para fornecer feedback em tempo real ao usuário.

4.2.1 Diagrama de Contexto

O diagrama de contexto representa, de forma simplificada, a interação entre o sistema de irrigação semiautomático e os elementos externos que influenciam seu funcionamento. Seu objetivo é apresentar uma visão geral do fluxo de informações e dos principais componentes envolvidos no processo.

A figura 5 ilustra o diagrama de contexto utilizado no projeto.

Figura 5 – Diagrama de Contexto



Fonte: Autor (2025)

No centro do diagrama está o Arduino UNO, responsável por receber os dados dos sensores e controlar os atuadores. O sensor de umidade envia informações sobre o estado hídrico do solo, enquanto o sensor de distância laser monitora o nível de água no reservatório. A partir dessas informações, o microcontrolador decide se deve acionar ou interromper a bomba de irrigação, utilizando um módulo relé como interface de potência.

O usuário interage com o sistema por meio de um botão, que permite o acionamento manual da irrigação, e recebe feedback em tempo real através do display LCD, que exibe mensagens sobre umidade, nível de água e possíveis erros. Além disso, um LED integrado fornece alertas visuais importantes, como a necessidade de reabastecimento do reservatório.

Esse diagrama evidencia o fluxo de dados essencial para o funcionamento da solução, mostrando como cada componente contribui para a tomada de decisão do sistema e para a experiência do usuário. Ele também reforça a natureza centralizada do Arduino UNO como unidade de controle, garantindo simplicidade e clareza no entendimento da arquitetura geral.

4.2.2 Componentes Utilizados

Cada componente utilizado neste projeto foi essencial para cumprir o propósito de irrigar uma planta sem a interferência humana e visando o consumo consciente de água. Os componentes são:

- Arduino UNO R3;
- Módulo sensor de umidade de solo;
- Minibomba de água 3V – 6V;
- Módulo relé 5V;
- Sensor de distância laser VL53L0X;
- Display LCD 16x2 com I2C;
- Protoboard 400 furos;
- LED 5MM vermelho difuso;
- Jumpers.

O ambiente utilizado para a criação do código será de uma máquina:

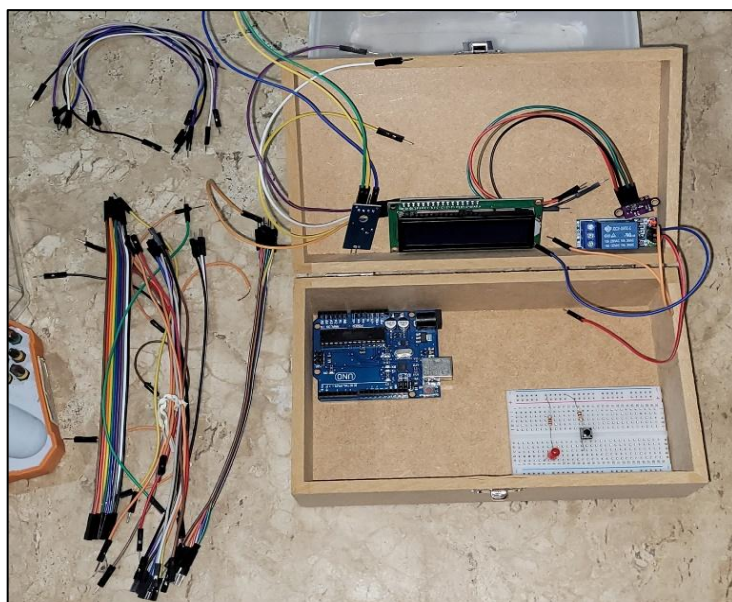
- **Laptop:** **CPU:** Intel Core I5 10th geração, **RAM:** 8gb a 2400mhz, **Placa Mãe:** *unknown* e **GPU:** NVIDIA GeForce MX350. **Sistema Operacional:** Windows 11 versão 23H2, utilizará do software Arduino IDE 2.3.4.

4.3. Construção

A fase de construção na engenharia representa o momento em que o projeto é transformado em um sistema funcional. Esse estágio abrange a implementação, testes e integração dos componentes, garantindo que a aplicação atenda aos requisitos definidos previamente. A eficiência e a qualidade dessa etapa são fundamentais para o êxito do projeto como um todo Pressman (2016).

A figura 6 ilustra os componentes do projeto.

Figura 6 – Componentes do Projeto



Fonte: Autor (2025)

4.3.1. Codificação

A codificação é a etapa em que o código fonte do software é escrito e descarregado no Arduino via cabo USB Tipo A para Tipo B. Em conforme com o Pressman (2011) essa etapa é geralmente realizada por desenvolvedores de software, que são responsáveis por implementar as funcionalidades definidas no projeto.

Na codificação, é importante seguir as boas práticas de desenvolvimento de software, como o uso de padrões de projeto, a adoção de metodologias ágeis e a realização de testes unitários. Sommerville (2019).

4.3.2. Testes

Os testes são essenciais para garantir a qualidade do software. Eles são realizados para verificar se cada parte do código funciona corretamente antes da integração completa do sistema. Esses testes garantem a detecção precoce de erros, reduzindo retrabalho e aumentando a confiabilidade do software. Pressman & Maxim (2016).

No contexto deste projeto, os testes podem ser divididos em diferentes categorias:

- Testes de unidade: verificam se cada componente do sistema, como a leitura do sensor de umidade do solo, a ativação da bomba de irrigação e a exibição de informações no display LCD, funciona corretamente de forma independente;
- Testes de integração: garantem que os módulos do sistema trabalham em conjunto de maneira eficiente, como a comunicação entre o Arduino UNO, o sensor e os atuadores responsáveis pela irrigação.

Além desses, outros testes desempenham papéis específicos, como os teste de desempenho, que analisam a resposta do sistema sob diferentes condições, os testes de segurança, que verificam se o sistema pode operar sem riscos, como evitar acionamentos indevidos da bomba que possam desperdiçar água ou danificar o circuito elétrico e os testes de usabilidade que avaliam a clareza das informações exibidas no LCD.

Conforme destacado por Pressman (2016), a execução sistemática desses testes ao longo do desenvolvimento do projeto contribui para um sistema mais eficiente, confiável e alinhado aos objetivos de economia de água.

4.3.3. Desenvolvimento do Código

O desenvolvimento do código seguiu o modelo em cascata, o qual, segundo Pressman (2016), é estruturado em etapas sequenciais bem definidas, permitindo que cada fase seja finalizada antes do avanço à próxima. Essa abordagem foi fundamental para garantir o bom funcionamento do sistema e a organização do desenvolvimento, já que o projeto envolvia diversos componentes integrados.

4.3.4. Estrutura Inicial

O primeiro passo foi a inclusão das bibliotecas essenciais, a definição dos pinos para os dispositivos conectados ao Arduino, a criação de instâncias para o display e para o sensor.

A figura 7, ilustra toda a estrutura inicial do código.

Figura 7 – Estrutura Inicial

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Adafruit_VL53L0X.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

Adafruit_VL53L0X lox;

#define LED_PIN 8
#define SOIL_SENSOR_PIN A0
#define RELAY_PIN 7
#define BUTTON_PIN 2
```

Fonte: Autor (2025)

A biblioteca *Wire.h* é utilizada para comunicação I2C, enquanto a *LiquidCrystal_I2C.h* permite a comunicação com o display LCD I2C. Já a biblioteca

Adafruit_VL53L0X.h é responsável pela comunicação com o sensor de distância VL53L0X, que mede o nível de água no reservatório.

4.3.5. Estrutura Setup

Além da configuração básica dos pinos, a função *setup()* também desempenha um papel essencial na inicialização correta de todos os dispositivos conectados. Esse processo garante que cada componente seja identificado pelo Arduino logo no início, evitando falhas de comunicação ao longo da execução.

Outro ponto relevante é que essa estrutura define as condições iniciais para o sistema. Isso inclui desde o posicionamento inicial dos atuadores até a definição do estado dos sensores, assegurando que o protótipo entre em funcionamento de forma previsível e confiável.

A figura 8, ilustra toda a função *setup()*.

Figura 8 – Estrutura *Setup()*

```
void setup() {  
    Serial.begin(9600);  
  
    lcd.init();  
    lcd.backlight();  
    lcd.clear();  
    lcd.home();  
  
    pinMode(LED_PIN, OUTPUT);  
    pinMode(RELAY_PIN, OUTPUT);  
    pinMode(SOIL_SENSOR_PIN, INPUT);  
    pinMode(BUTTON_PIN, INPUT_PULLUP);  
  
    digitalWrite(LED_PIN, LOW);  
    digitalWrite(RELAY_PIN, LOW);  
  
    if (!llox.begin()) {  
        Serial.println("Falha ao iniciar VL53L0X!");  
        while (1);  
    }  
  
    lcd.setCursor(0, 0);  
    lcd.print("Sistema Iniciado");  
    delay(2000);  
    lcd.clear();  
}
```

Fonte: Autor (2025)

Essa parte garantiu que todos os dispositivos estivessem corretamente configurados para operação.

A figura 9, ilustra a mensagem de “Sistema Iniciado” no LCD, garantindo assim a inicialização correta do sistema.

Figura 9 – Inicialização do Sistema



Fonte: Autor (2025)

4.3.6. Função Medir Distância

A função *medirDistancia()* foi criada para retornar a distância em centímetros detectada pelo sensor.

A figura 10, ilustra a função medir distância.

Figura 10 – Função *medirDistancia()*

```
int medirDistancia() {  
    VL53L0X_RangingMeasurementData_t measure;  
    lox.rangingTest(&measure, false);  
  
    if (measure.RangeStatus != 4) {  
        return measure.RangeMilliMeter / 10;  
    } else {  
        return -1;  
    }  
}
```

Fonte: Autor (2025)

Essa medição é essencial para saber se há água suficiente no reservatório. Quando o valor retorna -1, significa que houve falha na leitura do sensor.

4.3.7. Exibição no LCD e Alerta de Nível de Água

A exibição no LCD é feita através da função *exibirLCD()*, que também aciona o LED caso o nível de água esteja abaixo do esperado. Ao exibir mensagens de status, como o nível da água e possíveis falhas de leitura, o sistema se torna mais acessível e intuitivo.

Além disso, a combinação entre o display e o LED de alerta cria um sistema de comunicação, reforçando a percepção do usuário sobre o estado do reservatório. Essa estratégia é especialmente útil em situações de baixa atenção, já que a presença de um aviso visual aumenta a confiabilidade da informação transmitida. Em projetos futuros, essa etapa pode ser expandida para incluir outros tipos de avisos, como alarmes sonoros ou notificações via aplicativo.

A figura 11, ilustra a função de exibição no LCD.

Figura 11 – Função *exibirLCD()*

```
void exibirLCD(int dist) {
    lcd.setCursor(0, 0);
    if (dist >= 0) {
        lcd.print("Distancia: ");
        lcd.print(dist);
        lcd.print("cm  ");

        lcd.setCursor(0, 1);
        if (dist > 10) {
            lcd.print("Reabastecer!!! ");
            digitalWrite(LED_PIN, HIGH);
        } else {
            lcd.print("Nivel Adequado ");
            digitalWrite(LED_PIN, LOW);
        }
    } else {
        lcd.setCursor(0, 0);
        lcd.print("ERRO NO SENSOR! ");
        lcd.setCursor(0, 1);
        lcd.print("                ");
    }
}
```

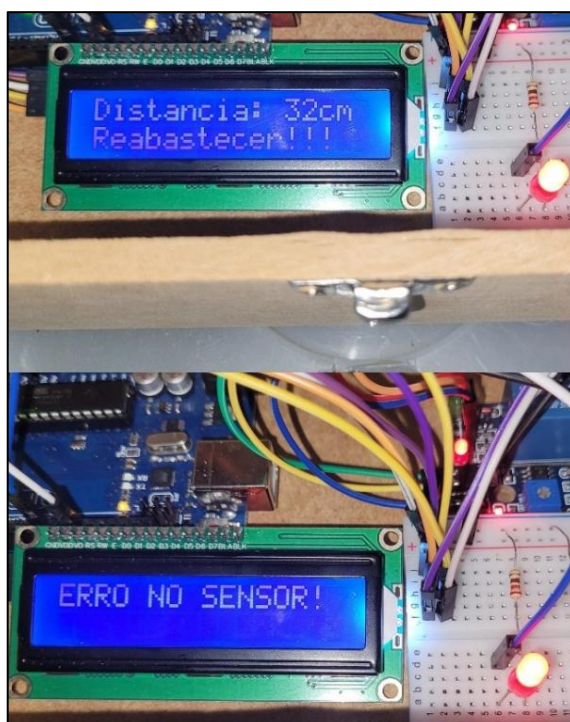
Fonte: Autor (2025)

Essa abordagem de exibição contínua também contribui para a usabilidade, já que o usuário pode monitorar rapidamente o estado do sistema sem precisar interagir diretamente com o Arduino ou abrir softwares complementares. Em outras palavras, o display atua como uma interface homem-máquina simplificada, trazendo clareza às operações internas do protótipo.

Outro aspecto interessante é que o uso do LCD permite que o projeto seja utilizado em diferentes contextos, desde hortas domésticas até pequenos experimentos acadêmicos. A clareza das informações exibidas facilita a replicação do sistema em ambientes educacionais, onde estudantes podem compreender melhor a integração entre sensores, atuadores e a lógica de programação. Dessa forma, o recurso visual não apenas apoia o funcionamento do sistema, mas também reforça seu potencial como ferramenta de aprendizado.

A figura 12, ilustra a exibição da mensagem “ERRO NO SENSOR!” e “Reabastecer!!!” no LCD junto ao LED aceso durante a execução do programa.

Figura 12 – Mensagem de Aviso



Fonte: Autor (2025)

4.3.8. Controle da Bomba de Irrigação

A lógica principal de irrigação do projeto foi implementada na função *controlarBomba()*.

A figura 13, ilustra a implementação da função de controle da bomba de irrigação.

Figura 13 – Função *controlarBomba()*

```
void controlarBomba() {  
  if (!manualIrrigation) {  
    int umidade = analogRead(SOIL_SENSOR_PIN);  
    if (umidade < 750) {  
      digitalWrite(RELAY_PIN, HIGH);  
    } else {  
      digitalWrite(RELAY_PIN, LOW);  
    }  
  }  
}
```

Fonte: Autor (2025)

Este sensor de umidade realiza a leitura de tensão com um valor digital de 0 a 1023. Com essa função, o sistema verifica a umidade do solo constantemente, caso esteja com um valor digital abaixo de 750 a bomba é ativada, quando o valor lido está acima de 750 a bomba é desativada, evitando o desperdício de água.

A figura 14, ilustra o acionamento da bomba de irrigação.

Figura 14 – Bomba de Irrigação



Fonte: Autor (2025)

4.3.9. Irrigação Manual

Para dar ao usuário o controle manual da irrigação, foi utilizado um botão, cuja lógica está na função *loop()*.

A figura 15, ilustra a implementação do controle manual para irrigação.

Figura 15 – Função *loop()*

```
void loop() {  
  if (digitalRead(BUTTON_PIN) == LOW) {  
    piscarLED(3);  
    manualIrrigation = true;  
    digitalWrite(RELAY_PIN, HIGH);  
    digitalWrite(RELAY_PIN, LOW);  
    manualIrrigation = false;  
    delay(3000);  
  }  
}
```

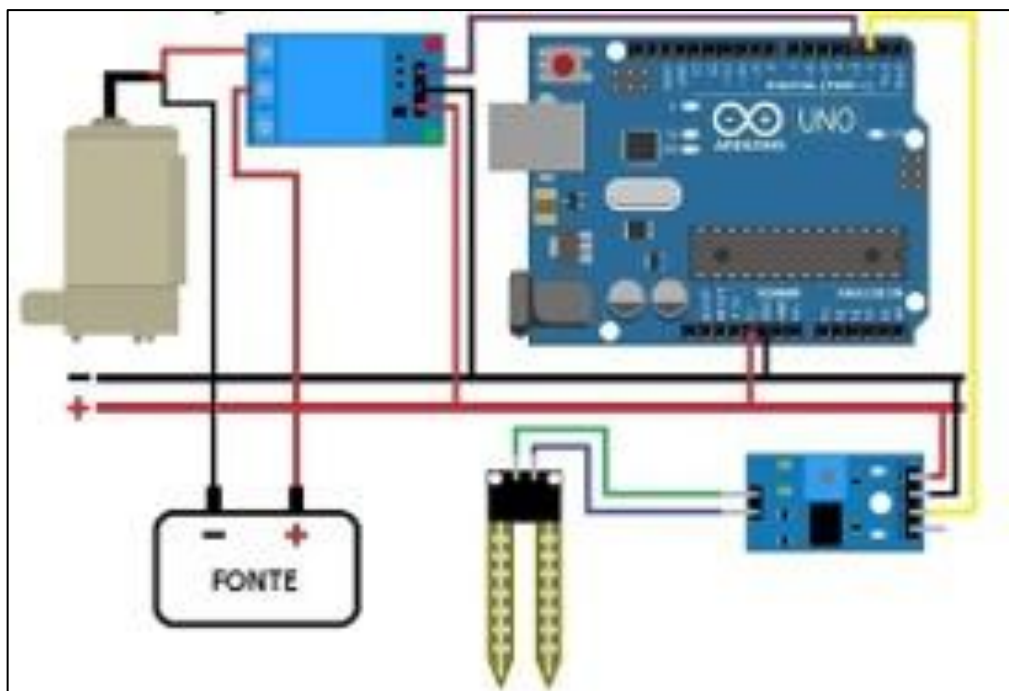
Fonte: Autor (2025)

Quando o botão é pressionado, o sistema realiza três piscadas no LED como forma de confirmação e ativa a bomba por um curto período.

Durante o desenvolvimento, cada trecho de código foi escrito e testado separadamente, assegurando o correto funcionamento de cada módulo antes de sua integração final. Esse processo modular, alinhado ao modelo em cascata, permitiu maior controle sobre o avanço das etapas, minimizando erros durante a fase de integração.

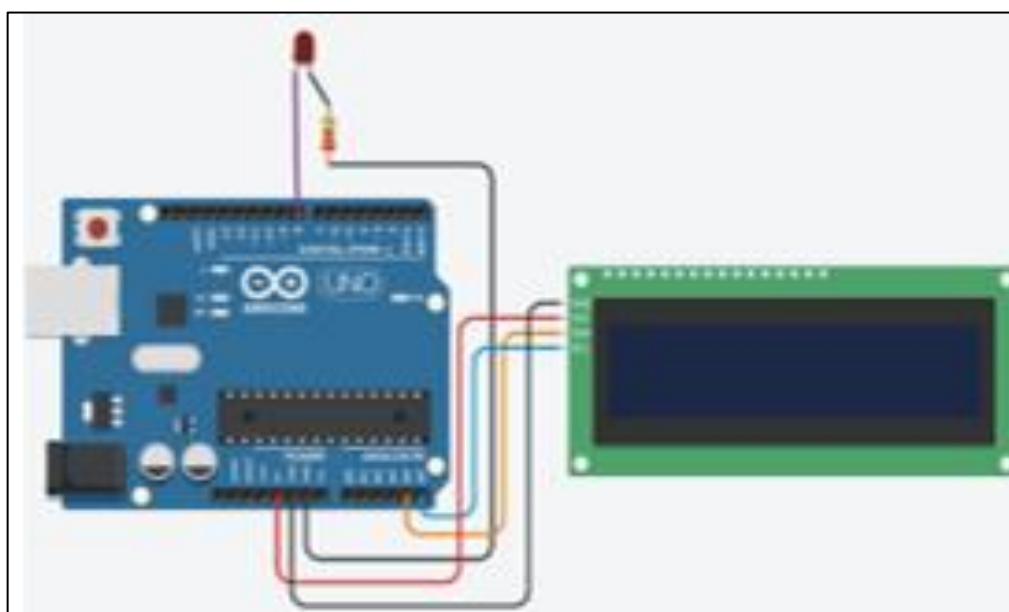
A figura 16, 17 e 18 ilustra a pinagem utilizada no projeto.

Figura 16 – Pinagem da Bomba D'água



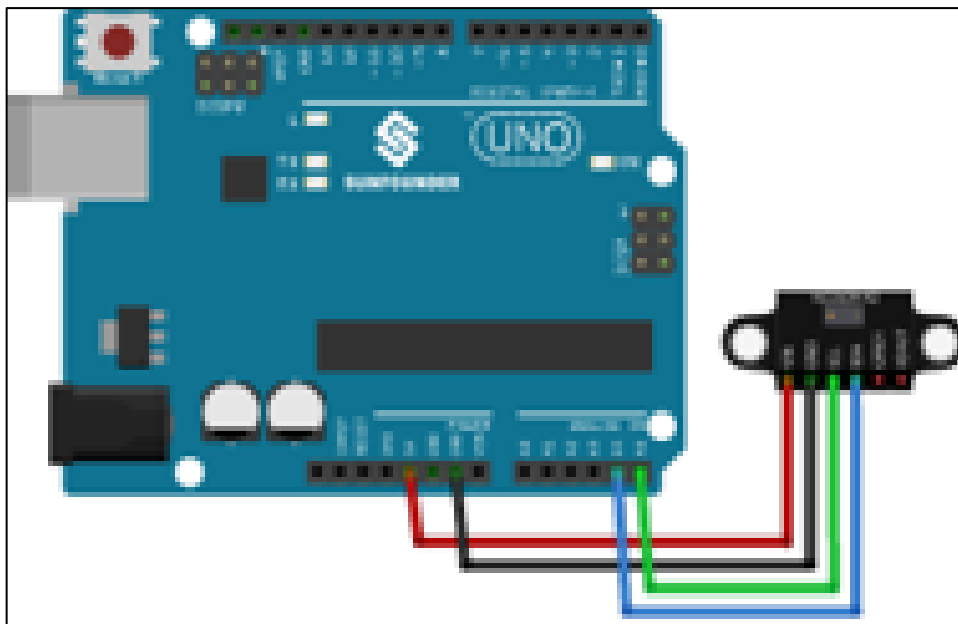
Fonte: Autor (2025)

Figura 17 – Pinagem do LCD



Fonte: Autor (2025)

Figura 18 – Pinagem do Sensor de Distância



Fonte: Autor (2025)

A figura 19, ilustra a vista panorâmica do projeto.

Figura 19 – Vista Panorâmica



Fonte: Autor (2025)

5. CONSIDERAÇÕES FINAIS

O desenvolvimento do sistema de irrigação inteligente com Arduino UNO demonstrou ser uma alternativa viável e eficiente para promover o uso racional da água em pequenos cultivos e ambientes domésticos. A integração de sensores de umidade, sensor de nível de água, atuadores e display LCD permitiu criar um protótipo funcional que alia simplicidade, baixo custo e eficiência no manejo hídrico.

Os testes realizados comprovaram que o sistema é capaz de reduzir desperdícios ao acionar a irrigação somente quando necessário, garantindo um fornecimento adequado de água às plantas. Além disso, a opção de acionamento manual torna o projeto mais flexível e acessível para diferentes perfis de usuários.

Outro aspecto relevante é o impacto educacional e social que a proposta pode gerar. Projetos como este podem estimular a adoção de práticas sustentáveis e servir de base para soluções mais robustas em ambientes agrícolas. No futuro, o sistema pode ser expandido para incluir conectividade sem fio IoT, integração com aplicativos móveis e uso de energias renováveis, como a solar.

Assim, o trabalho alcançou seu objetivo de comprovar a viabilidade de um sistema semiautomático de irrigação de baixo custo, eficiente e sustentável, capaz de auxiliar na redução do consumo de água e de contribuir para um manejo mais consciente desse recurso essencial.

5.1 Dificuldades Encontradas

Durante o desenvolvimento do projeto, algumas dificuldades foram enfrentadas e se mostraram fundamentais para o processo de aprendizado.

A primeira delas esteve relacionada ao tamanho reduzido dos componentes, já que os módulos e sensores, por serem pequenos, ocasionaram em diversos momentos problemas de mau contato nas conexões da protoboard, exigindo constantes ajustes para garantir a estabilidade do sistema.

Outra dificuldade relevante foi a calibração do sensor de umidade do solo, visto que diferentes tipos de terra apresentam variações significativas na capacidade de retenção de água. Dessa forma, foi necessário realizar uma série de testes práticos até se alcançar um valor de referência considerado adequado para o acionamento da bomba de irrigação.

Além disso, a estabilidade das leituras também representou um desafio, pois em algumas situações os sensores apresentaram falhas ou valores inconsistentes, o que poderia provocar acionamentos indevidos do sistema. Para contornar esse problema, tornou-se necessária a implementação de verificações adicionais no código, a fim de garantir maior confiabilidade durante a operação.

REFERÊNCIAS

ADAFRUIT. I2C LCD 16x2 Datasheet. Adafruit, 2022. Disponível em: <https://learn.adafruit.com/i2c-spi-lcd-backpack>. Acesso em: 03 de Março de 2025.

ALMEIDA, F.; SOUZA, R. Uso de sensores para otimização da irrigação em pequenas propriedades. Revista de Tecnologia Agrícola, v. 15, n. 2, p. 45-58, 2019

ARDUINO. Arduino IDE 2.3.4 User Guide. Arduino, 2024. Disponível em: <https://docs.arduino.cc/software/ide-v2>. Acesso em: 28 de Fevereiro de 2025.

ARDUINO. LiquidCrystal_I2C Library Documentation. Arduino, 2022. Disponível em: <https://www.arduino.cc/reference/en/libraries/liquidcrystal-i2c>. Acesso em: 03 de Março de 2025.

CREMASCO M. Lideranças da Netafim visitam projetos de irrigação subterrânea por gotejamento em grãos em Goiás e Mato Grosso. 2021. Disponível em: <https://revistacultivar.com.br/noticias/liderancas-da-netafim-visitam-projetos-de-irrigacao-subterranea-por-gotejamento-em-graos-em-goias-e-mato-grosso>. Acesso em: 09 de Outubro de 2025.

FERREIRA, A. C. Irrigação: Princípios e Métodos. 2. ed. Brasília: EMBRAPA, 2020.

FILLIPI, D. Irrigação do solo: conheça os principais métodos e sistemas. 2020. Disponível em: <https://raks.com.br/conheca-os-principais-metodos-e-sistemas-de-irrigacao/>. Acesso em: 09 de Outubro de 2025.

GOMES, D. P. Eficiência da irrigação e manejo sustentável da água. São Paulo: Oficina de Textos, 2017.

IRRIGAT. Irrigação Semiautomática: Vantagens e Benefícios. Irrigat, 2023. Disponível em: <https://irrigat.com.br/irrigacao-semiautomatica-vantagens-e-beneficios/>. Acesso em: 26 de Fevereiro de 2025.

PENNACCHI, J. P. Irrigação por Gotejamento: o guia completo para o produtor. 2022. Disponível em: <https://aegro.com.br/blog/irrigacao-por-gotejamento/>. Acesso em 09 de Outubro de 2025.

PRESSMAN, R. S. Engenharia de Software: Uma Abordagem Profissional. 7 ed. São Paulo: McGraw-Hill, 2011.

PRESSMAN, R. S.; MAXIM, B. R. Engenharia de Software: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016.

SANTOS, J.; OLIVEIRA, M. Sistemas de Irrigação Automatizados: Eficiência e Sustentabilidade. Editora AgroTech, 2020.

SILVA, M. A. Sistemas de irrigação: fundamentos e aplicações. Campinas: Instituto Agrônômico de Campinas (IAC), 2018.

SILVA, P., et al. Automação na Irrigação: Impacto na Economia de Água e Produtividade Agrícola. Congresso Brasileiro de Engenharia Agrícola, 2021.

SOMMERVILLE, I. Engenharia de Software. 10. ed. São Paulo: Pearson, 2019.

STMICROELECTRONICS. VL53L0X Time-of-Flight Distance Sensor Datasheet. STMicroelectronics 2016. Disponível em: <https://www.st.com/resource/en/datasheet/vl53l0x.pdf> Acesso em: 12 de Março de 2025.

VAHID, F.; GIVARGIS, T. Sistemas embarcados: uma introdução baseada em hardware/software. Rio de Janeiro: LTC, 2011.

RESOLUÇÃO n° 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante Hakawã Luiz Bernardi
do Curso de Engenharia de Computação, matrícula 2021.1.0033.0019-3,
telefone: (66) 9 9607-1282 e-mail hakawa.bernardi@gmail.com, na qualidade de titular dos
direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do autor),
autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o
Trabalho de Conclusão de Curso intitulado
Projeto de Irrigação Semiautomático,
gratuitamente, sem ressarcimento dos direitos autorais, por 5
(cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial
de computadores, no formato especificado (Texto (PDF); Imagem (GIF ou JPEG); Som
(WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da
área; para fins de leitura e/ou impressão pela internet, a título de divulgação da
produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 30 de Setembro de 2025.

Assinatura do(s) autor(es):  **HAKAWA LUIZ BERNARDI**
Data: 30/09/2025 14:33:54-0300
Verifique em <https://validar.iti.gov.br>

Nome completo do autor: Hakawã Luiz Bernardi

Assinatura do professor-orientador:  **ANDRE LUIZ ALVES**
Data: 30/09/2025 20:10:11-0300
Verifique em <https://validar.iti.gov.br>

Nome completo do professor-orientador: André Luiz Alves