

CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE ITAQUERA
PROFESSOR MIGUEL REALE

EDNALDO SEVERINO DA SILVA
LUIZ ANTÔNIO DE ARAÚJO

**CONTROLE ELETRÔNICO VIA AÇÃO DE ARDUINO E SISTEMA DE DOSAGEM
E MISTURADOR**

São Paulo, 2025

EDNALDO SEVERINO DA SILVA

LUIZ ANTÔNIO DE ARAÚJO

**CONTROLE ELETRÔNICO VIA AÇÃO DE ARDUINO E SISTEMA DE DOSAGEM E
MISTURADOR**

Trabalho acadêmico realizado como requisito parcial para a conclusão do curso de Tecnologia em Manutenção Industrial.

Orientador: Me. Daniel Rodrigues de Sousa

Coorientador: Me. Lincoln Nascimento Ribeiro

São Paulo, 2025

FOLHA DE APROVAÇÃO

EDNALDO SEVERINO DA SILVA LUIZ
ANTÔNIO DE ARAÚJO

CONTROLE ELETRÔNICO VIA AÇÃO DE ARDUINO E SISTEMA DE DOSAGEM E MISTURADOR

Projeto tecnológico elaborado como requisito final para a conclusão do Curso Superior de Tecnologia em Manutenção Industrial

Orientador(a): Me. Daniel Rodrigues de Sousa

Aprovado em: ____ / ____ / ____

BANCA EXAMINADORA

Prof. Me. Orientador: Daniel Rodrigues de Souza

Assinatura: _____

Instituição: FATEC ITAQUERA

Data ____ / ____ / ____

Prof. Dr. Leandro Cardoso da Silva

Assinatura: _____

Instituição: FATEC ITAQUERA

Data ____ / ____ / ____

Prof. Especialista: Alex Arnoso

Assinatura: _____

Instituição: FATEC ITAQUERA

Data ____ / ____ / ____

RESUMO

A automação e o controle de processos são essenciais para a evolução tecnológica e a otimização industrial, destacando-se pelo uso de tecnologias como o Arduino e o (Controle Lógico Programáveis) CLP. O Arduino é amplamente utilizado para prototipagem e projetos menores devido à sua flexibilidade e custo acessível, enquanto o CLP se sobressai pela robustez e confiabilidade em ambientes industriais complexos. Na automação industrial, máquinas como agitadores com palhetas desempenham papel crucial. Esses equipamentos são usados para misturar líquidos e materiais com propriedades complexas. Com controle integrado ao Arduino e monitoramento via IoT, é possível ajustar a velocidade dos atuadores e sensores para garantir precisão na dosagem e homogeneização. O software FluidSIM otimiza o controle e a correção de variações, garantindo assim maior eficiência nos processos. A automação moderna vai além da simples redução do trabalho repetitivo; ela cria funções especializadas, impulsiona a produtividade e a qualidade, e contribui para a sustentabilidade. Essa integração tecnológica redefine os processos produtivos, elevando os padrões e a competitividade no setor industrial.

Palavras-chave: Controle Eletrônico Via Ação de Arduino, e CLP.

ABSTRACT

Automation and process control are essential for technological evolution and industrial optimization, notably through the use of technologies such as Arduino and PLC (Programmable Logic Controllers). Arduino is widely used for prototyping and smaller projects due to its flexibility and affordable cost, while PLCs stand out for their robustness and reliability in complex industrial environments. In industrial automation, machines such as paddle stirrers play a crucial role. These devices are used to mix liquids and materials with complex properties. With control integrated via Arduino and monitoring through IoT, it is possible to adjust the speed of actuators and sensors to ensure precision in dosing and homogenization. FluidSIM software enhances the control and correction of variations, ensuring efficiency. Modern automation, by reducing repetitive work, creates new specialized functions, in addition to promoting greater productivity, quality, and sustainability. This integration of technologies redefines production processes, raising standards and competitiveness in the industrial sector.

Keywords: Electronic Control via Arduino and PLC Action.

LISTA DE FIGURAS

Figura 1 - Lógica com relés e lógica Ladder.....	15
Figura 2 - Circuito de Selo.....	16
Figura 3 - CLP com relés	16
Figura 4 - Diagrama em blocos, típico de um CLP.....	17
Figura 5 - Arduino Ethernet Shield.	22
Figura 6 - Arduino Bluetooth.	22
Figura 7 - Arduino e suas conexões.....	23
Figura 8 - Arduino uno	24
Figura 9 - Conectores do Arduino.	25
Figura 10 - Conectores do Arduino.	27
Figura 11 - Arduino IDE Tela Principal do Arduino.....	28
Figura 12 - Sistema de controle em malha fechada.	31
Figura 13 - Passos na programação do Arduino.	32
Figura 14 - Módulo Arduino relés.	33
Figura 15 - Microbomba dosadora.	34
Figura 16 - Bombas: Dosagem.....	36
Figura 17 - Fonte de alimentação.....	36
Figura 18 - Sensor de pressão.....	37
Figura 19 - sensor de nível tipo boia (float switch).	38
Figura 20 - Motores de passo.....	38
Figura 21 - Fluxograma do processo.....	41
Figura 22 - Reservatório e Bombas.....	50
Figura 23 - Simulação Tinkecard.....	51
Figura 24 - Válvulas.	52
Figura 25 - Arduino.	52
Figura 26 - Relés.....	52
Figura 27 - Simulação com o relé.....	52
Figura 28 - sensor de nível.....	53
Figura 29 - Motor DC.....	59
Figura 30 - Motor DC.....	59
Figura 31 – PWM	61
Figura 32 – PWM	61
Figura 33 – Potencômetro	62
Figura 34 - Display LCD 16x2	65
Figura 35 - Display LCD 16x2	65
Figura 36 - Esquema Elétrico.....	66
Figura 37 - Painel de Comando.....	69
Figura 38 - Painel de Comando.....	69
Figura 39 - Motor DC.....	71
Figura 40 - Módulo Relé.....	73
Figura 41 - Módulo PWM (42) Display LCD 16x2 (43)	75
Figura 42 - Fonte chaveada, 12V DC.....	76
Figura 43 - Painel montado	80
Figura 44 - Painel montado.	83

SUMÁRIO

1. INTRODUÇÃO	9
2. JUSTIFICATIVA	11
3. OBJETIVOS.....	12
3.1 Objetivo Geral	12
3.2 Objetivos Específicos	12
4. METODOLOGIA	13
5. DESENVOLVIMENTO	14
5.1 A criação do Controlador Lógico Programável (CLP).....	14
5.1.1 <i>Controlador Lógico Programável (CLP)</i>	<i>14</i>
5.1.2 <i>Características e Evolução dos CLPs.....</i>	<i>14</i>
5.1.3 <i>Engenharia de Controle e CLPs.....</i>	<i>15</i>
5.1.4 <i>Funcionamento dos Sistemas de Controle com CLPs.....</i>	<i>15</i>
5.1.5 <i>Fluxograma de funcionalidade dos CLPs</i>	<i>17</i>
5.2 Sistema Automatizado.....	18
5.3 Automação Industrial	18
5.4 Arduino	19
5.4.1 <i>Estrutura do Arduino</i>	<i>20</i>
5.4.2 <i>Etapas do funcionamento do Arduino.....</i>	<i>20</i>
5.5 Internet das Coisas (IoT)	21
5.5.1 <i>Arduino Ethernet Shield</i>	<i>21</i>
5.5.2 <i>Arduino Bluetooth</i>	<i>22</i>
5.6 Prototipagem.....	23
5.7 Tecnologias Aplicadas	24
5.8 Hardware – Arduino	25
5.8.1 <i>Alimentação</i>	<i>25</i>
5.8.2 <i>Conectores.....</i>	<i>25</i>
5.8.3 <i>Funções dos conectores.....</i>	<i>26</i>
5.8.4 <i>Oscilador.....</i>	<i>27</i>
5.8.5 <i>Interface USB.....</i>	<i>28</i>
5.9 Software.....	28
5.10 Desenvolvimento para novas aplicações para o Arduino.....	29
5.11 MATERIAIS E MÉTODOS.....	29
5.11.1 <i>Método da Pesquisa</i>	<i>29</i>
5.11.2 <i>Diagrama de bloco Arduino.....</i>	<i>32</i>
5.11.3 <i>Arduino relés 5 V módulo 8 canais</i>	<i>32</i>

5.11.4	<i>Bombas dosadora</i>	33
5.11.5	<i>Sensor de pressão</i>	36
5.11.6	<i>Sensor Nível</i>	37
5.11.7	<i>Motor Misturador</i>	38
5.12	Desenvolvimento projeto do Arduino	39
5.12.1	<i>Sistema de Dosagem Automática de Tinta com Arduino</i>	40
5.13	DESCRIÇÃO E IDENTIFICAÇÃO DOS COMPONENTES	42
5.13.1	<i>Lista de peças referente a montagem do dosador</i>	42
5.13.2	<i>Lista de peças referente a montagem do dosador</i>	43
5.13.3	<i>Posicionamento de montagem dos pinos do Arduino Uno.</i>	43
5.13.4	<i>LIGAÇÕES DO MÓDULO RELÉ E ATUADORES</i>	44
5.13.5	<i>Tabela de Custos dos Componentes eletrônica do dosador.</i>	46
5.14	METODOLOGIA TÉCNICA E APLICAÇÃO DO PROJETO	48
5.14.1	<i>Calibração E Teste Será Verificado Durante O Ensaio</i>	50
5.15	Hardware; montagem dos componentes	51
5.15.1	<i>Identificação dos componentes</i>	51
5.16	PRINCÍPIO DE FUNCIONAMENTO DO SENSOR NÍVEL	53
5.16.1	<i>RELATÓRIO TÉCNICO DO FUNCIONAMENTO DOS SENSORES (MODELO- XKC-Y25-V)</i>	55
5.17	Botões e LEDs:	56
5.18	Motor Elétrico de Corrente Contínua (DC) com Escovas	58
5.19	FUNCIONAMENTO PWM, (Pulse Width Modulation)	60
5.20	Esquema, pino a pino – ligação direta (usar potenciômetro do HW-0 .	62
5.21	Display LCD 1602 com Módulo I2C	63
5.22	LCD 16x2 (COM MÓDULO I2C, CABIADO)	66
5.23	Função dos componentes no esquema do projeto dosador	67
5.24	Descrição da Montagem	67
5.25	Considerações de Segurança	69
5.26	MOTOR DC ESCOVADO (5 – 12 V)	70
5.27	MÓDULO DE RELÉS 8 CANAIS (SRD-05VDC-SL-C)	72
5.28	MÓDULO PWM, FIGURA 42 – CONTROLADOR DE VELOCIDADE DE MOTOR DC	74
5.29	FONTE CHAVEADA 12 V DC (INDUSTRIAL)	76
5.30	RESUMO FUNCIONAL DO SISTEMA	77
5.31	FUNCIONAMENTO ELÉTRICO EM CONJUNTO	78
5.31.1	<i>Mapeamento e Identificação dos Pinos no Esquema</i>	83
6.	conclusão	85

REFERÊNCIAS.....	86
Anexos	94
Apêndice a - CHECKLIST DE TESTES - SISTEMA DE BOMBAS PERISTÁLTICAS PWM	102
Apêndice b - TABELA DE CONEXÕES ELÉTRICAS – SISTEMA DE BOMBA PWM	
103	
apêndice d - Painel Do Projeto.....	105
apêndice e - Base Dos Tanques De Dosagem	106
apêndice f - Posição Dos Tanques	107
apêndice g - Helice de Agitação.....	108
apêndice h - Tanque De Agitação	109
apêndice i - Diagrama Elétrico Funcionamento Dos Sensores De Nivel, Elaborado No Cade_Simu.....	110
apêndice j - Diagrama Elétrico Lcd I2c Lógica Arduino.....	111
apêndice k - Diagrama Elétrico Painel De Comando Desenvolvido No, Softwer Cade_Simu.....	112
apêndice l - Diagrama Do Circuito Eletrônico Arduino Lógica Bloco De Relés 8 Canais, Desenvolvido No Softwer Cade_Simu	113
apêndice m - Diagrama Elétrico De POTÊNCIA, PARA Ligação Do Painel De Comando Desenvolvido No Cade_ Simu	114
ApêNDICE N - DIAGRAMA ELÉTRICO DE POTÊNCIA, PARA LIGAÇÃO DO PAINEL DE COMANDO DESENVOLVIDO NO CADE_ SIMU.....	115
APÊNDICE O – DIAGRAMA DE COMANDO CONTROLE DE VELOCIDADE VIA PWM E CHAVEANEETO DE RELES R7 E R8, DSENVOLVIDO DO SOFTWER CAdESIMU.....	116
apêndice p - FICHA TÉCNICA DOS PARAFUSOS FICHA TÉCNICA – PARAFUSOS MÉTRICOS M3 / M10 / M12.....	117
apêndice Q - FICHA TÉCNICA DAS PORCAS M10, M12	119
apêndice R - FICHA TÉCNICA DOS PARAFUSOS	122

1. INTRODUÇÃO

A automação é a intervenção ativa sobre um processo ou as condições que afetam, visando manter variáveis dentro de limites determinados. A introdução de tecnologias como o Arduíno e o CLP (Controlador Lógico Programável) desempenham um papel central na transformação dos processos produtivos. O Arduíno, com sua flexibilidade e custo acessível, permite a criação de sistemas automatizados que combinam sensores, atuadores e comunicação em tempo real (Carvalho; Santos; Santos, 2023). Já o CLP se destaca por sua robustez e confiabilidade, sendo ideal para ambientes industriais exigentes (Ribeiro, 1999).

O Arduino, com sua flexibilidade e custo acessível, permite a criação de sistemas automatizados que integram sensores, atuadores e comunicação em tempo real (Carvalho; Santos, 2023). Já o CLP destaca-se por sua robustez e confiabilidade, sendo ideal para ambientes industriais exigentes (Ribeiro, 1999).

Segundo Ribeiro (1999), no contexto produtivo, o controle automático busca otimizar sistemas, permitindo a fabricação de produtos com menores custos, maior produtividade, redução de tempo e qualidade aprimorada. Atualmente, a automação industrial é uma realidade consolidada em praticamente todas as fábricas do mundo.

Embora a automação possa reduzir a necessidade de mão de obra direta, ela também cria oportunidades de emprego em funções especializadas para operar sistemas mais complexos. Esses profissionais, devido à sua especialização, recebem remunerações mais altas (Ribeiro, 1999).

A engenharia de controle, fundamentada no princípio da realimentação (feedback), é essencial para a gestão de variáveis em sistemas automatizados. Suas aplicações abrangem instrumentação, hardware de medição e controle, projeto de sistemas, comunicação digital, programação e manutenção (Ribeiro, 1999).

A centralização e o monitoramento em tempo real das variáveis revolucionaram a produção, permitindo níveis de precisão e eficiência antes inatingíveis com o controle manual.

Dos sistemas mecânicos e hidráulicos aos digitais e automatizados, os avanços tecnológicos redefiniram a fabricação de produtos, resultando em maior produtividade, melhor qualidade e redução de custos (BAYER,2011).

Quanto às máquinas agitadoras de palhetas, elas são amplamente utilizadas em instalações de processamento e manufatura em todo o mundo, sendo aplicadas na mistura de alimentos, produtos químicos, farmacêuticos, artigos domésticos, plásticos, minerais e outros materiais. Devido à grande variedade de produtos e processos — como agitação, mistura, dispersão e homogeneização — envolvidos na produção de tintas, vernizes e revestimentos, é essencial dimensionar e projetar agitadores industriais para pintura de forma personalizada (MOBILBATCH, 2025).

2. JUSTIFICATIVA

O desenvolvimento deste projeto justifica-se pela crescente necessidade de otimização nos processos industriais, onde a automação e o controle desempenham papel essencial na garantia da qualidade e na redução de custos operacionais. No contexto da manipulação de líquidos e tintas, processos manuais estão sujeitos a erros de dosagem, desperdício de matéria-prima e falta de homogeneidade na mistura final.

A escolha da plataforma Arduino como núcleo de controle fundamenta-se em sua flexibilidade e custo acessível, permitindo a criação de sistemas eficientes de prototipagem e aplicações industriais de menor porte, sem a necessidade dos altos investimentos tipicamente exigidos por Controladores Lógicos Programáveis (CLPs) de grande porte. Além disso, a integração com tecnologias de monitoramento e a possibilidade de comunicação via IoT alinham o projeto às tendências da Indústria 4.0, promovendo maior conectividade e eficiência.

Dessa forma, este trabalho propõe uma solução técnica viável e econômica para automatizar a dosagem e a mistura, substituindo o trabalho manual repetitivo por um sistema capaz de realizar funções especializadas com maior precisão, contribuindo para a produtividade e a competitividade no setor.

3. OBJETIVOS

3.1 Objetivo Geral

O objetivo deste projeto é automatizar o controle de dosagem pela ação da velocidade do atuador para o estudo dos dados obtidos e corrigir possíveis variações através do controlador Arduino. O sistema será integrado com dispositivos de comunicação via portas lógicas e digitais para controle de sensores de posição.

3.2 Objetivos Específicos

Planejamento do Projeto Eletrônico: assegurar agilidade e precisão na execução das funções, reduzindo o tempo dos processos de fabricação e minimizando erros. Essa abordagem é viável tanto para produções em pequena quanto em grande escala.

Desenvolvimento do Sistema: o presente trabalho será desenvolvido e testado em ambiente de laboratório, utilizando recursos próprios e externos. A infraestrutura necessária será disponibilizada pela instituição de ensino, viabilizando o desenvolvimento do sistema.

4. METODOLOGIA

A metodologia aplicada classifica-se como pesquisa experimental, focada no desenvolvimento de um sistema automatizado de dosagem. O protótipo utiliza um Arduino Uno integrado a módulos de relés, bombas peristálticas e um motor de agitação controlado via PWM, com monitoramento via sensores de nível capacitivos e interface por display LCD.

A montagem segregou circuitos de potência e controle para proteção elétrica, enquanto o software foi estruturado em uma máquina de estados na IDE do Arduino para gerenciamento simultâneo de tarefas. A validação do projeto ocorreu através de testes unitários e calibração dos tempos de dosagem, garantindo a funcionalidade e a precisão operacional do conjunto.

5. DESENVOLVIMENTO

5.1 A criação do Controlador Lógico Programável (CLP)

5.1.1 Controlador Lógico Programável (CLP)

A criação do Controlador Lógico Programável (CLP) representou um marco na automação industrial, substituindo os sistemas de controle a relé por um dispositivo

eletrônico programável e resistente. A indústria automobilística, com a General Motors como pioneira em 1969, foi a primeira a adotar essa tecnologia, que rapidamente se tornou o padrão para controle automatizado na manufatura. Dick Morley é amplamente reconhecido como o "pai do CLP" por seu papel crucial na idealização dessa solução em 1968, culminando no desenvolvimento do MODICON 084 por sua equipe na Bedford Associates. O CLP surgiu em um momento em que a indústria buscava alternativas mais confiáveis e de menor manutenção aos sistemas de relés, sendo projetado com a linguagem ladder, que facilitou sua compreensão e utilização por técnicos e engenheiros devido à sua semelhança com diagramas de circuitos de relés (PLC, 2021).

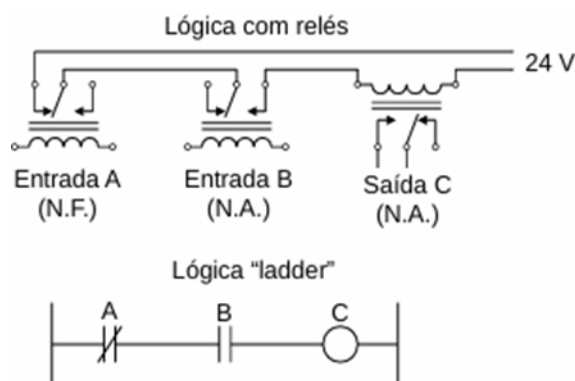
5.1.2 Características e Evolução dos CLPs

Esses computadores industriais compactos e modulares foram projetados para operar em ambientes fabris adversos, resistindo a condições como poeira, umidade e variações extremas de temperatura. Sua capacidade de executar lógicas de controle complexas de forma confiável e centralizada consolidou sua importância na automação, mantendo sua relevância e contínua evolução até os dias atuais. O aprimoramento dos sistemas de controle industrial com a introdução do CLP no século XX transformou a eficiência e a flexibilidade dos processos produtivos (PLC, 2021).

5.1.3 Engenharia de Controle e CLPs

A engenharia de controle evoluiu de sistemas controlados por humanos para sistemas elétricos baseados em relés e, mais recentemente, para CLPs. Os relés, que utilizam um campo magnético para acionar chaves elétricas, foram cruciais no controle elétrico inicial e permitiram a implementação de lógica simples. O desenvolvimento do CLP na década de 1970 revolucionou o controle de manufatura, com a linguagem ladder sendo o principal método de programação - escolhido estrategicamente por sua semelhança com os diagramas de relés, o que minimizou a necessidade de retreinamento. Embora os relés ainda estejam presentes em sistemas modernos, seu uso para lógica complexa tornou-se raro, sendo o CLP a solução dominante. Um relé funciona energizando uma bobina para criar um campo magnético que move um contato, fechando os contatos normalmente abertos (NA) e abrindo os contatos normalmente fechados (NF). Como mostra a Figura 1, um circuito com relés NA e NF pode controlar uma saída C, representada tipicamente em lógica ladder. Neste exemplo, C ativa-se quando A está desativado e B está ativado (Mendes, 2021).

Figura 1 - Lógica com relés e lógica Ladder.



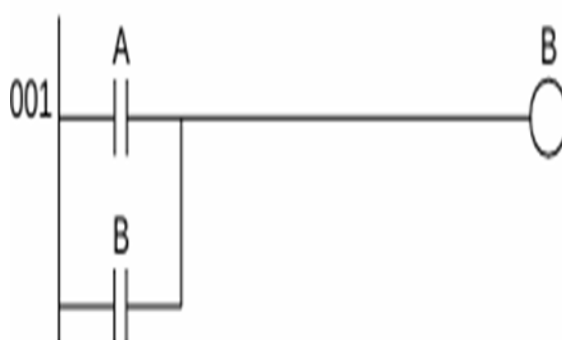
Fonte: Mendes, 2021.

5.1.4 Funcionamento dos Sistemas de Controle com CLPs

Muitos iniciantes cometem o erro de tentar fazer com que a lógica ladder corresponda estritamente aos tipos de entrada física (Mendes, 2021).

A Figura 2 ilustra um circuito de selo (latching circuit), no qual um relé de saída funciona simultaneamente como entrada, permitindo a manutenção do estado da saída mesmo após a desativação do sinal de entrada original. Este conceito demonstra a capacidade de realimentação em circuitos de relés, possibilitando a criação de circuitos de retenção (Mendes, 2021).

Figura 2 - Circuito de Selo

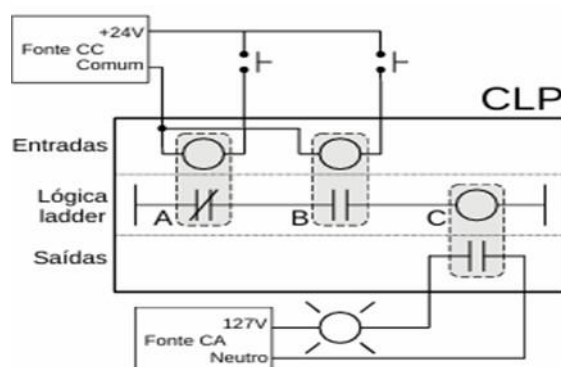


Fonte: Mendes, 2021.

Quando o botão A é acionado, a saída B é ativada e, através da realimentação, a entrada B também é energizada, mantendo o circuito permanentemente ativado até que a alimentação seja interrompida.

Nota: A linha de retorno à direita foi omitida intencionalmente, estando implicitamente representada nestes diagramas (Mendes, 2021).

Figura 3 - CLP com relés



Fonte: Mendes, 2021.

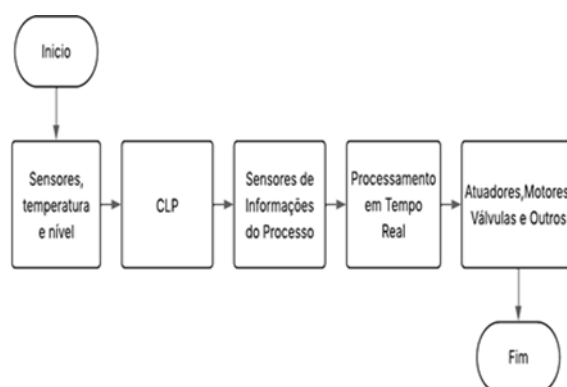
O funcionamento de sistemas de controle que utilizam Controladores Lógicos Programáveis (CLPs) estabelece uma distinção entre a lógica do sistema e sua implementação prática. A Figura 2 apresenta uma visão simplificada da lógica de controle, enquanto a Figura 3 exibe um diagrama completo do CLP, incluindo seus circuitos de entrada e saída. Nesta representação, o CLP recebe sinais de entrada provenientes de botões, os quais ativam relés de 24 VCC que, por sua vez, acionam um relé de saída para controlar dispositivos como lâmpadas de 115 VCA. É fundamental observar que, em CLPs reais, as entradas não são implementadas por relés, embora as saídas possam sê-lo. A linguagem de programação do CLP é denominada (diagrama ladder), um programa computacional que pode apresentar diferenças em relação às entradas e saídas físicas.

5.1.5 Fluxograma de funcionalidade dos CLPs

O funcionamento dos CLPs em sistemas de automação segue um ciclo com três etapas principais: leitura de entradas, processamento e atuação nas saídas. Sensores captam dados físicos como temperatura ou nível e os convertem em sinais elétricos.

O CLP processa essas informações conforme o programa armazenado, aplicando lógicas previamente definidas. Com base nesse processamento, o CLP envia comandos aos atuadores, como motores e válvulas, que executam as ações físicas necessárias. Esse ciclo se repete continuamente, permitindo o controle automatizado e eficiente de processos industriais. Conforme descrito na Figura 4.

Figura 4 - Diagrama em blocos, típico de um CLP.



Fonte: Autores, 2025.

5.2 Sistema Automatizado

A Relação entre CLP e IHM em Sistemas de Automação. Embora o CLP seja o "cérebro" do sistema de automação, executando a lógica e o controle, a Interface Homem-Máquina (IHM) atua como a "ponte" essencial entre o operador humano e a complexidade do sistema. A IHM desempenha um papel fundamental ao: Contextualizar a operação: A IHM vai além da simples representação do que ocorre em campo. Ela está inserida em um contexto operacional e, portanto, não pode ser desenvolvida de forma isolada (ex.: luminosidade, ruído, quantidade de operadores, aspectos ergonômicos, entre outros).

Agir como camada de proteção: Juntamente com o Sistema de Alarmes, a IHM pode se tornar uma camada adicional de proteção e resposta a condições anormais do processo.

Centralizar informações e decisões: Sendo o centro de informações e o ponto de tomada de decisões, a gestão adequada da IHM é fundamental para garantir seu desempenho e confiabilidade.

Enquanto os CLPs garantem a automação precisa, a IHM é a ferramenta que transforma o sistema complexo em uma plataforma gerenciável e segura para o operador, tornando-se um componente indispensável para o sucesso e a eficiência de qualquer processo industrial automatizado (ANSI/ISA-101.01-2015).

5.3 Automação Industrial

A automação industrial é bem definida pela interação de sistemas integrados, computacionais e mecânicos, desenvolvidos para implementação e otimização de processos produtivos.

Isso resulta em um conjunto integrado de sistemas e tecnologias com a finalidade de melhorar a produtividade, qualidade e eficiência dos processos, minimizando custos e garantindo melhor infraestrutura e segurança de forma automatizada. A automação pode ser aplicada em diversos segmentos, como a indústria automobilística, química, alimentícia e outros.

A automação industrial é uma área em constante evolução, com a utilização de tecnologias avançadas como a Internet das Coisas (IoT), a Indústria 4.0 e a Inteligência Artificial (IA), que se tornam cada vez mais uma realidade na industrialização e otimização de processos industriais. Essas tecnologias tornam os processos mais eficientes, proporcionando diminuição de custos operacionais, segurança na automação e agilidade na integração de sistemas.

Diante desse contexto, temos diversas ferramentas que se destacam de forma versátil para o desenvolvimento tecnológico de projetos, interagindo com plataformas eletrônicas, software e hardware. Nesse contexto, destaca-se uma ferramenta acessível utilizada no desenvolvimento de projetos e dispositivos eletrônicos, proporcionando precisão e confiabilidade nos processos:

O Arduino tem sido utilizado em projetos diversificados na automação de processos industriais com uma plataforma de prototipagem eletrônica baseada em hardware e software de código aberto, permitindo a criação de sistemas na automação com fácil acesso, baixo custo e facilidade de manutenção. Com esse objetivo, o Arduino na automação industrial se apresenta com suas vantagens e limitações em diferentes contextos e aplicações produtivas (Carvalho,2023).

5.4 Arduino

Desenvolvido em 2005 por Massimo Banzi e David Cuartielles. Segundo Banzi (2011), o objetivo era fornecer uma ferramenta simples e acessível para desenvolvimento de projetos eletrônicos que não necessitasse de conhecimentos avançados em eletrônica, possibilitando que qualquer pessoa acesse, melhore e modifique a plataforma conforme a necessidade do projeto e do processo a ser aplicado.

5.4.1 Estrutura do Arduino

Segundo Banzi (2011), o microcontrolador é programado por uma linguagem de programação baseada na linguagem C. A placa do Arduino possui uma série de pinos de entrada e saída, permitindo a conexão de componentes eletrônicos como sensores, atuadores, displays e outros. Segundo Monk e Scherz (2013), a maioria dos pinos de entrada e saída são compatíveis com os padrões de interface digital (HIGH/LOW) e analógica (PWM), permitindo uma ampla variedade de conexões com diversos componentes eletrônicos.

5.4.2 Etapas do funcionamento do Arduino

Segundo Monk e Scherz (2013), o funcionamento do Arduino se destaca em quatro etapas: Carregamento do programa: O programa é carregado na placa Arduino e armazenado na memória do microcontrolador.

Interpretação do programa: Durante a interpretação do programa, o microcontrolador analisa cada instrução e determina qual ação deve ser tomada. As instruções podem envolver várias ações e contextos operacionais deste controle, leitura, operacionais, interpretação e instrução.

Execução das instruções: O microcontrolador executa as ações determinadas pelo programa desenvolvido.

Interação com o ambiente externo: A interação ocorre através da leitura das entradas e saídas por meio dos pinos da placa Arduino. O Arduino foi originalmente projetado para aplicações de prototipagem com fácil acesso, proporcionando integração em aplicações industriais para sistemas automatizados.

Nesse contexto, o Arduino possui limitações com relação à capacidade de processamento em aplicações em ambientes mais agressivos onde se exige mais robustez nas suas aplicações.

Vantagens do Arduino na Automação Industrial: Embora haja limitações e desafios na automação com aplicação do Arduino, a plataforma apresenta vantagens em relação a outros sistemas de automação industrial.

O baixo custo e a facilidade de uso tornam o Arduino atraente para pequenas e médias empresas. A plataforma é fácil de aprender e usar, podendo ser adaptada a uma ampla gama de aplicações industriais com sua capacidade de conexão a diversos dispositivos de entrada e saída a capacidade de expansão entre módulos e placas. Isso possibilita o desenvolvimento de projetos customizados e adaptados às necessidades específicas dos processos industriais, sem a necessidade de investimentos altos e sistemas complexos (Rodrigues,2015).

5.5 Internet das Coisas (IoT)

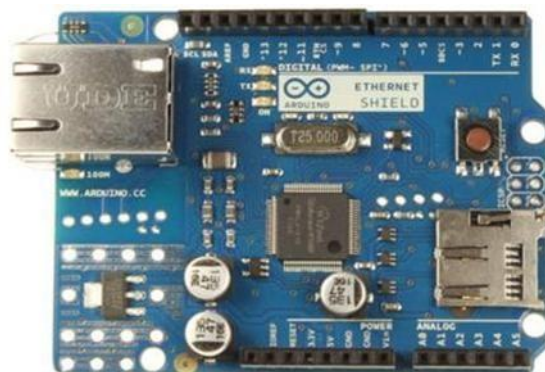
A Internet das Coisas (IoT) refere-se à interconexão de dispositivos físicos à internet, permitindo que eles colem e transmitam dados de maneira inteligente. O Arduino é uma ferramenta excelente nesse contexto, pois sua plataforma permite desenvolver dispositivos IoT devido à sua versatilidade, acessibilidade e baixo custo. Isso possibilita a criação de dispositivos capazes de se comunicar entre si e com a internet, coletando dados em tempo real e enviando-os para a nuvem, permitindo o monitoramento dos dados obtidos através de aplicativos ou interfaces web (Ferroni,2015).

5.5.1 *Arduino Ethernet Shield*

O Arduino Ethernet Shield é conectado ao Arduino de forma empilhada, basta ligar este módulo à placa Arduino (Lima,2015). permite que uma placa Arduino conecte-se à internet. Ele baseia-se na Wiznet W5100 chip de Ethernet que, por sua vez, fornece uma rede (IP) que suporta até quatro conexões de soquetes simultâneos.

A conexão é feita através de cabo Ethernet convencional par trançado Categoria 5. A Figura 5, apresenta o Arduino Ethernet Shield.

Figura 5 - Arduino Ethernet Shield.



Fonte: DESTACOM,2012.

5.5.2 *Arduino Bluetooth*

O Arduino BT é uma placa de microcontrolador que foi originalmente baseada no ATmega168, mas que atualmente é fornecido com o 328. Ele suporta comunicação serial sem fio por Bluetooth (embora não seja compatível com fones de ouvido Bluetooth ou outros dispositivos de áudio). Tem 14 pinos digitais de entrada / saída, dos quais 6 podem ser usados como saídas PWM e ainda para redefinir o WT11 módulo, 6 entradas analógicas, e uma entrada que trabalha na faixa de 16 MHz, cristal oscilador, terminais de parafuso e um botão de reset. Ele contém o necessário para suportar o microcontrolador e pode ser programado sem fio através da conexão Bluetooth (Lima,2015). Arduino, Figura 6.

Figura 6 - Arduino Bluetooth.



Fonte: DESTACOM,2012.

5.6 Prototipagem

A prototipagem é essencial para o desenvolvimento, permitindo a criação de versões iniciais para testar conceitos e funcionalidades com o intuito de produzir em grande escala. O uso do Arduino na prototipagem é facilitado pelo baixo custo das placas e pela programação simples, o que acelera o desenvolvimento de protótipos funcionais, mesmo para usuários sem conhecimento aprofundados em eletrônica. Possui plataforma de hardware computação física baseada em microcontrolador simples. Possui capacidade de dados via sensores (temperatura, umidade, fumaça vibração, infravermelho, presença), e saída de dados, para atuadores (motores, relés, leds), lâmpadas. E equipamento com oscilador de 16 MHz, um regulador de tensão de 5 Volts, portas de alimentação e USB. Os sensores permitem a captação de dados ambientais como temperatura, umidade, presença luminosidade. sua plataforma de hardware e computação física, baseada em uma placa de microcontrolador simples e um ambiente de desenvolvimento para escrever o software de controle. não possui um fabricante oficial, não existe um Arduino original ou produto pirata, o projeto é livre. O esquema lógico e elétrico é disponibilizado no site oficial para quem desejar construir o seu próprio Arduino, Figura 7, (Felisardo, 2023).



Fonte: DESTACOM,2012.

5.7 Tecnologias Aplicadas

Massimo Banzi (2012) com objetivo de ensinar programação de forma aplicado a projetos de arte, automação e robótica superando a falta de placas didáticas e poderosas e baratas no mercado, juntamente com o engenheiro espanhol David Cuartielles, decide criar sua própria placa contando com ajuda de um de seus alunos chamado David Mellis, o qual ficou responsável pela linguagem de programação do Arduino.

O Arduino, conforme McRobertsé (2010), é um sistema embarcado, de código aberto (hardware e software em C ou C++), funcionalmente análogo a um CLP, que integra automação e controle em contextos domésticos, comerciais ou moveis. O Arduino pode ter suas aplicações estendidas utilizando placas que contêm outros dispositivos, as quais são facilmente conectadas a ele. Estas placas são chamadas de módulos ou Shields Tais placas podem funcionar como receptores GPS, módulos de rede Ethernet ou wireless, dentre outros (McRoberts, 2010). Existem diversos modelos de placa da plataforma Arduino. A Figura 8, apresenta uma versão recente do Arduino, chamada Arduino UNO.

Figura 8 - Arduino uno



Fonte: DESTACOM,2012.

5.8 Hardware – Arduino

A placa Arduino e, Segundo Monk (2010), essencialmente uma estrutura de suporte para microcontrolador, permitindo comunicação com outros dispositivos via portas de acesso e transmissão de dados. Componentes cuja finalidade se resume à alimentação do microcontrolador, funções limitadas à alimentação condução de sinais e aspectos estruturais serão desconsideradas fundamental para o uso adequado.

5.8.1 Alimentação

Próximo à porta serial universal (USB) da placa existe um regulador de tensão de 5 V, de modo que variações na tensão de entrada são ajustadas para o valor de tensão citado. Normalmente utiliza-se um regulador de tensão com grande tamanho físico quando comparado aos demais componentes da placa, com o intuito de favorecer à dissipação de calor gerado na regulação da tensão (Monk, 2010).

5.8.2 Conectores

Ao observar os cantos inferior e superior da placa, os quais são mostrados na Figura 9, é possível identificar conectores do tipo "fêmea" para ligação do Arduino com outros periféricos.

Figura 9 - Conectores do Arduino.



Fonte: DESTACOM,2012.

5.8.3 Funções dos conectores

A seguir são descritas as funções de cada conector:

RESET: Quando energizado em 5 V, reinicializa o microcontrolador.

GND: Conector com tensão 0 V (aterramento), usado como referência de tensão nula para os demais conectores.

3V3 (3,3 V), 5 V, Vin: Conectores de tensão, utilizados para levar alimentação aos módulos conectados à placa.

Entradas Analógicas (de 0 a 5): Conectores de entrada que recebem sinais analógicos ligados a cada um deles.

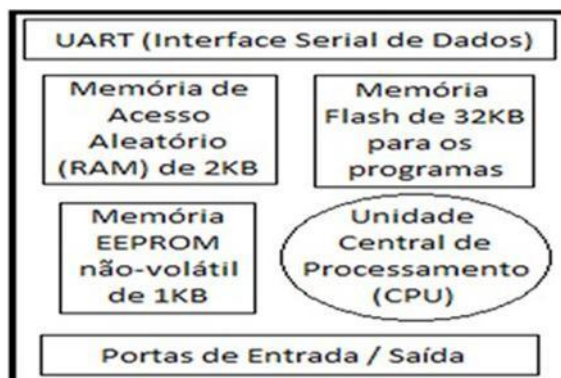
A corrente que circulará por eles é pequena, uma vez que a resistência interna destes circuitos é grande, resultando em informações que poderão ser utilizadas de acordo com o projeto no qual o Arduino for aplicado.

Conexões Digitais (0 a 13, GND e AREF): As conexões digitais 0 a 13 são conexões de entrada e saída. Especificamente as conexões 0 e 1 possuem a indicação RX e TX respectivamente, que são conexões seriais de recepção e transmissão de dados.

As conexões digitais numeradas do Arduino geralmente operam com tensões absolutas de 0 V ou 5 V. A conexão GND atua como aterramento (referência 0 V). A AREF pode ser usada para medições analógicas precisas, utilizando a tensão de alimentação do Arduino (tipicamente 5 V) como referência. Especificamente, as conexões digitais 3, 5, 6, 9, 10 e 11 são PWM (Pulse Width Modulation) e podem fornecer uma tensão de saída variável, ao contrário das demais conexões digitais que oferecem apenas 0 V ou 5 V. O design da placa Arduino facilita o acesso de dispositivos externos aos terminais do microcontrolador. Este microcontrolador, considerado o "coração do Arduino" por Monk (2010), é fixado em um soquete DIL, o que simplifica sua remoção e substituição. Ele controla todo o funcionamento do dispositivo e pode ser programado para definir todas as tensões de entrada, saída e a lógica dos sinais em seus terminais, permitindo sua aplicação em diversas finalidades.

A Figura 10 mostra o diagrama de blocos do microcontrolador ATmega328, bem como todos os seus componentes.

Figura 10 - Conectores do Arduino.



Fonte: DESTACOM,2012.

Internamente o microcontrolador possui uma memória EEPROM (memória de somente-leitura programável e eletricamente apagável), utilizada para a gravação de dados como instruções de programas, sem o risco de que estes dados sejam perdidos no caso de reinicialização (através da função RESET descrita anteriormente, ou por meio do botão de mesmo nome, localizado na placa), ou falha de alimentação.

Além disso, o microcontrolador também apresenta uma memória RAM (memória de acesso aleatório) que registra as informações geradas durante o uso do dispositivo e que são eliminadas sempre que há a reinicialização ou desligamento do equipamento.

5.8.4 Oscilador

Outro componente fundamental para o funcionamento do Arduino localiza-se próximo ao botão RESET. oscilador formado por um cristal de quartzo que ao vibrar a 16 MHz (Monk, 2010) e gera um campo elétrico senoidal (Matos, Tavares,1998), controla a impedância entre os elétrodos e determina a frequências de operação do microcontrolador.

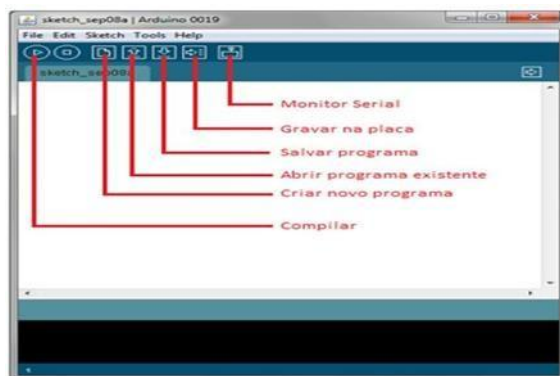
5.8.5 Interface USB

A interface USB, da placa arduino permite conexão com computadores para programação e troca de dados. Essa comunicação é viabilizada por um chip ATMEL (ATmega8U2 ou ATmega16U2), que convertem sinais para o arduino (ATMEL, 2012), o firmware desses chips pode ser reprogramado para que o computador identifique a placa como outro periférico (McRoberts,2010).

5.9 Software

O software do arduino, essencial para a lógica de seus sinais elétricos, consiste em programas desenvolvidos em um ambiente de desenvolvimento integrado (IDE) no computador (Margolis,2011), esses programas são convertidos em instruções para o microcontrolador do arduino, permitindo a gravação, execução e controle de todas as suas operações. microprocessador. Em outras palavras, tudo o que será realizado pelo Arduino, seja através do próprio microcontrolador ou de periféricos a ele conectados, deverá ser programado em scripts inseridos por intermédio da (IDE), onde são definidas as orientações necessárias para cada execução. A interface da IDE do Arduino é intuitiva. No entanto, como depende da programação de comandos e execuções a mesma exige do usuário certos conhecimentos de algoritmos e linguagem de programação, em geral, C ou C++. A Figura 11 mostra a interface inicial da IDE do Arduino.

Figura 11 - Arduino IDE Tela Principal do Arduino.



Fonte: MARGOLIS,2011.

Através da IDE é possível projetar, alterar, compilar e gravar programas que comandam o Arduino e qualquer Shield ou módulos a ele conectados. A IDE do Arduino traz diversos projetos exemplo prontos para serem executados. O programador pode utilizar estes projetos, bem como alterá-los ou usá-los como base para a criação de novos programas.

5.10 Desenvolvimento para novas aplicações para o Arduino

O potencial do Arduino para novas aplicações, destacando a criatividade do usuário como fator chave para sua utilização, superando barreiras de conhecimento técnico em eletrônica ou em programação. conforme Margolis (2011), o Arduino possibilita a criação de dispositivos que interagem com o ambiente, respondendo a luzes, emitindo sons e realizando movimentos. Suas aplicações são vastas, indo de automação residencial, visando aprimorar o conforto e a qualidade de vida.

5.11 MATERIAIS E MÉTODOS

5.11.1 Método da Pesquisa

O desenvolvimento teórico tem como base os conhecimentos obtidos durante o curso e pesquisas relacionadas ao tema, fazendo com que aplicação prática na linha de produção traga os resultados dos materiais utilizados e das dificuldades encontradas para implementar o projeto.

O trabalho tem como objetivo aplicar técnicas de controle em um sistema de dosagem de líquidos, visando realizar a mistura homogênea, utilizando um controlador programado por meio de uma linguagem de programação, dentre desse objetivo aplicar conceitos e técnicas de automatização as possibilidades de manuseio de circuitos eletrônicos, eletropneumáticos no projeto serão estudados e efetuados os métodos e meios para controle deste sistema.

Será realizado o controle de dosagem do atuador por velocidade e tempo programado, através de programação via Arduino com aplicação de software, hardware e placas de interfaces de atuadores eletrônicos, será interagido com dispositivos de comunicação via portas lógicas e digital, para controle do fluxo do líquido (no caso água) que possa possibilitar ao usuário definir os valores de set point por meio de software possibilitando a comunicação e controle via internet das coisas. As aplicações dos métodos de controle referente aos movimentos rotativos ou lineares por controle de tempo e velocidade com objetivo de realizar a dosagem de líquidos não inflamáveis com aplicação das misturas dos líquidos envolvidos no processo para realizar a Homogeneização do líquido a ser adquirido para uso no processo, no caso em questão mistura de líquido silicone industrial, mistura de tintas, essas aplicações será realizado ensaios para possíveis variáveis, com possibilidade de monitorar as grandezas envolvidas no processo. A plataforma é composta por circuito impresso com microcontrolador interface de programação permitindo a criação de códigos para controlar os componentes eletrônicos conectados a placa. O trabalho consiste em substituir o trabalho manual aplicando melhorias no processo, substituir um trabalho braçal por outro com ênfase na capacidade mental do funcionário.

O controle bases se no princípio da realimentação obtiva no controle de determinadas variáveis de um sistema com aplicações de técnicas de controle com a instrumentação, hardware para medição e controle aduado melhorando agilidade, melhoria no processo, confiabilidade, diminuído custo e garantia da qualidade.

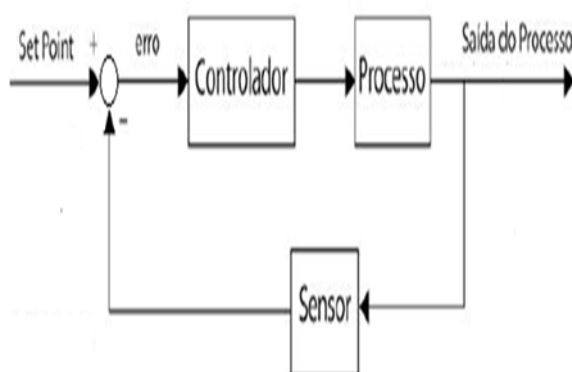
A centralização das variáveis do processo possibilita a fabricação de produtos que seria impossível por meio do controle manual para atingir o nível de produção que o processo envolvido e qualidade almeja alcançar. O Arduino foi determinante para a escolha na aplicação do projeto devido a inúmeras gamas de possibilidades e comunicação com os dispositivos que será usado no projeto.

O Arduino e prototipagem e acessibilidade as placas de baixo custo a programação simples o que possibilita a configuração não sendo necessário conhecimentos avançados em eletrônica proporcionando desenvolve protótipos projetos em um curto período a flexibilidade para se desenvolver diferentes configurações de hardware e software.

Exigindo interações rápidas para desenvolvimento e testa diferentes abordagens e modificar o protótipo rapidamente economizando tempo e recursos. Software para escrever e copilar, carregar códigos para a placa. O Arduino IDE e gratuito e está disponível em plataformas digitais, embora haja desafios com relação a aplicação do Arduino a plataforma apresenta vantagens com relação a outras plataformas nesse contexto de sistemas de automação o baixo custo e flexibilidade para uso com relação a outros sistemas de automatização de processos industriais, com sua capacidade de conexão a variedade de dispositivos e entrada e saída através de módulos (placas de expansão) customizadas e adaptadas as necessidades especificas cada processo sem a necessidade de investimentos altos e sistemas complexos. A interação de dispositivos físicos a internet permitindo que eles colem troquem dados de maneira inteligente. O Arduino e fundamental nesse contexto pois sua plataforma permiti desenvolver dispositivos IoT, por sua versatilidade capaz de comunicar entre si com a internet, conectados a placa Arduino realizando a coleta de dados em tempo real e os envia para nuvem podendo monitorar os dados obtidos utilizando aplicativo ou interfaces web com a plataforma flexível podendo ser adaptada ao processo em desenvolvimento proporcionando melhoria , controle de acesso do processo eficiência e qualidade, flexibilidade e personalização do processo.

Com o uso de técnicas de controle, é possível aliar duas ou mais áreas para criar um sistema automatizado com capacidade de avaliar e corrigir seus próprios desvios de maneira inteligente e comportamental. É possível englobar diversos componentes e métodos para gerar um único sistema capaz de ser autossuficiente. A Figura 12 exemplifica o sistema de controle.

Figura 12 - Sistema de controle em malha fechada.

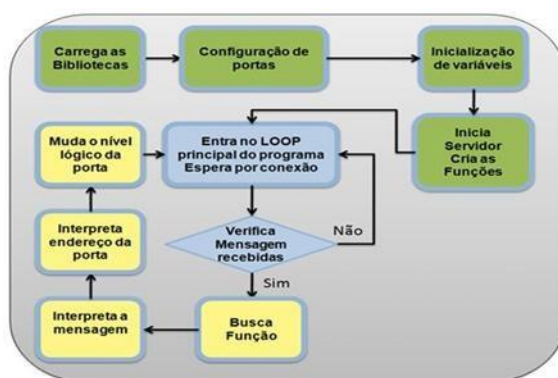


Fonte: VEDRAMINI,2018.

5.11.2 Diagrama de bloco Arduino

O sistema Arduino apresenta dois componentes básicos: a placa Arduino, que é um elemento físico (hardware) utilizado para construir seus objetos, e a IDE que é um programa de computador onde é gerado o código que será executado na placa Arduino (BANZI, 2015). Na Figura 13 é apresentado um diagrama de blocos genérico que apresenta os principais passos na programação do Arduino.

Figura 13 - Passos na programação do Arduino.



Fonte: Lucidcha, 2025.

A base do Arduino é o microcontrolador AVR da Atmel e existem diversas como o ATmega8 e o ATmega256 de 256k de memória de programa Flash. As principais diferenças dos Arduino comparando com o microcontrolador que os integram, são suas derivações, que é mostrada as variedades de microcontroladores que podem acompanhar o Arduino (Monk, 2013).

5.11.3 Arduino relés 5 V módulo 8 canais

Placa chaveadora com 8 relés. Funciona com 5 V, a tensão mais usada em projetos digitais, com 8 saídas inversoras de alta corrente, completamente independentes e isoladas (10A). As entradas são optoacopladas, requerem baixíssima corrente para o acionamento, diretamente compatível com a maioria dos microcontroladores. (Arduino, 8051, AVR, PIC, DSP, ARM, ARM, MSP430, lógica TTL).

A Figura 14 apresenta o Arduino Relés 5 V módulo 8 Canais.

Figura 14 - Módulo Arduino relés.



Fonte: codeproject,2025.

A Tabela 1 apresenta as características elétricas do Arduino Relé 5 V.

Especificações:	
Canais	
Tensão de funcionamento	5V
Tenssai de carga dos Relês	Corrente de saída:10 ^a (250 AC – 30V DC 28V DC

Fonte: Autores, 2025.

5.11.4 Bombas dosadora

Bombas dosadoras são bombas de deslocamento positivo, possuem uma ou mais câmaras em cujo interior o movimento de um órgão propulsor comunica energia de pressão ao líquido, provocando o seu deslocamento. Proporciona assim as condições para que se realize o escoamento na tubulação de aspiração até a bomba e na tubulação de recalque até o ponto de utilização. Podem ser classificadas como bombas de Descarga Controlada, deslocam com precisão um determinado volume de líquido em um tempo pré-estabelecido. São acionadas por motores, usando em geral mecanismos do tipo eixo de manivela-biela. A bomba utilizada no desenvolvimento do projeto é a microbombas dosadora de engrenagens com acoplamento magnético marca micro pump, com acionamento através de motor elétrico.

Deslocamento: 0,316 ml/revolução

Máxima Pressão Diferencial: 5,6 bar (80 psi)

Máxima Pressão do Sistema: 21 bar (300 psi)

Temperatura: -46 a 121°C

Viscosidade: 0,2 a 1500 cPs

Velocidade Máxima: 10000 RPM

Peso: 0,45 kg

A vazão só pode ser alterada se houver meios para modificar o volume deslocado ou a velocidade de rotação. No caso da variação da rotação, a vazão e a potência variarão na proporção direta da rotação, apresentando assim uma mudança de comportamento da funcionalidade do equipamento. O comportamento do equipamento é diretamente influenciado pela variação de rotação, vazão e potência de um conjunto motobomba dosador que apresenta duas entradas: velocidade de rotação e potência, e uma saída: essas três variáveis são independentes, a alteração em uma delas vazão. Provoca uma variação proporcional nas outras duas, resultando em uma mudança no comportamento funcional do equipamento. O sistema possui duas entradas (velocidade de rotação e potência) e uma saída (vazão).um controlador integrado permite gerencia até quatro bombas simultaneamente. Estas bombas podem operar em quatro regimes de trabalho distintos ou serem desativadas, suportando um fluxo contínuo. À bomba operará indefinidamente em uma velocidade programada, ou seja, enquanto a receita estiver em operação, à bomba estará dosando com uma vazão fixa, não ocorrendo variação na velocidade de rotação do motor elétrico. A bomba operará de maneira intermitente, ou seja, vai operar em ciclos com tempo de operação e tempo de intervalo predefinidos.

A bomba ilustrada na Figura 16 inicia sua operação quando a receita é ativada e se desliga após um tempo pré-estabelecido, trabalhando com a vazão solicitada (programada). Neste processo de dosagem de líquidos serão utilizadas cinco bombas.

Figura 16 - Bombas: Dosagem.



Fonte: Autores, 2025.

Cada bomba irá despejar um específico pigmento, e deve ser acionada, assim como sua vazão controlada, até que atinja o volume necessário. Fonte de alimentação. A tarefa básica de uma fonte de alimentação é converter a tensão da rede elétrica alternada para a tensão de alimentação dos circuitos eletrônicos, geralmente +5, +12 e +24 VCC para a comunicação com o programador ou computador. A Figura 17 mostra um modelo de fonte de alimentação que será utilizado no projeto.

Figura 17 - Fonte de alimentação.



Fonte: BRAGA ,2017.

5.11.5 Sensor de pressão

Os sensores de pressão são dispositivos utilizados para medir a pressão em um determinado ambiente. Eles funcionam por meio de uma membrana sensível que é deformada pela pressão, gerando uma mudança na resistência elétrica, capacitância ou outra propriedade elétrica do sensor.

Essa mudança é então convertida em um sinal elétrico que pode ser processado e interpretado para determinar a pressão (Cassiolato, 2015). Eles são utilizados para medir a pressão de fluidos em sistemas hidráulicos, pneumáticos e de refrigeração, bem como para monitorar a pressão atmosférica em aplicações meteorológicas e ambientais. (BOJORGE, 2018). A Figura 18 ilustra um sensor de pressão.

Figura 18 - Sensor de pressão.



Fonte: cassiolato, 2015.

5.11.6 Sensor Nível

O modelo mais indicado para determinação de um nível fixo é o sensor chave de nível boia, por possuir um tamanho pequeno, alta confiabilidade, e capacidade de comutação, que pode ser obtida com o auxílio de uma microchave (*micro switch*) (OMAZINI, 2009). Porém também existem os sensores de nível de líquidos com haste vertical boia é um dispositivo utilizado para medir o nível de líquidos em tanques ou reservatórios. Ele consiste em uma haste vertical que é instalada dentro do recipiente, com uma boia que flutua na superfície do líquido. Conforme o nível do líquido sobe ou desce, a boia se move para cima ou para baixo na haste, transmitindo essa informação para o sensor.

A Figura 19 exemplifica o sensor de haste vertical, cujo flutuador se movimenta conforme o nível do líquido.

Figura 19 - sensor de nível tipo boia (float switch).

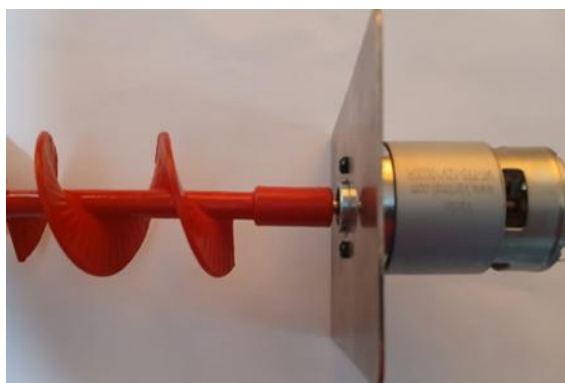


Fonte: omazini, 2009.

5.11.7 Motor Misturador

O misturador elétrico é indicado para dissolver, agitar ou homogeneizar, tinta, cola, entre outros, e com tecnologia de arranque suave, possibilitando partidas leves que evitam os respingos durante o uso do equipamento. A Figura 20 ilustra motor acoplado a uma hélice utilizados para misturar tintas.

Figura 20 - Motores de passo.



Fonte: Autores, 2025.

Essa etapa é composta pela movimentação do misturador até a posição correta, e então o seu acionamento. A movimentação é feita por um motor de reversão que, ao girar um eixo fixo com a hélice.

Após o recebimento dos pigmentos, o recipiente deve ser transferido para a etapa de homogeneização para determinar o tempo necessário para que todos os pigmentos sejam incorporados na base até obter um resultado homogêneo. A porcentagem de homogeneidade atingida.

5.12 Desenvolvimento projeto do Arduino

Diagrama de Estados para um Dosador Automático de Tinta com Arduino O sistema visa dispensar quantidades precisas de diferentes cores de tinta para criar uma mistura específica. Estado Inicial e Repouso No estado inicial, o sistema faz a transição para 'Aguardando Receita'. Este é o estado de repouso, onde o Arduino está ligado e: Aguarda a entrada de uma nova receita (ex.: via teclado, interface serial ou aplicativo). Monitores os sensores de nível dos reservatórios de tinta. Validação da Receita e Estoque. Ao ser carregada uma Receita de Dosagem (transição 'Receita Carregada'), o sistema transiciona para o estado 'Verificar Estoque e Válvulas'. Verificação de Estoque: O Arduino verifica se o volume de tinta solicitado na receita (ex.: 'Volume A' para Tinta A, 'Volume B' para Tinta B) é suportado pelo nível de tinta nos respectivos reservatórios (lendo sensores de nível ou peso). Se houver tinta suficiente para a receita ('Estoque OK para Receita'), a transição automática é para 'Dosar Tinta A'. Contudo, caso qualquer cor de tinta esteja abaixo do limite necessário para a receita ('Estoque Insuficiente'), o processo é desviado para o estado de 'Alerta de Reposição/Falha'. O sistema para, exibe a cor faltante e aguarda a reposição antes de retornar ao estado 'Aguardando Receita'. Verificação de Válvulas: Se todas as válvulas de dosagem (acionadas por solenoides ou bombas peristálticas) estiverem em suas posições de repouso (leitura de fim de curso ou feedback do atuador), o processo segue.

Caso contrário, aciona-se o estado de alarme/falha. Ciclos de Dosagem de Tinta. O sistema entra em uma sequência de estados, um para cada cor de tinta, por exemplo:

'Dosar Tinta A': O Arduino aciona a bomba/válvula correspondente à Tinta A. A dosagem é controlada por:

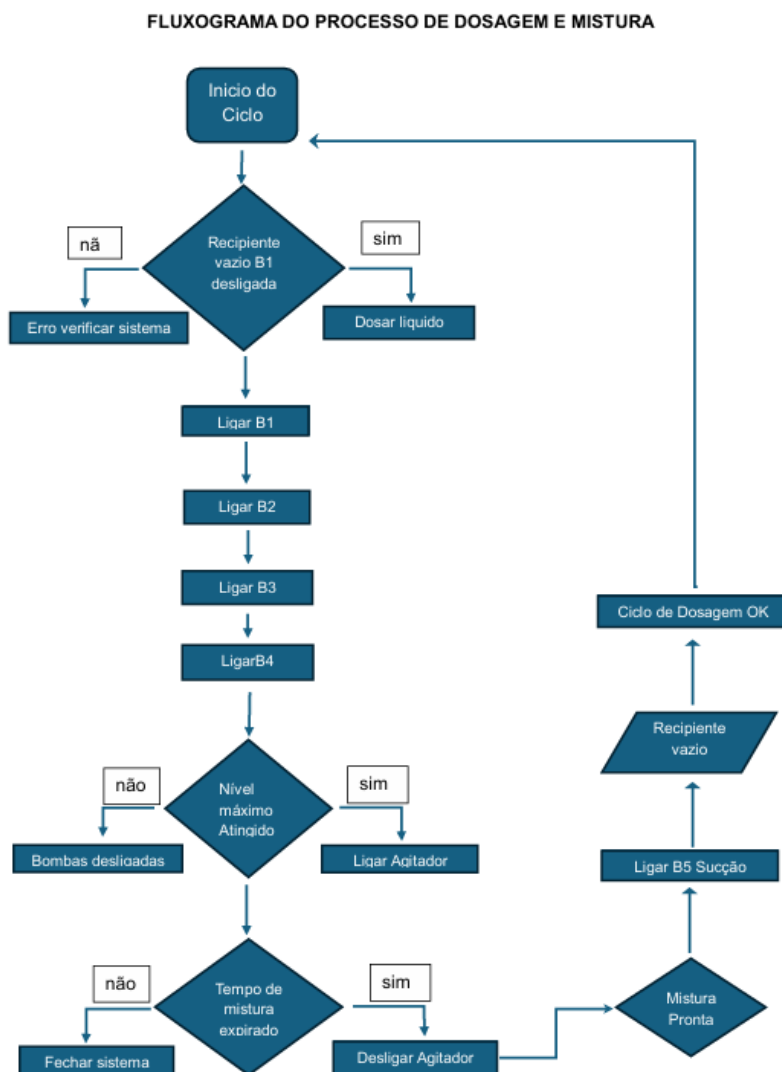
Tempo: A bomba funciona por um tempo calculado para liberar o volume necessário. Sensor de Fluxo: A bomba é acionada até que o sensor de fluxo indique o volume exato. Célula de Carga: A bomba é acionada até que a célula de carga sob o recipiente de mistura atinja o peso correspondente ao volume de Tinta A. Transição: Quando o volume desejado for alcançado ('Volume A Disp.', ou 'T1'), o sistema transiciona para 'Dosar Tinta B'. 'Dosar Tinta B': Segue a mesma lógica de controle do volume, mas para a Tinta B. Após o volume ser atingido ('Volume B Disp.', ou 'T2'), o sistema transiciona para 'Misturar'. (Os ciclos de dosagem se repetem para tinta C, D, etc., conforme a complexidade da receita.) Mistura e Finalização. 'Misturar': Após todas as tintas serem dosadas, o motor do misturador é acionado por um tempo pré-determinado (ex.: 15 segundos). O sentido de rotação pode ser controlado por uma Ponte H ou driver de motor, assim como no seu exemplo original. Transição: Ao final do tempo de mistura ('Mistura Completa', ou 'T3'), o sistema transiciona para 'Ciclo Finalizado'. 'Ciclo Finalizado': O Arduino desliga o motor misturador, limpa os contadores de volume e pode registrar o ciclo como concluído (salvando dados em um SD Card ou enviando via rede). A transição final é de volta para o estado 'Aguardando Receita', pronto para um novo pedido de dosagem.

5.12.1 Sistema de Dosagem Automática de Tinta com Arduino

O projeto apresenta um sistema automatizado de dosagem de tinta baseado em Arduino, desenvolvido para preparar misturas com alta precisão. A operação inicia-se com o posicionamento e acionamento do misturador por um motor de passo, responsável por incorporar os pigmentos à base. Após a adição das tintas, o recipiente segue para a homogeneização, etapa em que o sistema define o tempo necessário para atingir a uniformidade da mistura. O Arduino gerencia todo o ciclo por meio de um diagrama de estados, iniciando pela leitura da receita e pela verificação do nível de tinta e das válvulas de dosagem somente após validar estoque e condições de operação, o sistema libera cada cor, controlando os volumes por tempo, sensor de fluxo ou célula de carga. Concluída a dosagem, o misturador é acionado por um período pré-determinado até completar o processo. Sensores de nível e botões de comando garantem segurança e controle. No fim, o sistema registra o ciclo e retorna ao estado de prontidão.

O projeto proporciona precisão, repetitividade e eficiência para a produção automatizada de tintas. Na figura 21 vemos como é o processo de dosagem de líquidos.

Figura 21 - Fluxograma do processo.



Fonte: Autores, 2025.

5.13 DESCRIÇÃO E IDENTIFICAÇÃO DOS COMPONENTES

5.13.1 Lista de peças referente a montagem do dosador

Tabela de Relação de Componentes

Unid.	Componentes	Função	Qtde	Obs
1	Arduino R3 328	Controle central do sistema	1	Alimentação por 5v programada via USB
2	Módulo relê 8 canais	Acionando bombas e motor	1	Alimentação, 12v isolado por optoacoplador
3	LCD 16x2 DC	Exibição do Status do processo	1	Endereço padrão Dx27; usa pino A4 (SDA) E A5 (SCL)
4	Bombas RN 385	Dosagem de Reagentes e Seção do líquido	5	Motor DC, 12v; consumo médio 300mA cada
5	Motor de mistura	Mistura de Reagentes do tanque	1	Motor DC 12; acionado por relê
6	Sensor de nível	Detector quando tanque esta cheio(presença 1)	1	Sensor capacitivo ligado ao pino A0
7	Sensor de nível baixo	Detector quando tanque esta vazio (pesença 0)	1	Sensor capacitivo ligado ao pino A0
8	Botão Start	Iniciar o ciclo automático	1	Entrada digital com resistor pull-up interno
9	Chave seletora M/A	Alternado entre modo manual e automático	1	Entrada digital ; leitura no pino D9
10	Fonte 12 v; chaveada	Alimentação das bobinas com relés	1	Corrente mínima recomendada; 2A
11	Fonte 5v ou regulador	Alimentação do Arduino	1	Pode ser regulador LM 7805 pou fonte fonte USB
12	Jumpers terminasi	Conexão entr e módulos	1	Usar fios de qualidades e terminais crimpados
13	Caixa de painel	Montagem físicas dos componentes	1	Preferencialmente com tampa termoplásticos

Fonte: Autores, 2025.

5.13.2 Lista de peças referente a montagem do dosador

Tabela de Relação de Componentes

Item	COMPONENTES UTILIZADOS	
	COMPONENTE	FUNÇÃO
1	Arduino Uno	Controlador principal do sistema; gera o sinal PWM e faz a leitura do potenciômetro.
2	Display LCD 16x2 (I2C)	Exibe a velocidade do motor e status do sistema.
3	Módulo Relé 5V	Faz a comutação elétrica do motor, permitindo ligar ou desligar a carga.
4	Módulo PWM	Controla a largura do pulso que alimenta o motor, regulando a rotação.
5	Motor DC escovado (5–12 V)	Carga principal controlada pelo sistema.
6	Potenciômetro (10 kΩ)	Permite o ajuste manual da velocidade do motor.
7	Fusível 10 A com porta-fusível	Protege o circuito contra sobrecorrente.
8	Fonte de alimentação 12 V	Fornece energia ao circuito e ao motor.
9	Cabos de conexão / jumpers / bornes	Realizam as ligações elétricas entre os módulos.

Fonte: Autores, 2025.

5.13.3 Posicionamento de montagem dos pinos do Arduino Uno.

Tabela de Relação de Componentes

Pino do Arduino	Componentes conectados	Propósito
Pino Digital D2	Sensor de nível (pino de saída)	Estado do Recipiente (cheio/vazio)
D4	Módulo Relé J1N 1	Controle da bomba dosadora 1
D5	Módulo Relé-JN2	Controle de bomba dosadora 2
D6	Módulo Relé-JN3	Controle de bomba dosadora 3
D7	Módulo Relé-JN4	Controle de bomba dosadora 4

D8	Módulo Relé-JN5	Controle de bomba dosadora 5
D9	Módulo Relé-JN6	Controle de bomba dosadora 6
D10	Botão Iniciar Selecionar	Iniciar ciclo/selecionar(opção no manu)
D11	Botão aumentar (+) (φ) parar	Aumentar valor/parada de emergência
D12	Botão diminuir (-)	Diminuir valor/parada de emergência
Pinos analógicos (T2C)	X	x
A4(SDA)	Display LCD 12C(Pino SDA)	Linhas de dados para comunicações 12C
A4(SCL)	Display LCD 12C(Pino SCL)	Linhas de dados para comunicações 12C
Pinos de alimentação	X	x
Sv	Módulo Relé(VCC)	Alimentação da lógica do módulo Relé
Sv	Display LCD J2C (VCC)	Alimentação do Display
Sv	Sensor de nível (VCC)	Alimentação do sensor
Sv	Botões (Via resistor de pull-down)	Alimentações dos botões
GND	Módulo Relé (GND)	Referência de terra comum
GND	Display LCD J2C(GND)	Referência de terra comum
GND	Sensor de nível(GND)	Referência de terra comum
GND	Botão (via resistor de pull-down)	Referência de terra comum
GND	Fonte de Alimentação 12v (negativo)	Importante: conectar o terra da fonte ao terra do Arduino

Fonte: Autores, 2025.

5.13.4 LIGAÇÕES DO MÓDULO RELÉ E ATUADORES

Alimentações do módulo Relé:

Conecte o pino VCC do módulo relé ao pino 5v do Arduino. Conecte o pino GND do módulo relé ao GND do Arduino.

Jumper JD-VCC: Se o módulo relé tiver um Jumper.

JD-VCC, remova-o, e conecte o pino JD-VCC a uma fonte externa de 5v, se disponível, ou ao pino 5v, do Arduino, isso isola opticamente o Arduino da alimentação dos reles, o que é uma boa prática.

Conexão dos atuadores (Bombas e Motor)

Quando positivo (+) da fonte 12v; conecte este polo ao positivo comum (C) de cada um dos 6 relés que estas utilizados (Relé 1 a 6). Podendo ser uma parte (Jumpers) entre os pinos, C, de cada relé.

Fio positivo (+) de cada atuador conecte o fio positivo de cada bomba/motor ao pino normalmente aberto (NA ou NO) do respectivo relé.

Bomba 1(+) ->

Relé 1(NO)

Bomba 2(+) ->

Relé 2(NO)

Bomba 3(+) ->

Relé 3(NO)

Bomba 4(+) ->

Relé 4(NO) Motor

5(+) -> Relé 5(NO)

Bomba Sucção -> (+) relé -> 6(NO)

Polo Negativo (-) de cada atuador:

Conecte todos os fios negativos das bombas e do motor juntos. Em seguida conecte este ponto comum ao polo negativo (-) da fonte de 12v.

5.13.5 Tabela de Custos dos Componentes eletrônica do dosador.

Custos por peças

	Componentes	Qtd e	P,Unitário	Custo total	Fonte fornecedora
1	Arduino uno R3-328	1	39,90	39,90	Mercado Livre
2	Módulo relè, 8 canais	1	35,00	35,00	Mercado Livre
3	LCD 16x2; 12C	1	19,00	19,00	Mercado Livre
4	Bombas RN 385	5	18,00	90,00	Mercado Livre
5	Motor Misturador; 12v	1	25,00	25,00	Mercado Livre
6	Sensor Nível Capacitivo	2	22,00	44,00	Mercado Livre
7	Botão Start	4	3,00	12,00	Mercado Livre
8	Chave seletora N/A	1	6,00	6,00	Mercado Livre
9	Fonte , chaveada; 12V; 2A	1	30,00	30,00	Mercado Livre
10	Fonte; 5V, ou LM7805	1	12,00	12,00	Mercado Livre
11	Caixa do painel	1	40,00	40,00	Mercado Livre
12	Jumpers e terminais	1	40,00	40,00	Mercado Livre
13	Pote Hermético	1	20,00	20,00	Mercado Livre
14	Caixa/PVC, Montagem Elétrica	1	134,00	134,00	Mercado Livre
15	Mini Motor, 12V,duplo rolamento	1	79,00	79,00	Mercado Livre
16	PWM, 5V/12V,10A	1	19,00	19,00	Mercado Livre
17	Display LCD, 16x2 Mod,I12C	1	32,00	32,00	Mercado Livre
18	Canaleta,20x20 EBS(PVC)	1	25,00	25,00	Mercado Livre
19	Garrafas/acrílico	5	22,40	112,00	Mercado Livre
20	Leds Piloto, 12V	10	10,80	129,60	Mercado Livre
21	Suporte L, Mutiuso	1	42,00	42,00	Mercado Livre
22	Botão/250Vac,3ª, Emergência	1	30,00	30,00	Mercado Livre

23	Chave Gangorra Liga/desliga, 4 polos 110/220V	1	12,00	12,00	Mercado Livre
24	Organizador de Fios E cabos espiral,6mm	5m	5,00	25,00	Mercado Livre
25	Chave Interruptor, Botão 4 polos,liga e desliga	1	19,00	19,00	Mercado Livre
26	Porta fusivel regulador para painel	1	9,90	9,90	Mercado Livre
27	Fonte chaveada colmeia,120W, 12V,10A	1	32,00	32,00	Mercado Livre
28	Sensor de nível,líquido sem contatto	2	78,00	157,80	Mercado Livre
29	Kit cabo jamper,macho/macho+ Macho/femea,120x	1	29,00	29,00	Mercado Livre
30	Kit cabo Jamper,macho/macho,40 Unid, 20cm,arduino	1	17,00	17,00	Mercado Livre
31	Fonte estabilizadora, 12V x 1A, B,volt,arduino	1	22,00	22,00	Mercado Livre
32	Case slim Black Abs prato, Arduino Uno R3	1	19,00	19,00	Mercado Livre
33	Bloco de Relés,Modulo Relés,8 canais, 5V Arduino	1	43,00	43,00	Mercado Livre
34	Bloco compativel com Arduino Uno R3 Snd, Atmega 308 + cabo usb	1	122,0 0		Mercado Livre
35	Mini bomba D'agua,12V	6	37,90	227,40	Mercado Livre
TOTAL				1.749,80	

Fonte: Autores, 2025.

5.14 METODOLOGIA TÉCNICA E APLICAÇÃO DO PROJETO

A metodologia técnica aplicada neste projeto visa o desenvolvimento de um sistema automatizado de dosagem e mistura de líquidos, baseado na plataforma Arduino. O objetivo principal é permitir o controle automatizado do processo por meio do gerenciamento da velocidade dos atuadores, garantindo precisão na dosagem e eficiência na mistura. O sistema também integra sensores e comunicação digital para monitoramento e controle em tempo real.

Abordagem Metodológica:

A metodologia segue um ciclo iterativo de desenvolvimento com as seguintes etapas: análise de requisitos, projeto (hardware e software), montagem, testes e validação.

Esta abordagem garante robustez, precisão e confiabilidade ao sistema.

Análise de Requisitos Nesta fase, foram definidos:

As funcionalidades desejadas do sistema;

Parâmetros de desempenho;

Interfaces de hardware e software.

Projeto de Hardware:

A seleção de componentes priorizou flexibilidade, custo-benefício e facilidade de integração. Os principais elementos são:

Arduino Uno ou equivalente: microcontrolador principal;

Módulo de Relés (5V, 8 canais DC): acionamento de bombas e motor;

Bombas dosadoras (RS-395 DC): para controle preciso de líquidos;

Motor misturador (DC ou passo): acionamento do sistema de agitação;

Sensor de nível (XKC-Y25-V DC): monitoramento sem contato;

Sensor de pressão: verificação de fluxo e detecção de falhas;

Fonte de alimentação chaveada 12 v DC: fornece energia estável para todos os componentes;

Interface com usuário: botões para controle, parada de emergência, display LCD para ajustes e feedback via Serial Monitor.

Projeto de Software

O software foi desenvolvido na IDE do Arduino, utilizando C/C++. As funcionalidades

principais incluem:

Controle de dosagem via PWM nas bombas;

Controle do motor misturador (tempo e velocidade); Leitura e interpretação dos sensores de nível e pressão;

Lógica de controle integrada com resposta automática a eventos;

Comunicação serial para feedback e ajustes de parâmetros;

Implementação de falhas e paradas de emergência. Montagem e Prototipagem:

A montagem ocorreu em ambiente laboratorial, seguindo as etapas:

Montagem Mecânica:

Inclui a fixação das bombas, motor, recipientes e demais componentes em estrutura física adequada ao protótipo.

Conexões Elétricas

Será feito de acordo com um diagrama elétrico detalhado, garantindo polaridade e corrente compatíveis com os componentes. Ainda faremos os testes de todas as conexões para continuidade e funcionamento básico.

Testes e Validação:

A etapa de testes foi dividida em fases, assegurando o correto funcionamento e a confiabilidade do sistema.

Testes Unitários:

Cada componente foi testado isoladamente:

Bombas;

Motor;

Sensores;

Relés;

Interface de usuário;

Testes de Integração;

Verificou-se a comunicação entre os módulos e a atuação coordenada dos atuadores com base nas leituras dos sensores.

Testes de Desempenho: Medições de:

Precisão da dosagem;

Tempo e qualidade da mistura;

Estabilidade do sistema em operação contínua.

5.14.1 Calibração E Teste Será Verificado Durante O Ensaio

As bombas serão calibradas no ensaio para garantir precisão no volume dispensado. Esse processo envolveu a medição do volume em tempo definido e ajustes no código de controle.

- Testes de Robustez:

Simulações de falhas foram conduzidas para validar a resiliência do sistema, incluindo:

- Detecção de recipiente vazio;
- Obstruções no fluxo;
- Resposta de emergência automática.
- Ambiente de Desenvolvimento
- **IDE Arduino:** utilizada para codificação, compilação e gravação no microcontrolador.
- **Serial Monitor:** ferramenta de depuração e monitoramento em tempo real.
- **EEPROM ou cartão SD (opcional):** para armazenamento de dados e parâmetros configuráveis.
- **FluidSIM** (opcional): para simulação de processos e controle.

Realizado a montagem mecânica e a fixação das mangueiras, e bicos dosadores dos reservatórios, (**figura 22**) e soldagem dos cabos positivos e negativos das microbombas.

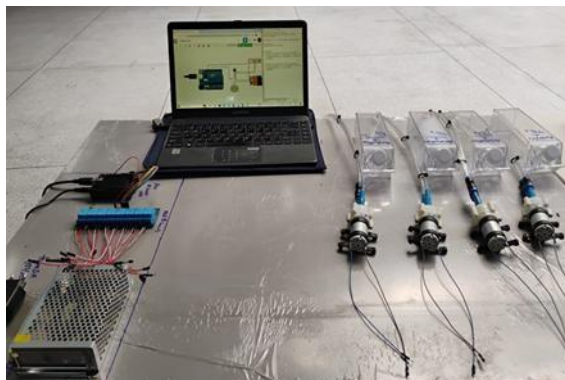
Figura 22 - Reservatório e Bombas.



Fonte: Autores, 2025.

Teste do simulador, (Figura 23), de uma microbomba do software pelo Tinkercard.

Figura 23 - Simulação Tinkecard.



Fonte: Autores, 2025.

5.15 Hardware; montagem dos componentes

5.15.1 Identificação dos componentes

Nas figuras 24,25,26 e 27, mostrados abaixo na sua perspectiva física:
Sensores ou válvulas solenoides (parte superior esquerda) — utilizados para controlar passagem de fluido ou ar, dependendo do projeto.

Placa Arduino Uno e cabo USB (parte inferior esquerda) — microcontrolador responsável por processar os sinais e comandar os atuadores.

Módulo de relés (parte inferior direita) — serve para isolar eletricamente e permitir o acionamento de cargas de maior potência (motores, lâmpadas, válvulas).

Montagem parcial (parte superior direita) — mostra a conexão entre o Arduino, o módulo de relés e os atuadores, com cabos de alimentação e controle.

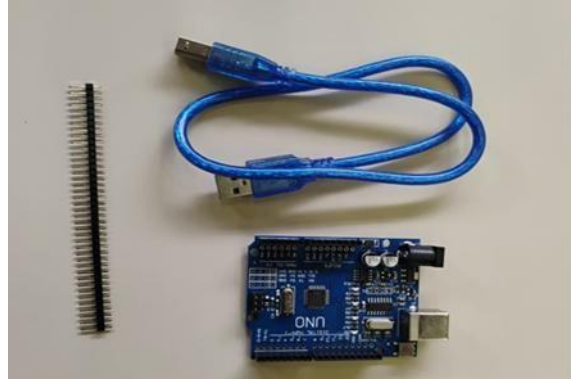
Nas figuras 24,25,26 e 27, Sensores ou válvulas solenoides, perspectiva física:

Figura 24 - Válvulas.



Fonte: Autores, 2025.

Figura 25 - Arduino.



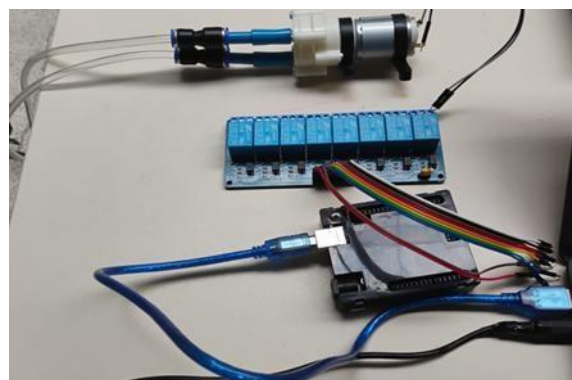
Fonte: Autores, 2025.

Figura 26 - Relés



Fonte: Autores, 2025.

Figura 27 - Simulação com o relé



Fonte: Autores, 2025.

5.16 PRINCÍPIO DE FUNCIONAMENTO DO SENSOR NÍVEL

Sensores de nível capacitivos, Figura 28 operam pela variação da capacitância entre dois eletrodos, causada pela mudança no material dielétrico. Quando o nível de líquido muda, altera-se a capacitância, que é convertida em sinal elétrico. No sensor XKC-Y25-V, mesmo sem contato direto, a aproximação do líquido aumenta a capacitância, ativando a saída de sinal.

Descrição do Sensor XKC-Y25-V

O XKC-Y25-V é um sensor de nível sem contato, que detecta líquidos através de recipientes não metálicos como vidro, plástico ou cerâmica.

Figura 28 - sensor de nível.



Fonte: Autores, 2025.

Características principais:

Alimentação: 5 V a 24 V DC;

Saída digital (ex: PNP/NPN);

Detecção até cerca de 20 mm (parede + líquido);

Sem partes móveis, com alta durabilidade;

É ideal para aplicações onde não se deseja contato com o líquido nem perfuração do recipiente.

Vantagens e Limitações Vantagens:

Detecção sem contato direto;

Alta sensibilidade;

Fácil instalação;

Ampla faixa de alimentação;

Limitações:

Recipientes metálicos bloqueiam o campo elétrico;

A precisão depende das propriedades dielétricas do líquido limite de detecção de até 20 mm.

Aplicações Típicas:

Monitoramento de nível em aquários, caixas d'água, máquinas de café, impressoras 3D

Indústrias alimentícia e química, onde o sensor não deve tocar o líquido;

Situações que exigem sinal de nível (cheio/vazio), sem medição contínua;

Instalação e Uso:

Fixar externamente ao recipiente na altura desejada;

Usar somente em superfícies não metálicas (ou adaptar);

Conectar alimentação e sinal de saída ao sistema de controle;

Ajustar sensibilidade se o modelo permitir.

5.16.1 RELATÓRIO TÉCNICO DO FUNCIONAMENTO DOS SENSORES (MODELO- XKC-Y25-V)

Projeto de Controle de Nível e Agitação com Arduino

Conexões comuns dos sensores XKC-Y25-V

Cor do fio	Função elétrica	Ligação
Marron	Alimentação + 12 DC	Fonte 12 V
Azul	Terra (GND)	GND comum (Fonte + Arduino)
Preto	Saida digital principal	Arduino A0 (nível alto)
Amarelo	Saida digital auxiliar	Arduino A1 (nível baixo)
Amarelo	Saida digital auxiliar	Arduino A1 (nível baixo)

Componentes principais

Arduino UNO / Nano – unidade de controle. Sensores de nível XKC-Y25-V (2 unidades):

S1 → nível

alto (A0) S2

→ nível baixo

(A1)

Atuadores:

Motor DC → agitação.

Bomba M5 → enchimento/sucção.

Módulo relé 8 canais – controla motor e bombas.

Fonte chaveada 12 V / 10 A – alimenta sensores, relés e Arduino.

5.17 Botões e LEDs:

K1 → liga/desliga geral.

K2 → emergência (corta 12 V).

HL1–HL3 → sinalização (ligado, operação, emergência).

Interligações principais

Componente	Conexão	Observação
Fonte PS1	+12 V / GND → sensores, relés e Arduino	Fonte principal
S1 (nível alto)	Preto → A0	Detecta tanque cheio
S2 (nível baixo)	Amarelo → A1	Detecta tanque vazio
Relés IN1–IN6	D2–D7	Acionamento das bombas e motor
Relé VCC/GND	+5 V / GND do Arduino	Alimentação lógica
Arduino Vin	+12 V da fonte	Alimentação via regulador interno

Lógica de funcionamento

S2 (nível baixo) detecta tanque vazio → Arduino ativa bombas 1–4 (D2– D5) para enchimento/dosagem.

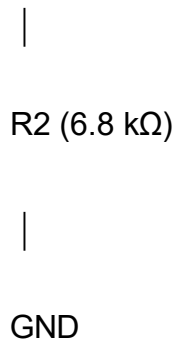
S1 (nível alto) detecta tanque cheio → Arduino aciona motor de agitação (D6) por tempo determinado.

Após a agitação, bomba 5 (D7) é ligada para sucção/esvaziamento.

Botão K2 (emergência) corta os 12 V, desligando todo o sistema.

LEDs HL1–HL3 indicam status operacional.

Sinal do sensor — R1 (10 k Ω) —  —> Pino Arduino (A0/A1)



Reduz o nível de 12 V para aproximadamente 4,8 V (seguro para o Arduino).

Resumo do fluxo funcional

PS1 alimenta todo o sistema.

S2 → nível baixo → bombas 1–4 ligam.

S1 → nível alto → motor de agitação liga, depois bomba 5.

K2 → emergência → corta alimentação.

HL1–HL3 → indicam estados do sistema.

5.18 Motor Elétrico de Corrente Contínua (DC) com Escovas

Os motores DC escovados são amplamente utilizados em sistemas que exigem controle de velocidade e torque. Funcionam convertendo energia elétrica em movimento rotacional por meio da interação entre campos magnéticos. A tensão contínua aplicada gera corrente no rotor, que interage com o campo magnético fixo do estator, produzindo torque. As escovas de carvão e o comutador garantem a inversão da polaridade no rotor a cada meia-volta, mantendo a rotação contínua, com base na Lei de Lorentz.

Estrutura e Componentes;

Estator: Parte fixa com ímãs permanentes.

Rotor (armadura): Parte móvel com enrolamentos de cobre.

Comutador: Faz a comutação da corrente no rotor.

Escovas: Transferem corrente elétrica ao comutador.

Eixo: Transmite o movimento para a carga.

Aplicações Usado em:

Automação e robótica (Arduino, microcontroladores) Ventiladores e eletrodomésticos

Ferramentas portáteis (furadeiras, lixadeiras)

Modelos educacionais e protótipos;

Bombas em miniatura;

Vantagens;

Simples de controlar;

Baixo custo;

Alto torque em baixas rotações;

Fácil integração com eletrônica;

Desvantagens;

Desgaste das escovas e comutador;

Requer manutenção;

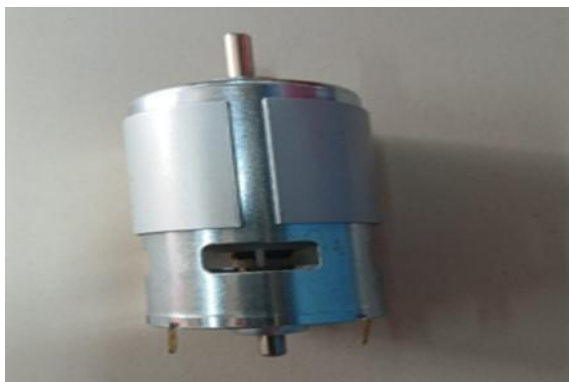
Geração de ruído elétrico;

Considerações Técnicas:

Operam normalmente entre 6 V e 24 V, com rotações de 3.000 a 10.000 RPM, variando com a carga. A rotação pode ser invertida trocando-se a polaridade da

alimentação. O controle de velocidade é feito com modulação por largura de pulso (PWM). Apesar do desgaste natural, são ideais para aplicações práticas, didáticas e econômicas. Motor Elétrico de Corrente Contínua (DC) Figuras 29 e 30.

Figura 29 - Motor DC



Fonte: Autores, 2025.

Figura 30 - Motor DC



Fonte: Autores, 2025

5.19 FUNCIONAMENTO PWM, (Pulse Width Modulation)

O PWM (Modulação por Largura de Pulso) é uma técnica digital usada para controlar a potência em cargas elétricas, ligando e desligando a alimentação em alta frequência, o que reduz perdas e aquecimento.

Ciclo de Trabalho (Duty Cycle):

É a porcentagem de tempo em que o sinal fica ligado em cada período. Um duty cycle de 10% fornece energia apenas 10% do tempo; 100% significa energia contínua. Ajustando o potenciômetro, é possível variar o duty cycle de 0% a 100%, controlando velocidade de motores ou brilho de LEDs.

Vantagem:

Alta eficiência, pois o componente de chaveamento (geralmente um MOSFET) dissipa pouco calor quando totalmente ligado ou desligado.

Características Técnicas

Tensão de Operação: varia conforme o modelo (ex.: 5–16V ou 12–40V DC).

Corrente Máxima: cerca de 8A contínuos e 10A de pico.

Potência Máxima: até 400W (40V × 10A).

Frequência PWM: geralmente entre centenas de Hz e alguns kHz (ex.: 500Hz a 13kHz).

Faixa de Controle: duty cycle ajustável de 0% ou 10% até 100%.

Componente Central: Circuito Integrado NE555 em modo astável, usado em controladores para motores DC com escovas (brushed).

Principais Aplicações

Controle de Velocidade de Motores DC: em veículos elétricos, ventiladores, bombas e robôs, mantendo bom torque em baixas rotações.

Controle de Iluminação: ajuste de brilho em fitas e lâmpadas LED.

Controle de Aquecedores: regulagem de temperatura em resistências.

Controle de Solenoides e Válvulas: ajuste preciso de fluxo ou abertura.

Sistemas Solares: usado em controladores de carga PWM para baterias.

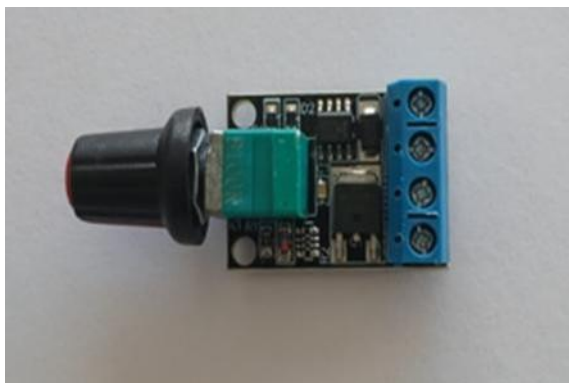
Como Utilizar:

Alimentação: conectar fonte DC aos terminais power+ e power- (atenção à polaridade).

Carga: ligar o dispositivo a ser controlado nos terminais output+ e output-.

Operação: ajustar o potenciômetro para variar a potência, controlando velocidade ou brilho. FUNCIONAMENTO PWM, Figuras 31 e 32.

Figura 31 – PWM



Fonte: Autores, 2025.

Figura 32 – PWM



Fonte: Autores, 2025.

5.20 Esquema, pino a pino – ligação direta (usar potenciômetro do HW-0

Pinos do HW-070 (conforme sua foto). de cima para baixo:

1. Power- (GND fonte)
2. Output- (Saida – para motor)
3. Output+ (Saida motor)
4. Power+ (Vcc fonte)

Ligação (Padrão):

Fonte DC (+12V) → Power+

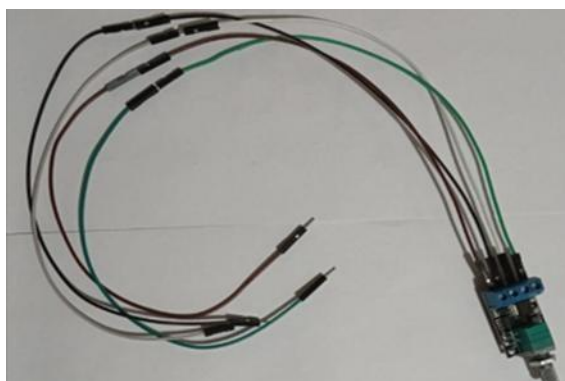
(pino 4) Fonte DC (GND) →

(Power- (pino) Motor (+) →

Output+ (pino) 3)

Motor (-) → Output_ (pino 2) - Potenciômetro, Figura,33

Figura 33 – Potenciômetro



Fonte: Autores, 2025.

5.21 Display LCD 1602 com Módulo I2C

O display LCD 1602 com módulo I2C é um dispositivo amplamente utilizado em projetos de eletrônica e automação, especialmente com plataformas como Arduino e Raspberry Pi, destinado à exibição de textos, números e informações de sensores. Ele combina um display LCD alfanumérico 16x2 (16 colunas e 2 linhas) com um módulo I2C, o que simplifica a comunicação com o microcontrolador e reduz significativamente o número de pinos necessários para sua operação.

Funcionamento:

O módulo opera por meio do protocolo I2C (Inter-Integrated Circuit), que utiliza apenas dois fios: SDA (dados) e SCL (clock). O chip PCF8574, presente no módulo I2C, funciona como um expensor de portas, convertendo os dados seriais em sinais paralelos compreendidos pelo controlador do display HD44780. Essa integração permite controlar o texto, o cursor e a luz de fundo (backlight) do display de forma eficiente.

Endereçamento I2C:

Cada dispositivo I2C na mesma linha de comunicação deve possuir um endereço único. O endereço padrão do LCD 1602 I2C costuma ser 0x27, podendo variar para 0x3F conforme o fabricante. Alguns módulos permitem a alteração desse endereço por meio de jumpers (A0, A1, A2), possibilitando a configuração em uma faixa entre 0x20 e 0x27.

Ajuste de Backlight e Contraste:

O backlight pode ser ativado ou desativado através de um jumper localizado no módulo, enquanto o contraste do texto é ajustado por um potenciômetro (trimpot) azul na parte traseira. Girar o trimpot no sentido horário geralmente aumenta o contraste, tornando o texto mais nítido.

Características Técnicas:

- Tipo de display: LCD alfanumérico 16x2
- Controladores: HD44780 (display) e PCF8574 (I2C)
- Interface: I2C (2 fios – SDA e SCL)
- Tensão de operação: 5V DC (compatível com 3.3V em alguns modelos)
- Cores: Backlight azul com caracteres brancos (pode variar)
- Matriz de caractere: 5x8 pontos
- Pinos de conexão: VCC, GND, SDA, SCL
- Dimensões aproximadas: 80 × 36 × 13 mm

Aplicações:

O LCD 1602 I2C é amplamente aplicado em:

- Exibição de dados de sensores (temperatura, umidade, pressão);
- Interfaces de usuário simples e menus de navegação;
- Sistemas de monitoramento e automação residencial;
- Projetos educacionais e experimentos de eletrônica;
- Ajuste e exibição de parâmetros em dispositivos programáveis.

Uso com Arduino:

Para utilização com Arduino, deve-se incluir a biblioteca `LiquidCrystal_I2C`, que simplifica a comunicação e o controle do display.

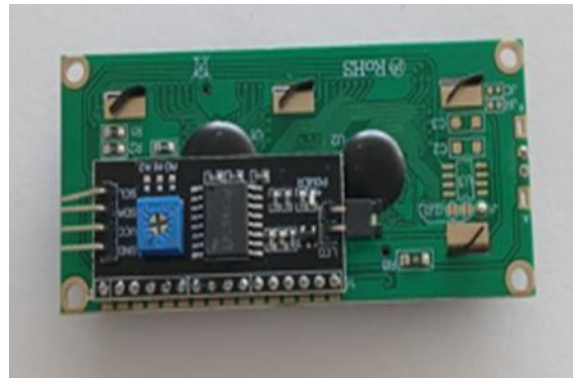
Com apenas algumas linhas de código, é possível exibir textos, valores de sensores e mensagens personalizadas, tornando o dispositivo ideal para projetos práticos e didáticos. Display LCD, Figura 34 e 35.

Figura 34 - Display LCD 16x2



Fonte: Autores, 2025.

Figura 35 - Display LCD 16x2



Fonte: Autores, 2025.

5.22 LCD 16x2 (COM MÓDULO 12C, CABIADO)

O módulo 12C (plaquinha azul colada atrás do LCD) possui 4 pinos A ligação é feita conforme o modelo do Arduino (no seu caso, UNO). Conexão elétrica do LCD 16X2 (COM MÓDULO 12C) GND,VCC,DAS,SCL

Esses pinos comunicam-se com o Arduino através do barramento 12C. Ligação pino a pino – LCD 16x2 (I2C) com Arduino UNO

Pino no Módulo I2C – Função Pino no Arduino UNO Cor recomendada, Observações

VCC Alimentação +5V - 5V  Vermelho Alimenta o LCD

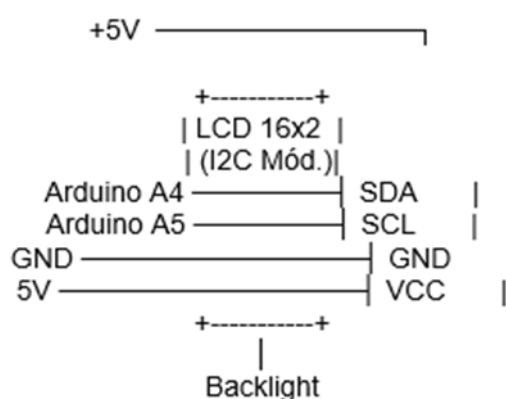
GND Terra (comum) GND  Preta GND comum com o sistema

DAS Linha de dados I2C A4  Verde Comunicação serial de dados

SCL Clock do barramento LC – A5  Azul Sincroniza a leitura de:

Figura 36 - Esquema Elétrico

Esquema elétrico (descrição textual)



Fonte: Autores, 2025.

5.23 Função dos componentes no esquema do projeto dosador

5.24 Descrição da Montagem

Etapa 1 – Alimentação

A fonte de 12 V é conectada ao módulo PWM e ao borne de entrada do fusível de proteção. O fusível de 10 A é inserido em série com a alimentação principal, garantindo a segurança contra curtos.

O Arduino Uno recebe alimentação de 5 V, que pode ser derivada de um regulador do módulo PWM ou de uma fonte externa dedicada.

Etapa 2 – Controle de velocidade (PWM)

O potenciômetro é conectado ao pino A0 do Arduino, com os terminais laterais em 5 V e GND. O Arduino lê o valor analógico e converte para um sinal PWM no pino D9, enviando-o ao módulo PWM que controla o motor DC.

O módulo relé atua como chave, permitindo ligar ou desligar o circuito de potência do motor.

Etapa 3 – Display LCD 16x2

O display é conectado ao Arduino via barramento I2C, utilizando:

SDA → pino A4 SCL → pino A5 VCC → 5 V GND → GND

O display exibe informações como “Velocidade: XX%” e “Motor: ON/OFF”.

Etapa 4 – Módulo Relé

Pino IN1 do relé → pino D7 do Arduino.

VCC e GND → alimentação de 5 V e GND do Arduino.

Os terminais COM e NO do relé são conectados em série com o motor, permitindo o controle da alimentação do motor DC.

Etapa 5 – Motor DC

O motor é conectado à saída do módulo PWM e controlado pelo relé.

A rotação varia conforme o ajuste do potenciômetro, e o valor é atualizado no display LCD em tempo real.

Etapa 6 – Esquema de Ligações (Resumo Pino a Pino):

Componente	Pino	Conexão
de Ligação Arduino	Potenciômetro	Centro A0 Potenciômetro Lateral 1
5V Potenciômetro	Lateral 2	GND

Módulo Relé IN1	D7	Módulo Relé VCC	5V	Módulo Relé GND	GND
LCD 16x2 (I2C) SDA	A4	LCD 16x2 (I2C) SCL	A5	LCD 16x2 (I2C) VCC	5V
LCD 16x2 (I2C)	GND	GND			

Módulo PWM	Entrada de controle Pino D9 (PWM)	Motor DC Saída
PWM Terminais do motor		

Fonte 12 V + / – Alimenta PWM e relé via fusível

Etapa 7 – Funcionamento Geral:

O usuário gira o potenciômetro, alterando a tensão lida pelo Arduino.

O Arduino converte esse valor em um sinal PWM e o envia ao módulo controlador. O motor DC responde à variação de largura de pulso, alterando a velocidade.

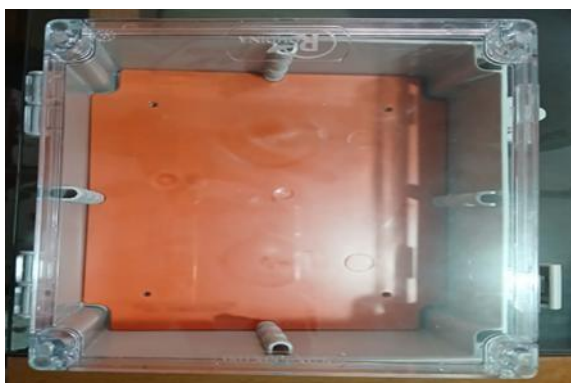
O display LCD mostra a velocidade percentual e o estado do motor.

O relé comanda o acionamento ou desligamento total do circuito de potência. O fusível protege o sistema em caso de sobrecorrente ou curto.

5.25 Considerações de Segurança

O circuito deve ser montado sobre base isolante de PVC, Figura 37 e 38, evitando curtos. É essencial garantir bom isolamento nos cabos de alta corrente (motor e fonte). O fusível deve ser dimensionado conforme a corrente máxima do motor. Recomenda-se o uso de bornes e conectores fixos para facilitar manutenção e montagem.

Figura 37 - Painel de Comando



Fonte: Autores, 2025.

Figura 38 - Painel de Comando



Fonte: Autores, 2025.

5.26 MOTOR DC ESCOVADO (5 – 12 V)

Função geral: Responsável por converter energia elétrica em energia mecânica rotativa.

É usado no sistema para acionar um eixo rotativo sob controle de velocidade via PWM e comando liga/desliga via relé.

Características técnicas típicas:

Parâmetro	Especificação
-----------	---------------

Tipo	Motor DC escovado de ímã permanente	Tensão nominal	5 V – 12 V DC
------	-------------------------------------	----------------	---------------

Corrente nominal	0,6 – 1,5 A	Corrente de partida	Até 3 A	Potência aproximada	6 – 18 W
------------------	-------------	---------------------	---------	---------------------	----------

Rotação típica	3000 – 6000 RPM
----------------	-----------------

Sentido de rotação	Invertido pela troca de polaridade
--------------------	------------------------------------

Terminais	2 (positivo e negativo)
-----------	-------------------------

Material do corpo;	Aço com eixo metálico cromado
--------------------	-------------------------------

Fixação;	Suporte isolante com parafusos e porcas de travamento
----------	---

Ligações elétricas:

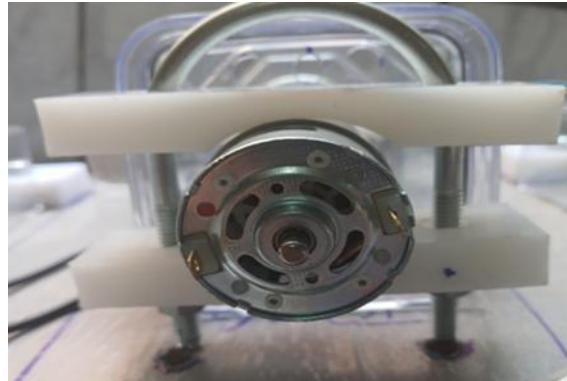
Positivo (+): Saída do módulo PWM (OUT+). Negativo (–): Saída do módulo PWM (OUT–). Proteção: Diodo 1N5408 em paralelo (cátodo no +).

Observações:

Usar fios de seção mínima 1,0 mm².

Fixar mecanicamente para evitar vibração e desalinhamento do eixo.
MOTOR DC ESCOVADO (5 – 12 V), Figura 39.

Figura 39 - Motor DC



Fonte: Autores, 2025.

5.27 MÓDULO DE RELÉS 8 CANAIS (SRD-05VDC-SL-C)

Função geral: Permite acionar cargas elétricas (motores, bombas,) através de sinais digitais de 5 V do Arduino.

Cada canal funciona como uma chave eletromecânica isolada controlada

eletronicamente.

Características técnicas:

Parâmetro Especificação

Tipo de relé Songle SRD-05VDC-SL-C Tensão da bobina 5 V DC

Corrente da bobina 70 mA por canal

Capacidade de comutação 10 A / 250 VAC ou 10 A / 30 VDC Entradas de controle, IN1 – IN8 (nível lógico 0–5 V)

Alimentação lógica VCC (5 V Arduino) Alimentação de potência JD-VCC (5 V da fonte) Isolamento Optoacopladores PC817

Jumper JD-VCC/VCC Une ou isola lógica e potência

Ligações elétricas:

IN1–IN8 → Pinos digitais do Arduino (D2–D9). VCC (lógica) → 5 V do Arduino.

GND (lógica) → GND do Arduino. JD-VCC (bobinas) → +5 V da fonte. GND (bobinas) → GND da fonte.

COM / NO / NC → Linhas de carga (alimentação do PWM).

Observações:

Remover jumper JD-VCC/VCC para isolamento óptico real.

Alimentar a bobina dos relés com fonte independente (não pelo Arduino).
Usar fusível 10 A em cargas indutivas.

Modo relês 8 canais permite acionar cargas elétricas (motores, bombas,),
Figura 40.

Figura 40 - Módulo Relé



Fonte: Autores, 2025.

5.28 MÓDULO PWM, FIGURA 42 – CONTROLADOR DE VELOCIDADE DE MOTOR DC

Função geral

Controla a velocidade do motor DC ajustando a largura dos pulsos elétricos (modulação PWM). Permite variar a rotação sem reduzir significativamente o torque.

Características técnicas:

Parâmetro Especificação Tensão de entrada 5 – 35 V DC Corrente máxima 10 A (contínua)

Frequência de trabalho 10 – 25 kHz Potência máxima 120 W

Controle Potenciômetro 10 k Ω (manual) ou sinal PWM externo

Eficiência $\geq 90\%$

Conectores VCC, GND, OUT+, OUT-, PWM IN (opcional)

Ligações elétricas

VCC (entrada) $\rightarrow$ Saída 12 V do relé (NO). GND (entrada) $\rightarrow$ GND da fonte.

OUT+ / OUT- $\rightarrow$ Motor DC.

PWM IN (opcional) $\rightarrow$ Pino PWM do Arduino (D5, D6, D9). Potenciômetro $\rightarrow$ Ajuste manual da velocidade.

Observações

Instalar capacitor 470 μ F/25 V entre VCC e GND. Montar com dissipador de calor no MOSFET.

Para cargas acima de 10 A, usar driver MOSFET externo.

DISPLAY LCD, FIGURA,43 16x2 (COM MÓDULO I2C)

Função geral

Display alfanumérico que exibe informações do sistema, como: Velocidade do motor;

Estado dos relés;

Nível de tensão ou corrente; Mensagens de status do Arduino. Características técnicas Parâmetro Especificação.

Tipo LCD 16x2 com interface I2C Controlador HD44780 ou compatível Tensão de operação 5 V DC Corrente típica 20 mA.

Interface I2C (2 fios) Endereço padrão 0x27 ou 0x3F Endereço padrão 0x27 ou 0x3F

Endereço padrão 0x27 ou 0x3F Endereço padrão 0x27 ou 0x3F Pinos VCC, GND, SDA, SCL

Ligações elétricas

Pino LCD Ligação Arduino Função VCC 5 V Alimentação

GND GND Referência SDA A4 Dados I2C SCL A5 Clock I2

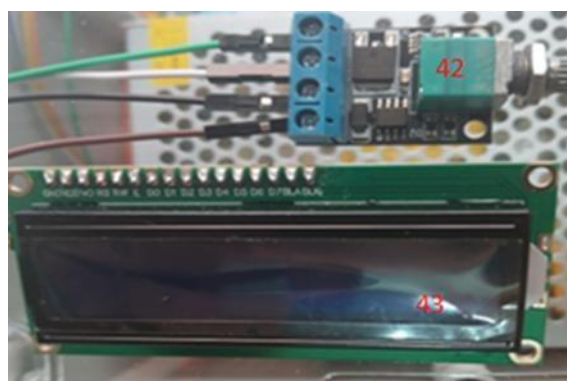
Observações

Usar biblioteca LiquidCrystal_I2C.h no Arduino IDE. Ajustar contraste pelo trimpot azul no módulo I2C.

Confirmar endereço I2C com scanner antes da programação.

Módulo PWM, figura 41, controlador de velocidade de motor DC, E Display LCD 16x2.

Figura 41 - Módulo PWM (42) Display LCD 16x2 (43)



Fonte: Autores, 2025.

5.29 FONTE CHAVEADA 12 V DC (INDUSTRIAL)

Função geral: Fonte de alimentação chaveada estabilizada que converte 110/220 V AC em 12 V DC para alimentar motores, relés e conversores step-down (5 V).

Características técnicas Parâmetro Especificação

Tipo Fonte chaveada industrial (carcaça metálica perfurada) Tensão de entrada 110/220 V AC automática

Tensão de saída 12 V DC regulado Corrente máxima 5 A (60 W)

Regulação de tensão $\pm 10 \%$ (trimpot de ajuste) Proteções, Sobrecorrente, sobretensão e curto-circuito

Terminais L (fase), N (neutro), $\perp$ (terra), V+, V-

Ligações elétricas

L $\rightarrow$ Fase (rede AC 110/220 V). N $\rightarrow$ Neutro.

$\perp$ (terra) $\rightarrow$ Aterramento do painel.

V+ (12 V) $\rightarrow$ Alimenta relés, módulo PWM e step-down (para 5 V).

V- (GND) $\rightarrow$ Comum a todos os módulos.

Observações

Instalar fusível 10 A na entrada AC.

Confirmar polaridade antes da ligação ao Arduino. Assegurar ventilação adequada e fixação firme.

Aterramento obrigatório na estrutura metálica.

Fonte de Alimentação Chaveada que converte 110/220 V AC em 12 V DC, Figura 42.

Figura 42 - Fonte chaveada, 12V DC



Fonte: Autores, 2025.

5.30 RESUMO FUNCIONAL DO SISTEMA

Componente	Função no Sistema	Alimentação, Controle
Motor DC	Gera movimento mecânico 12 V DC via PWM	PWM manual ou Arduino
Módulo de Relés 8 canais	Liga/desliga cargas 5 V bobina / 12 V carga	Arduino digital
Módulo PWM	Regula velocidade do motor, 12 V DC	Potenciômetro ou sinal PWM
LCD 16x2 I2C	Exibe dados e status, 5 V DC	I2C (A4/A5)
Fonte chaveada	Alimenta todo o sistema, 110/220 V	AC → 12 V

5.31 FUNCIONAMENTO ELÉTRICO EM CONJUNTO

1. Circuito de Potência (Força)

Este circuito que alimenta o motor, fornecido pela Fonte Chaveada. O fluxo de energia é chaveado (LIGA/DESLIGA) pelo Módulo de Relés e modulado (velocidade) pelo novo Driver de Motor.

Etapa	Componentes Envolvidos	Fluxo Elétrico e Ação
1. Fonte de DC	Fonte Chaveada	Converte $\{110/220\text{V AC}\}$ em $\text{+VDC}\{12\text{V}\}$ ou $\text{+VDC}\{24\text{V}\}$ e GND .
2. Chave Mestra ON/OFF	Fonte (+VDC), Módulo Relé	A linha de +VDC da Fonte entra no pino COM do Relé. O Arduino (Lógica) energiza a bobina do Relé, que fecha a chave interna, liberando o +VDC pela saída NO .
3. Alimenta ção do Driver	Relé (NO), Novo Driver de Motor	O +VDC chaveado alimenta o pino de entrada de potência (VIN+) do Driver. O GND da Fonte conecta-se ao GND do Driver.
4. Modulação (PWM)	Novo Driver de Motor, Motor DC	O Driver usa a tensão +VDC e a chaveia em alta frequência, criando um sinal PWM. Este sinal pulsado é enviado ao Motor DC.
5. Operação do Motor	Motor DC	O Motor DC recebe a potência pulsada. A tensão média eficaz (proporcional ao do PWM) determina a sua velocidade de rotação.

ii. *Circuito de Lógica (Controle e Sinais)*

Este circuito opera em baixa tensão ($\{5V\}$) é responsável por gerar os sinais de controle e exibir o status.

Componentes Envolvidos	Fluxo Elétrico e Ação	
1. Referencial Comum	Arduino, Fonte Chaveada	O pino $\text{\text{GND}}$ do Arduino deve ser conectado ao pino $\text{\text{GND}}$ da Fonte. Isto garante que os sinais lógicos ($\text{\text{5V}}$) e de potência ($\text{\text{+VDC}}$) tenham o mesmo ponto de referência para que a comunicação e o controle funcionem corretamente.
2. Sinal $\text{\text{ON/OFF}}$	Arduino ($\text{\text{D8}}$), Módulo Relé	O Arduino envia um sinal digital $\text{\text{LOW}}$ (típico) ou $\text{\text{HIGH}}$ para o pino $\text{\text{IN1}}$ do Módulo Relé. Esta baixa corrente ($\text{\text{mA}}$) aciona o circuito interno do relé, que fecha o contato de alta potência (Etapa 2, Circuito de Potência).
3. Sinal de Velocidade	Arduino ($\text{\text{D6}}$) $\sim$ PWM), Novo Driver	O Arduino usa a função <code>analogWrite()</code> para gerar um sinal PWM (geralmente em $\text{\text{5V}}$). Este sinal entra no pino de controle do Driver. O $\text{\text{Duty Cycle}}$ deste sinal informa ao Driver com que frequência ele deve chavear a alta potência do motor.
4. Exibição de Dados	Arduino, LCD	O Arduino usa os pinos $\text{\text{SDA}}$ ($\text{\text{A4}}$) e $\text{\text{SCL}}$ ($\text{\text{A5}}$) para comunicação $\text{\text{I2C}}$ (se estiver usando adaptador). O Arduino envia dados de texto e status (Ex: velocidade) ao LCD

- a. Painel Montado, Simulando Do Misturador E Dosador De Tinta
- b. Montagem do Pannel, Figura 43, realizando a configuração do ensaio

Figura 43 - Pannel montado



Fonte: Autores, 2025.

Processo de Montagem do Pannel:

Etapa 1 – Preparação dos componentes

Separação do Ardíno, Módulo relé, cabos jumper, conectores e sensores/atuadores Conferência de funcionamento dos módulos individuais.

Etapa 2 – Conexão elétrica

O Arduino Uno é ligado ao computador via cabo USB para alimentação e programação.

Os pinos digitais do Arduino são conectados às entradas do módulo de relés (normalmente IN1 a IN8).

O módulo de relés recebe alimentação 5 V e GND do Arduino.

Cada relé controla a ligação de um atuador (válvula, bomba, motor etc.), servindo como chave de controle.

Etapa 3 – Programação no Arduino

No software Arduino IDE, é desenvolvido um código para comandar cada relé conforme uma lógica (por exemplo, leitura de sensores, temporização ou acionamento manual).

O programa é enviado à placa via cabo USB.

Etapa 4 – Teste e validação do sistema

Após o upload, verifica-se o acionamento dos relés conforme os comandos definidos. Cada atuador é testado para garantir o funcionamento seguro e sincronizado.

3. Função do conjunto

O sistema mostrado é típico de bancadas didáticas de automação e controle, permitindo:

Simular processos industriais (como controle de válvulas e bombas); implementar protótipos de automação residencial ou predial;

Testar programações lógicas com Arduino e relés.

Ensaio de Funcionamento do Misturador e Dosador de Tinta

Erro LiquidCrystal_I2C no Projeto de Misturador e Dosador de Tinta

Durante a compilação do código do misturador e dosador de tinta, foi identificado o erro “LiquidCrystal_I2C.h: No such file or directory”, indicando ausência ou instalação incorreta da biblioteca responsável pelo controle do display LCD via interface I2C. Esse problema interrompe a execução do programa e impede que o sistema exiba informações essenciais de operação.

A correção exige a instalação adequada da biblioteca LiquidCrystal_I2C em sua versão compatível com o código. Após a reinstalação e verificação do caminho da biblioteca, o display passou a funcionar corretamente, permitindo o monitoramento das etapas de dosagem, agitação e esvaziamento.

Com o display restabelecido, o restante do sistema — estruturado em lógica não bloqueante e máquina de estados — voltou a operar normalmente, garantindo respostas rápidas, funcionamento seguro e interação adequada com os botões e sensores, incluindo o de emergência.

Durante o desenvolvimento do sistema automatizado de dosagem e mistura, foram identificados dois grupos principais de falhas: um erro no código referente ao display LCD I2C e problemas na alimentação elétrica dos sensores e dos módulos de relé.

1. Erro do LCD — Código e IDE Não Reconheciam a Biblioteca

No processo de compilação do programa, a IDE apresentou o erro “LiquidCrystal_I2C.h: No such file or directory”. Embora o hardware estivesse correto, o problema estava no código, que fazia referência à biblioteca LiquidCrystal_I2C sem que a IDE a reconhecesse. Isso indica que:

- a biblioteca não estava instalada,
- ou estava instalada em diretório incorreto,
- ou o código utilizava uma versão não compatível com a biblioteca existente.

A correção envolveu reinstalar a biblioteca correta e ajustar o código para que a IDE pudesse identificá-la. Após esse ajuste, o display LCD passou a funcionar normalmente, permitindo a exibição dos estados do sistema — dosagem, agitação e esvaziamento — e restabelecendo o monitoramento visual essencial para a operação.

2. Alimentação dos Sensores e do Arduíno

O projeto elétrico foi planejado com fontes independentes para separar os circuitos de potência e controle:

- 12 V destinados aos relés,
- 9 V destinados ao Arduíno,
- e 5 V no retorno dos sensores, para evitar sobrecarga nas entradas do microcontrolador.

Como os sensores trabalhavam originalmente acima de 5 V, foi instalada uma fonte chaveada regulada em 5 V para garantir que os sinais enviados ao Arduíno estivessem dentro do limite seguro. Essa adaptação funcionou corretamente na proteção do microcontrolador.

3. Falha na Alimentação dos Relés

Mesmo com a separação adequada das fontes, o sistema apresentou um problema crítico: não chegou tensão de 12 V ao módulo de relés, impedindo o acionamento da parte de potência. Foram realizadas diversas tentativas de correção utilizando divisores de tensão e ajustes no circuito, mas sem sucesso — o que era esperado, já que divisores não são projetados para alimentar dispositivos que exigem corrente elevada, como relés.

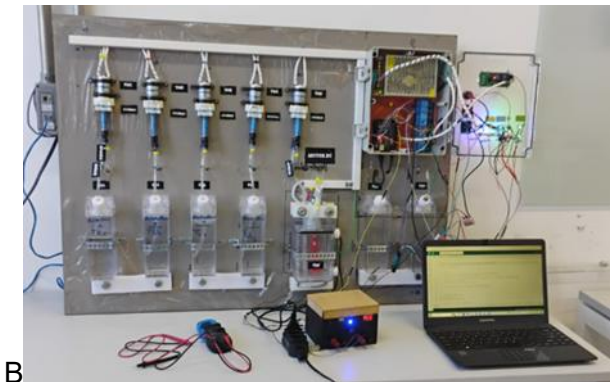
A ausência do sinal de 12 V sugere falhas como:

- interrupção na linha de alimentação da fonte de 12 V,
- fonte de potência defeituosa,
- mau contato ou fiação incorreta no módulo de relés,
- falta de aterramento comum entre as fontes.

Correção Técnica

A correção do erro no LCD, resolvida mediante ajuste no código e reconhecimento da biblioteca pela IDE, devolveu ao sistema sua interface de monitoramento, comprovando que a lógica de software estava funcional. No entanto, os problemas persistentes na alimentação dos relés e na distribuição da tensão dos sensores evidenciam que a parte elétrica de potência necessita de revisão profunda. Figura,44.

Figura 44 - Painel montado.



Fonte: Autores, 2025.

5.31.1 Mapeamento e Identificação dos Pinos no Esquema

MAPEAMENTO DE PINOS DEFINITIVO							
Arduíno Pin	Função(IN1/BTN)	Relé/Dispositivo	Obs	D8	Out(IN7)	REL7->Reserva Alarme	
D2	OUT(IN1)	REL1->Bomba M1	Active Low/Active HIGH,Ajustar				
D3	OUT(IN2)	REL2->Bomba M2		D9	Out(IN8)	REL->Reserva Indicador	
D4	OUT(IN3)	REL3->Bomba M3					
D5	OUT(IN4)	REL4->Bomba M4					
D6	OUT(IN5)	REL5->Bomba M5(Sucção)		D10	BTN	BTNM1	Botão

D7	OUT(IN6)	REL6->Motor Agitador M6(INPUT_PULLUP)	GND				Entre Pino
D11	BTN	BTNM2(INPUT_PULL UP)	A4		Sensor	Sensor Nível Condi- cionado	Usar Divi- sor, Trans- stor
D12	BTN	BTNM3(INPUT_PULL UP)					
A0	BTN	BTNM4(INPUT_PULL UP)					
A1	BTN	BTNM5(INPUT_PULL UP)					
A2	BTN	BTN Agitador(opt)	A5		Stop	Energie- ncy Stop (INPUT - PULLU	Ativo Low Desli- Ga- tudo
A3	MODE	Seletor Auto Manual (INPUT_PULLUP)					

Fonte: Autores, 2025.

6. CONCLUSÃO

O projeto demonstrou que é plenamente viável desenvolver um sistema automatizado de dosagem e mistura de líquidos utilizando o microcontrolador Arduino como núcleo de controle. A integração entre sensores, atuadores, módulos de relés e interfaces de comunicação digital permitiu criar uma solução de baixo custo, flexível e funcional, capaz de controlar o processo com precisão e repetitividade.

A implementação da lógica de controle, baseada em leitura de sensores, acionamento sequencial das bombas e gerenciamento do motor misturador, comprovou que o Arduino pode executar tarefas típicas de sistemas industriais quando aplicado de forma adequada. Durante os testes, o sistema mostrou capacidade de identificar variações no processo, ajustar a dosagem conforme o tempo e a velocidade programados e reagir a situações de falha, garantindo maior segurança operacional.

O estudo também evidenciou a importância da prototipagem e da calibração para atingir a precisão necessária na dosagem, além da relevância dos sensores de nível e pressão para monitoramento confiável do processo. O uso de comunicação digital e da estrutura modular do Arduino contribuiu para facilitar futuras expansões, indicando que o sistema pode evoluir para aplicações mais complexas ou integradas à IoT.

Conclui-se, portanto, que a automação proposta atinge seu objetivo principal: substituir procedimentos manuais por um processo automatizado mais rápido, preciso e seguro. O projeto representa um avanço acessível para pequenas e médias operações industriais, mostrando que tecnologias abertas podem ser utilizadas para otimizar processos, reduzir erros e aumentar a eficiência produtiva.

REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (Brasil); International Society of Automation. ANSI/ISA-101.01-2015: human machine interfaces for process automation systems. Research Triangle Park, NC: ISA, 9 July 2015. 62 p. ISBN 978- 1-941546-46-8. Acesso em 19 junho 2025.

BAYER, Fernando Mariano; ARAÚJO, Olinto César Bassi de. Controle automático de processos. Santa Maria, RS: Universidade Federal de Santa Maria; Colégio Técnico Industrial de Santa Maria; e Tec Brasil, 2011. 92 p. Disponível em: <https://www.e-tec.br/cursos/2011/controle-automatico-de-processos>. Acesso em: 19 jun. 2025.

BANZI, Massimo; SHILOH, Michael. Primeiros Passos com o Arduino–2ª Edição: A plataforma de prototipagem eletrônica open source. Novatec Editora, 2015. Banzi, M., & Shiloh, M. (2015). Primeiros Passos com o Arduino–2ª Edição: A plataforma de prototipagem eletrônica open source. Novatec Editora. Novatec Editora, 2015. acesso em 25 abril 2025.

Carvalho, Gilson Amorim, José Vicente Cardoso Santos, and Raul Santos. "Arduino na automação industrial: vantagens, limitações e perspectivas futuras." *Apoena* 6 (2023): 274-283. CARVALHO, Gilson Amorim; SANTOS, José Vicente Cardoso; SANTOS, Raul. Arduino na automação industrial: vantagens, limitações e perspectivas futuras. *Apoena*, v. 6, p. 274-283, 2023. Carvalho, G. A., Santos, J. V. C., & Santos, R. (2023). Arduino na automação industrial: vantagens, limitações e perspectivas futuras. *Apoena*, 6, 274-283. acesso em :16 abril 2025.

Carvalho, Gilson Amorim; SANTOS, José Vicente Cardoso; SANTOS, Raul. Arduino na automação industrial: vantagens, limitações e perspectivas futuras. *Apoena*, v. 6, p. 274-283, 2023. Carvalho, G. A., Santos, J. V. C., & Santos, R. (2023). Arduino na automação industrial: vantagens, limitações e perspectivas futuras. *Apoena*, 6, 274- 283. acesso em: 15 abril 2025.

Carvalho, Gilson Amorim; SANTOS, José Vicente Cardoso; SANTOS, Raul. Arduino na Automação Industrial: Vantagens, Limitações e Perspectivas Futuras. Apoená Revista Eletrônica: Revista eletrônica do Programa de Iniciação Científica da UNIJORGE, Salvador, v. 6, p. 274-283, junho 2023. Disponível em: <https://transformauj.com.br/wp-content/uploads/2023/06/22.-Arduino-na-Automacao-Industrial-Vantagens-Limitacoes-e-Perspectivas-Futuras.pdf>. Acesso em: 28 mar. 2025.

CAPELLI, Alexandre. Automação industrial: controle do movimento e processos contínuos. Ed. Érica, 2008. Capelli, A. (2008). Automação industrial: controle do movimento e processos contínuos. acesso em :15 março 2025.

CHAPMAN, S. J. Máquinas Elétricas e Transformadores. 5ª ed. Porto Alegre: AMGH, 2012. Acesso em : 18/10/2025.

FERRONI, Eduardo, et al. "A plataforma Arduino e suas aplicações." Revista da UI_IPSantarém (2015). FERRONI, Eduardo et al. A plataforma Arduino e suas aplicações. Revista da UI_IPSantarém, 2015. Ferroni, E., Vieira, H., Nogueira, J., Santos, R., Lemos, R., & Rodrigues, T. (2015). A plataforma Arduino e suas aplicações. Revista da UI_IPSantarém. Acesso em :04 maio 2025.

FELISARDO, Raul José Alves et al. USO DO ARDUINO COMO FERRAMENTA DE PROTOTIPAGEM PARA DESENVOLVIMENTO DE DISPOSITIVOS AUTOMÁTICOS: UMA REVISÃO. Caderno de Graduação-Ciências Exatas e Tecnológicas-UNIT-SERGIPE, v. 8, n. 1, p. 11-26, 2023. Felisardo, R. J. A., dos Santos, G. N., & Galvão, D. G. (2023). USO DO ARDUINO COMO FERRAMENTA DE PROTOTIPAGEM PARA DESENVOLVIMENTO DE DISPOSITIVOS AUTOMÁTICOS: UMA REVISÃO. Caderno de Graduação-Ciências Exatas e Tecnológicas-UNIT-SERGIPE, 8(1), 11-26. acesso em 25 abril 2025.

FITZGERALD, A. E.; KINGSLEY, C.; UMANS, S. D. Máquinas Elétricas. 6ª ed. Porto Alegre: AMGH, 2014. Acesso em : 18/10/2025.

FREITAS. Medição de nível sem contato físico com os materiais (dissertação).

Universidade Federal de Itajubá – UNIFEI, 2013.

Disponível

em:

https://repositorio.unifei.edu.br/jspui/bitstream/123456789/922/1/dissertacao_freitas_2013.pdf. Acesso em: 18/10/2025.

ISA (International Society of Automation) em conjunto com o ANSI (American National Standards Institute, ANSI/ISA-101.01-2015).

LIMA, Carlos Henrique dos Santos et al. CONTROLE DE PROCESSOS PID UTILIZANDO CLP. 2023. Lima, C. H. D. S. (2023). CONTROLE DE PROCESSOS PID UTILIZANDO CLP. acesso 13 maio 2025

LIMA, Carlos Henrique dos Santos et al. CONTROLE DE PROCESSOS PID UTILIZANDO CLP. 2023. Lima, C. H. D. S. (2023). CONTROLE DE PROCESSOS PID UTILIZANDO CLP. acesso em 14 março 2025.

LIMA, Emanuel Maycon Santos; NOBRE, Antonio Ygo Magalhães; ALENCAR, Rômulo Alexandre Ellery de. Automação residencial de baixo custo com arduino Mega e ethernet shield. Curso de Sistemas de Informação no Centro Universitário Estácio do Ceará, 2015. Lima, E. M. S., Nobre, A. Y. M., & ALENCAR, R. A. E. D.

Livros 1. MALVINO, Albert; BATES, David J. Eletrônica. 8. ed. Porto Alegre: AMGH, 2016. v. 2.2. HOROWITZ, Paul; HILL, Winfield. The Art of Electronics. 3. ed. Cambridge: Cambridge University Press, 2015.

(2015). Automação residencial de baixo custo com arduino Mega e ethernet shield. Curso de Sistemas de Informação no Centro Universitário Estácio do Ceará.

.Manuais Técnicos (Datasheets) 4. TEXAS INSTRUMENTS. NE555 Precision Timer. Dallas, TX, 2023. Disponível em: <https://www.ti.com/lit/ds/symlink/ne555.pdf>. Acesso em: 19 out. 2025. 5. INFINEON McROBERTS, Michael. Beginning Arduino. New York: Apress, 2010.

Margolis, Michael, Brian Jepson e Nicholas Robert Weldin. Livro de receitas do Arduino: receitas para começar, expandir e aprimorar seus projetos. O'Reilly Média, 2020. MARGOLIS, Michael; JEPSON, Brian; WELDIN, Nicholas Robert. Livro de receitas do Arduino: receitas para começar, expandir e aprimorar seus projetos.

O'Reilly Média, 2020. Margolis, M., Jepson, B., & Weldin, NR (2020). Livro de receitas do Arduino: receitas para começar, expandir e aprimorar seus projetos. O'Reilly Média. Acesso em: 02 maio 2025.

MIMS III, Forrest M. Timer, Op Amp, and Optoelectronic Circuits and Projects. 1. ed. [S. l.]: Master Publishing, 2004.

MONK, Simon; MCCABE, Michael. Programming Arduino: getting started with sketches. New York: McGraw-Hill Education, 2016. Monk, S., & McCabe, M. (2016). Programming Arduino: getting started with sketches (Vol. 176). New York: McGraw-Hill Education. acesso em :14 abril 2025.

MONK, Simon. Programação com Arduino: começando com Sketches. Bookman Editora, 2017. Monk, S. (2017). Programação com Arduino: começando com Sketches. Bookman Editora. acesso em: 25 abril 2025.

MONK, Simon. 30 projetos Arduino para o gênio do mal. Nova Iorque: McGraw-Hill Education, 2013. Monk, S. (2013). 30 projetos Arduino para o gênio do mal. Nova Iorque: McGraw-Hill Education. acesso em: 30 abril 2025.

MEDEIROS, D. A.; MENEZES, L. M. Motores Elétricos – Teoria e Prática. São Paulo: Érica, 2016. Acesso em: 18/10/2025.

MENDES, Rafael. Programação de CLPs Métodos Técnicos: Controladores Lógico Programáveis. 2. Estratégias de programação. 3. Linguagens de programação. 4. Técnicas de implementação. I. Autor. II. Título. São Carlos: Editora Scienza, 2021. 286

p. v. volume 1. ISBN 978-65-5668-037-8. Disponível em: <https://referenciabibliografica.net/a/pt-br/ref/abnt>. Acesso em: 11 abr. 2025.

Manuel; MATOS, João Nuno. Osciladores para rádio frequência de elevada estabilidade. Eletrónica e Telecomunicações, v. 2, n. 2, p. 169-173, 1998. Tavares, J. M., & Matos, J. N. (1998). Osciladores para rádio frequência de elevada estabilidade.

Eletrônica e Telecomunicações, 2(2), 169-173. acesso em 26 abril 2025.

SUNFOUNDER. I2C LCD 1602 — SunFounder Ultimate Sensor Kit documentation.

Disponível em: https://docs.sunfounder.com/projects/ultimate-sensor-kit/en/latest/components_basic/21-component_i2c_lcd1602.html. Acesso em: 19 out. 2025.

MobilBatch-Equipamentos Industriais-São Paulo-SP,

(www.mobilbatch.com.br/equipamentos-industriais). Acesso em 01/05/25.

NASCIMENTO, A. L. Motores Elétricos DC e Controle Eletrônico. São Paulo: Érica, 2020. Acesso em: 18/05/2025.

OLIVEIRA, AL LIMA et al. Instrumentação—Fundamentos de Controle de Processo—. CPM-Programa de Certificação de Pessoal de Manutenção. Vitória: SENAI-ES, 1999. OLIVEIRA, A. L., OLIVEIRA, W., TRAZZI, R., DIAS, F., NETO, E., & CARDOSO,

R. (1999). Instrumentação—Fundamentos de Controle de Processo—. CPM-Programa de Certificação de Pessoal de Manutenção. Vitória: SENAI-ES. acesso 14 maio 2025.

OGATA, SOUZA, Heloisa Coimbra de; TANNURI, Eduardo Aoun. Engenharia de controle moderno. 1998. Ogata, K., Souza, H. C. D., & Tannuri, E. A. (1998). Engenharia de controle moderno. acesso em 14 março 2025.

PATSKO, Luís Fernando. Tutorial controle de motor de passo. Maxwell Bohr—Instrumentação Eletrônica, 2006. Patsko, L. F. (2006). Tutorial controle de motor de passo. Maxwell Bohr—Instrumentação Eletrônica. acesso 17 maio 2025.

PORTES, W. A. D. O. (2014). Utilização de arduino e eletrônica da automação residencial com acessibilidade à pessoa portadora de deficiência. PORTES, Wagner

A. de O. Utilização de arduino e eletrônica da automação residencial com acessibilidade à pessoa portadora de deficiência. 2014. acesso 11 maio 2025.

PLC Technician Program - On line Education Program. The Evolution of PLCs. 2021. Disponível em: <https://www.plctechnician.com/news-blog/evolution-plcs>. Acesso em: 17 abr.2025.

Rodrigues, T. (2015). A plataforma arduíno e suas aplicações. Revista da UI_IPSantarém. Acesso em: 01 maio 2025.

RIBEIRO, Marco Antônio. Automação Industrial. 4. ed. Salvador: Tek Treinamento & Consultoria Ltda, Outorno 1999. 498 p. Disponível em: https://www.voltimum.com.br/sites/www.voltimum.com.br/files/pdflibrary/02_livro_de_e_automacao_industrial.pdf. Acesso em: 28 mar. 2025.

RIBEIRO, Lucas; MALAGUTTI, T. Motores de passo: funcionamento e aplicações. Revista Linguagem Acadêmica, 2020. Ribeiro, L., & Malagutti, T. (2020). Motores de passo: funcionamento e aplicações. Revista Linguagem Acadêmica. acesso 13 de maio 2025.

Ribeiro, M. A. (1999). Automação industrial. Salvador:[sn]. RIBEIRO, Marco Antônio. Automação industrial. Salvador:[sn], 1999. acesso 11 maio 2025.

Ribeiro, M. A. (2005). Controle de Processo. Oitava edição Tek Treinamento & consultoria±2005. RIBEIRO, Marco Antônio. Controle de Processo. Oitava edição Tek Treinamento & consultoria±2005, 2005. acesso 11 maio 2025.

SANTOS, Guilherme Borges dos; SILVA, Raphael Lucas Elbl. Automatização de microambiente para desenvolvimento de culturas biológicas. 2015. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná. acesso 14 maio 2025.

SCHERZ, Paul. Practical electronics for inventors. McGraw-Hill, Inc., 2006. Scherz, P. (2006). Practical electronics for inventors. McGraw-Hill, Inc. acesso em: 16 abril 2026.

SENSOR CAPACITIVO PARA MEDIÇÃO DO NÍVEL DE ÁGUA”. Embrapa, [s.d.]. Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/105465/1/Sensor-capacitativo.pdf>. Acesso em: 18/10 /2025.

Sturaro, A. Z. (2009). AUTOMAÇÃO DE BOMBAS DOSADORAS COM AUXÍLIO DE CONTROLADOR LÓGICO PROGRAMÁVEL. STURARO, André Zelioli. AUTOMAÇÃO DE BOMBAS DOSADORAS COM AUXÍLIO DE CONTROLADOR LÓGICO PROGRAMÁVEL. 2009. acesso 13 maio 2025.

Sturaro, A. Z. (2009). AUTOMAÇÃO DE BOMBAS DOSADORAS COM AUXÍLIO DE CONTROLADOR LÓGICO PROGRAMÁVEL. STURARO, André Zelioli. AUTOMAÇÃO DE BOMBAS DOSADORAS COM AUXÍLIO DE CONTROLADOR LÓGICO PROGRAMÁVEL. 2009. acesso 16 maio 2025.

STURARO, André Zelioli. AUTOMAÇÃO DE BOMBAS DOSADORAS COM AUXÍLIO DE CONTROLADOR LÓGICO PROGRAMÁVEL. 2009. Sturaro, A. Z. (2009). AUTOMAÇÃO DE BOMBAS DOSADORAS COM AUXÍLIO DE CONTROLADOR LÓGICO PROGRAMÁVEL. acesso 17 maio 2025.

SHENZHEN XingKeChuang Technology Co., Ltd. “Operating principle of the capacitive liquid level sensor”. (2024-12-05) Disponível em: <https://www.xkc-sensor.com/info/1523.html>. Acesso em: 18/10/2025.

Tavares, João Manuel, João Nuno Matos. "Osciladores para rádio frequência de elevada estabilidade." *Eletrónica e Telecomunicações* 2.2 (1998): 169-173. TAVARES, João.

TECHNOLOGIES. IRF540N HEXFET® Power MOSFET. El Segundo, CA, [2004?]. Disponível em: https://www.infineon.com/dgdl/irf540n.pdf_irf540n.pdf/file_pdf.pdf. Acesso em: 19 out. 2025. Artigos e Publicações Online6. WILSON, Tony. Pulse Width Modulation (PWM) with a 555 Timer. All About Circuits, 22 ago. 2022. Disponível em: <https://www.allaboutcircuits.com/projects/pulse-width-modulation-pwm-with-a-555-timer/>. Acesso em: 19 out. 2025. LOVE, Bill. DC.

VIANNA SANTOS, GUILHERME. 2013. Como acionar um relé com arduino ou pic projeto e funcionamento. Disponível em: WAVESHARE WIKI. LCD1602 I2C Module. Disponível em: https://www.waveshare.com/wiki/LCD1602_I2C_Module. Acesso em: 19 out. 2025.[3] HACKSTER.IO. Learn To Use LCD 1602 (I2C & Parallel) With Arduino UNO . Disponível em: <https://www.hackster.io/Hack-star-Arduino/learn-to-use-lcd-1602-i2c-parallel-with-arduino-uno-f73f07>. Acesso em: 19 out. 2025.

<https://eletronicaemcasa.blogspot.com/2013/12/como-acionar-um-rele-com-arduino-ou-pic.html>. acesso em 25 abril 2025.

ANEXOS

i. Características Principais do Programa Arduino (Sketch):

ii. Linguagem Simplificada: C/C ++ (programa do misturador)

```
* PROJETO: Sistema Automatizado de Mistura de Líquidos v3.0 (Modificado)
* AUTOR: [programa TCC]
* DATA: 29/11/2025
*
* DESCRIÇÃO:
* Versão modificada do código para atender aos novos requisitos do usuário:
* 1. Bombas 1-4: 4s de dosagem com 10s de intervalo.
* 2. Motor de Mistura: 15s de agitação APÓS sensor de nível alto (S2) ser
ativado, com controle PWM.
* 3. Pausa de 5s após a agitação.
* 4. Bomba 5: 2s de enchimento para o Tanque 7.
*
* O código mantém a estrutura de MÁQUINA DE ESTADOS NÃO-BLOCANTE.
* -----
*/
```

```
// 1. INCLUSÃO DE BIBLIOTERAS
```

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

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

```
// 2. MAPEAMENTO DE PINOS E CONFIGURAÇÕES
```

```
// --- Pinos dos Relés (Saídas) ---
```

```
// ATENÇÃO: O pino 2 é necessário para a interrupção.
```

```
const int PINO_RELE_BOMBA_1 = 9;
```

```
const int PINO_RELE_BOMBA_2 = 3;
```

```
const int PINO_RELE_BOMBA_3 = 4;
```

```
const int PINO_RELE_BOMBA_4 = 5;
```

```
const int PINO_RELE_BOMBA_5_TANQUE_7 = 6; // Bomba 5 agora para Tanque
7
```

```
// Motor de Agitação movido para pino PWM (10)
```

```
const int PINO_MOTOR_AGITACAO_PWM = 11;
```

```
// --- Pinos dos Sensores e Botões (Entradas) ---
```

```
const int PINO_SENSOR_NIVEL_BAIXO = A1; // Sensor S1 (Vazio)
```

```
const int PINO_SENSOR_NIVEL_ALTO = A0; // Sensor S2 (Cheio)
```

```
const int PINO_BOTAO_INICIO = 11;
```

```
const int PINO_BOTAO_EMERGENCIA = 2; // PINO 2 SUPORTA
INTERRUPÇÃO NO UNO FANTASMA
```

```
// --- Configurações do Processo (NOVOS VALORES) ---
```

```
const unsigned long TEMPO_DOSAGEM_MS = 4000L; // 4 segundos
```

```
const unsigned long TEMPO_PAUSA_ENTRE_DOSAGEM_MS = 10000L; // 10
```

```

segundos
const unsigned long TEMPO_AGITACAO_MS = 15000L;    // 15 segundos
const unsigned long TEMPO_PAUSA_POS_AGITACAO_MS = 5000L; // 5
segundos
const unsigned long TEMPO_ENCHIMENTO_TANQUE_7_MS = 2000L; // 2
segundos

// --- Configuração do Motor PWM ---
const int VELOCIDADE_MOTOR_PWM = 180; // Valor entre 0 (desligado) e 255
(máximo)

// --- Configuração do Display LCD ---
LiquidCrystal_I2C lcd(0x27, 16, 2);

// --- Variáveis de Controle de Estado ---
enum EstadoSistema {
    PARADO,
    DOSANDO,
    PAUSA_DOSAGEM,
    AGUARDANDO_CHEIO, // Novo estado
    AGITANDO,
    PAUSA_POS_AGITACAO, // Novo estado
    ENCHENDO_TANQUE_7, // Novo estado
    ESVAZIANDO_TANQUE_MISTURA, // Estado original renomeado
    EMERGENCIA
};
EstadoSistema estadoAtual = PARADO;

volatile bool emergenciaAtivada = false; // Flag para interrupção
unsigned long tempoAnterior = 0;        // Variável para controle de tempo com
millis()
int bombaAtual = 0;                     // Controla qual bomba está dosando

// 3. FUNÇÃO DE EMERGÊNCIA (ISR)
void ISR_Emergencia() {
    // Apenas seta a flag. A lógica de desligamento é feita no loop principal.
    emergenciaAtivada = true;
}

// 4. SETUP
void setup() {
    // Configura pinos dos relés como SAÍDA
    pinMode(PINO_RELE_BOMBA_1, OUTPUT);
    pinMode(PINO_RELE_BOMBA_2, OUTPUT);
    pinMode(PINO_RELE_BOMBA_3, OUTPUT);
    pinMode(PINO_RELE_BOMBA_4, OUTPUT);
    pinMode(PINO_RELE_BOMBA_5_TANQUE_7, OUTPUT);
    pinMode(PINO_MOTOR_AGITACAO_PWM, OUTPUT); // Pino PWM

    // Garante que todos os relés e motor comecem desligados
    desligarTodosRele();

    // Configura pinos de entrada com resistor de pull-up interno
    pinMode(PINO_BOTAO_INICIO, INPUT_PULLUP);

```

```

pinMode(PINO_SENSOR_NIVEL_BAIXO, INPUT_PULLUP);
pinMode(PINO_SENSOR_NIVEL_ALTO, INPUT_PULLUP);
pinMode(PINO_BOTAO_EMERGENCIA, INPUT_PULLUP);

// Configura a interrupção para o botão de emergência no pino 2
attachInterrupt(digitalPinToInterrupt(PINO_BOTAO_EMERGENCIA),
ISR_Emergencia, FALLING);

// Inicializa o display LCD
lcd.init();
lcd.backlight();
mostrarMensagem("Sistema Pronto", "Aperte INICIAR");
}

// 5. LOOP PRINCIPAL
void loop() {
    // A primeira verificação é sempre o estado de emergência
    if (emergenciaAtivada) {
        gerenciarEstadoEmergencia();
        return; // Interrompe o loop normal para tratar a emergência
    }

    // Máquina de estados principal (não-blocante)
    switch (estadoAtual) {
        case PARADO:
            gerenciarEstadoParado();
            break;
        case DOSANDO:
            gerenciarEstadoDosando();
            break;
        case PAUSA_DOSAGEM:
            gerenciarEstadoPausa();
            break;
        case AGUARDANDO_CHEIO:
            gerenciarEstadoAguardandoCheio();
            break;
        case AGITANDO:
            gerenciarEstadoAgitando();
            break;
        case PAUSA_POS_AGITACAO:
            gerenciarEstadoPausaPosAgitacao();
            break;
        case ENCHENDO_TANQUE_7:
            gerenciarEstadoEnchendoTanque7();
            break;
        case ESVAZIANDO_TANQUE_MISTURA:
            gerenciarEstadoEsvaziandoTanqueMistura();
            break;
        // O estado de EMERGENCIA é tratado no início do loop
    }
}

// 6. FUNÇÕES DE GERENCIAMENTO DE ESTADOS

```

```

void gerenciarEstadoParado() {
    // Aguarda o botão de início ser pressionado
    if (digitalRead(PINO_BOTAO_INICIO) == LOW) {
        // Verifica se o tanque está vazio (sensor de nível baixo não detecta água)
        // HIGH = sensor não ativado = vazio
        if (digitalRead(PINO_SENSOR_NIVEL_BAIXO) == HIGH) {
            bombaAtual = 1; // Começa com a primeira bomba
            iniciarProximaDosagem();
        } else {
            // Pequeno delay aqui é aceitável, pois é apenas um feedback
            mostrarMensagem("Tanque nao esta", "vazio! (S1 ON)");
            delay(2000);
            mostrarMensagem("Sistema Pronto", "Aperte INICIAR");
        }
    }
}

```

```

void gerenciarEstadoDosando() {
    // Verifica se o tempo de dosagem da bomba atual já passou
    if (millis() - tempoAnterior >= TEMPO_DOSAGEM_MS) {
        desligarBomba(bombaAtual);

        // Se ainda houver bombas para dosar (1 a 4), entra em pausa
        if (bombaAtual < 4) {
            bombaAtual++;
            estadoAtual = PAUSA_DOSAGEM;
            tempoAnterior = millis(); // Inicia o cronômetro da pausa
            mostrarMensagem("Pausa (10s)...", "Prox: B" + String(bombaAtual));
        } else {
            // Se todas as bombas já dosaram, AGUARDA SENSOR CHEIO (S2)
            iniciarAguardandoCheio();
        }
    }
}

```

```

void gerenciarEstadoPausa() {
    // Aguarda o tempo de pausa entre as dosagens (10 segundos)
    if (millis() - tempoAnterior >= TEMPO_PAUSA_ENTRE_DOSAGEM_MS) {
        iniciarProximaDosagem(); // Inicia a dosagem da próxima bomba
    }
}

```

```

void gerenciarEstadoAguardandoCheio() {
    // Aguarda o sensor de nível alto (S2) ser ativado (LOW = ativado)
    if (digitalRead(PINO_SENSOR_NIVEL_ALTO) == LOW) {
        iniciarAgitacao();
    }
    // Se não estiver cheio, a mensagem é atualizada no loop
}

```

```

void gerenciarEstadoAgitando() {
    // Verifica se o tempo de agitação (15 segundos) já passou
    if (millis() - tempoAnterior >= TEMPO_AGITACAO_MS) {
        desligarMotorAgitacao(); // Desliga o motor
    }
}

```

```

    iniciarPausaPosAgitacao();
}
}

void gerenciarEstadoPausaPosAgitacao() {
    // Aguarda o tempo de pausa pós-agitação (5 segundos)
    if (millis() - tempoAnterior >= TEMPO_PAUSA_POS_AGITACAO_MS) {
        iniciarEnchimentoTanque7();
    }
}

void gerenciarEstadoEnchendoTanque7() {
    // Verifica se o tempo de enchimento do Tanque 7 (2 segundos) já passou
    if (millis() - tempoAnterior >= TEMPO_ENCHIMENTO_TANQUE_7_MS) {
        desligarBomba5(); // Desliga a Bomba 5

        // Transição para o estado original de esvaziamento (se necessário)
        // Se o esvaziamento não for mais necessário, o ciclo termina aqui.
        // Vou assumir que o ciclo termina, conforme a lógica fornecida.

        mostrarMensagem("Ciclo Concluído!", "Tanque 7 Cheio");
        delay(3000); // Pausa para mostrar a mensagem final

        estadoAtual = PARADO;
        mostrarMensagem("Sistema Pronto", "Aperte INICIAR");
    }
}

// O estado original de esvaziamento foi mantido, mas não é mais usado na nova
// lógica.
void gerenciarEstadoEsvaziandoTanqueMistura() {
    // O tanque é considerado vazio quando o sensor de nível baixo (S1) para de
    // detectar água
    // HIGH = sensor não ativado = vazio
    if (digitalRead(PINO_SENSOR_NIVEL_BAIXO) == HIGH) {
        digitalWrite(PINO_RELE_BOMBA_5_TANQUE_7, HIGH); // Desliga a bomba
        // de sucção

        mostrarMensagem("Esvaziamento OK!", "");
        delay(3000); // Pausa para mostrar a mensagem final

        estadoAtual = PARADO;
        mostrarMensagem("Sistema Pronto", "Aperte INICIAR");
    }
}

void gerenciarEstadoEmergencia() {
    // Entra no estado de emergência
    estadoAtual = EMERGENCIA;
    desligarTodosRele(); // Desliga TUDO imediatamente

    // Lógica de piscar a mensagem de emergência (não-blocante, mas com delay)
    // O delay aqui é aceitável pois o sistema já está parado e seguro.
    mostrarMensagem("! EMERGENCIA !", "REARME O BOTAO");
}

```

```

lcd.noBacklight();
delay(300);
lcd.backlight();
delay(300);

// Verifica se o botão de emergência foi liberado (desativado)
// INPUT_PULLUP: pino estará em HIGH quando o botão for solto
if (digitalRead(PINO_BOTAO_EMERGENCIA) == HIGH) {
    emergenciaAtivada = false; // Rearma a flag
    estadoAtual = PARADO; // Volta ao estado inicial seguro
    mostrarMensagem("Sistema Pronto", "Aperte INICIAR");
    delay(1000); // Pequena pausa para estabilizar
}
}

// 7. FUNÇÕES DE TRANSIÇÃO DE ESTADO

void iniciarProximaDosagem() {
    estadoAtual = DOSANDO;
    mostrarMensagem("Dosando Bomba " + String(bombaAtual),
tempoParaString(TEMPO_DOSAGEM_MS));
    ligarBomba(bombaAtual);
    tempoAnterior = millis(); // Inicia o cronômetro da dosagem
}

void iniciarAguardandoCheio() {
    estadoAtual = AGUARDANDO_CHEIO;
    mostrarMensagem("Aguardando Cheio", "Sensor S2...");
    // Não há ação de saída aqui, apenas espera pela entrada do sensor
}

void iniciarAgitacao() {
    estadoAtual = AGITANDO;
    mostrarMensagem("Agitando (PWM)",
tempoParaString(TEMPO_AGITACAO_MS));
    ligarMotorAgitacao(VELOCIDADE_MOTOR_PWM); // Liga o motor com PWM
    tempoAnterior = millis(); // Inicia o cronômetro da agitação
}

void iniciarPausaPosAgitacao() {
    estadoAtual = PAUSA_POS_AGITACAO;
    mostrarMensagem("Pausa (5s)", "Prox: B5 Tanque 7");
    tempoAnterior = millis(); // Inicia o cronômetro da pausa
}

void iniciarEnchimentoTanque7() {
    estadoAtual = ENCHENDO_TANQUE_7;
    mostrarMensagem("Enchendo Tanque 7",
tempoParaString(TEMPO_ENCHIMENTO_TANQUE_7_MS));
    ligarBomba5(); // Liga a Bomba 5
    tempoAnterior = millis(); // Inicia o cronômetro do enchimento
}

```

```

void iniciarEsvaziamentoTanqueMistura() {
    estadoAtual = ESVAZIANDO_TANQUE_MISTURA;
    mostrarMensagem("Esvaziando...", "Aguardando S1");
    // Se a Bomba 5 era usada para sucção, ela é ligada aqui.
    // digitalWrite(PINO_RELE_BOMBA_5_TANQUE_7, LOW);
}

```

// 8. FUNÇÕES UTILITÁRIAS

```

void ligarBomba(int numeroBomba) {
    // LOW liga o relé (assumindo módulo ativo em baixo)
    switch (numeroBomba) {
        case 1: digitalWrite(PINO_RELE_BOMBA_1, LOW); break;
        case 2: digitalWrite(PINO_RELE_BOMBA_2, LOW); break;
        case 3: digitalWrite(PINO_RELE_BOMBA_3, LOW); break;
        case 4: digitalWrite(PINO_RELE_BOMBA_4, LOW); break;
    }
}

```

```

void desligarBomba(int numeroBomba) {
    // HIGH desliga o relé (assumindo módulo ativo em baixo)
    switch (numeroBomba) {
        case 1: digitalWrite(PINO_RELE_BOMBA_1, HIGH); break;
        case 2: digitalWrite(PINO_RELE_BOMBA_2, HIGH); break;
        case 3: digitalWrite(PINO_RELE_BOMBA_3, HIGH); break;
        case 4: digitalWrite(PINO_RELE_BOMBA_4, HIGH); break;
    }
}

```

```

void ligarMotorAgitacao(int velocidade) {
    // Usa analogWrite para controle PWM
    analogWrite(PINO_MOTOR_AGITACAO_PWM, velocidade);
}

```

```

void desligarMotorAgitacao() {
    // Desliga o motor (PWM 0)
    analogWrite(PINO_MOTOR_AGITACAO_PWM, 0);
}

```

```

void ligarBomba5() {
    // Bomba 5 para Tanque 7 (LOW liga)
    digitalWrite(PINO_RELE_BOMBA_5_TANQUE_7, LOW);
}

```

```

void desligarBomba5() {
    // Bomba 5 para Tanque 7 (HIGH desliga)
    digitalWrite(PINO_RELE_BOMBA_5_TANQUE_7, HIGH);
}

```

```

void desligarTodosRele() {
    // Assume que HIGH desliga o relé.
    digitalWrite(PINO_RELE_BOMBA_1, HIGH);
    digitalWrite(PINO_RELE_BOMBA_2, HIGH);
    digitalWrite(PINO_RELE_BOMBA_3, HIGH);
}

```

```
digitalWrite(PINO_RELE_BOMBA_4, HIGH);  
digitalWrite(PINO_RELE_BOMBA_5_TANQUE_7, HIGH);  
desligarMotorAgitacao(); // Desliga o motor PWM  
}
```

```
void mostrarMensagem(String linha1, String linha2) {  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print(linha1);  
    lcd.setCursor(0, 1);  
    lcd.print(linha2);  
}
```

```
String tempoParaString(unsigned long tempo_ms) {  
    int segundos = tempo_ms / 1000;  
    return String(segundos) + "s";  
}
```

Fonte: Wilson/2025

**APÊNDICE A - CHECKLIST DE TESTES - SISTEMA DE BOMBAS
PERISTÁLTICAS PWM**

Etapa	Descrição do Teste	Status	Observações
1	Verificar fixação mecânica das bombas		
2	Conferir polaridade dos fios (vermelho = +, azul = -)		
3	Testar continuidade elétrica dos cabos		
4	Checar vedação das mangueiras e conexões		
5	Ligar fonte 12V DC e verificar tensão real		
6	Acionar cada bomba manualmente		
7	Testar controle PWM (variação de velocidade)		
8	Verificar resposta do relé/MOSFET ao comando		
9	Medir corrente de operação individual (A)		
10	Avaliar ruído e vibração durante funcionamento		
11	Testar fluxo e direção do líquido		
12	Verificar temperatura do motor após 5 min		
13	Checar funcionamento simultâneo de todas as bombas		
14	Confirmar desligamento e ausência de vazamentos		
15	Registrar observações finais		

Fonte: Autores, 2025.

APÊNDICE B - TABELA DE CONEXÕES ELÉTRICAS – SISTEMA DE BOMBA PWM

<u>Item</u>	<u>Componente</u>	<u>Pino/Terminal</u>	<u>Conexão Destino</u>	<u>Descrição da Função</u>	<u>Alimentação (V)</u>	<u>Observações</u>
<u>1</u>	<u>Bomba 1 (M1)</u>	<u>Fio Vermelho</u>	<u>COM do Relé 1</u>	<u>Alimentação positiva da bomba</u>	<u>+12 V</u>	<u>Controlada pelo Relé 1</u>
<u>2</u>	<u>Bomba 1 (M1)</u>	<u>Fio Azul</u>	<u>GND comum</u>	<u>Retorno negativo</u>	<u>0 V</u>	<u>GND compartilhado</u>
<u>3</u>	<u>Bomba 2 (M2)</u>	<u>Fio Vermelho</u>	<u>COM do Relé 2</u>	<u>Alimentação positiva da bomba</u>	<u>+12 V</u>	<u>Controlada pelo Relé 2</u>
<u>4</u>	<u>Bomba 2 (M2)</u>	<u>Fio Azul</u>	<u>GND comum</u>	<u>Retorno negativo</u>	<u>0 V</u>	
<u>5</u>	<u>Bomba 3 (M3)</u>	<u>Fio Vermelho</u>	<u>COM do Relé 3</u>	<u>Alimentação positiva da bomba</u>	<u>+12 V</u>	<u>Controlada pelo Relé 3</u>
<u>6</u>	<u>Bomba 3 (M3)</u>	<u>Fio Azul</u>	<u>GND comum</u>	<u>Retorno negativo</u>	<u>0 V</u>	
<u>7</u>	<u>Bomba 4 (M4)</u>	<u>Fio Vermelho</u>	<u>COM do Relé 4</u>	<u>Alimentação positiva da bomba</u>	<u>+12 V</u>	<u>Controlada pelo Relé 4</u>
<u>8</u>	<u>Bomba 4 (M4)</u>	<u>Fio Azul</u>	<u>GND comum</u>	<u>Retorno negativo</u>	<u>0 V</u>	
<u>9</u>	<u>Bomba 5 (M5)</u>	<u>Fio Vermelho</u>	<u>COM do Relé 5</u>	<u>Alimentação positiva da bomba</u>	<u>+12 V</u>	<u>Controlada pelo Relé 5</u>
<u>10</u>	<u>Bomba 5 (M5)</u>	<u>Fio Azul</u>	<u>GND comum</u>	<u>Retorno negativo</u>	<u>0 V</u>	
<u>11</u>	<u>Relé 1</u>	<u>IN1</u>	<u>Pino D2 do Arduino</u>	<u>Comando digital de acionamento</u>	<u>5 V (sinal)</u>	
<u>12</u>	<u>Relé 2</u>	<u>IN2</u>	<u>Pino D3 do Arduino</u>	<u>Comando digital</u>	<u>5 V (sinal)</u>	
<u>13</u>	<u>Relé 3</u>	<u>IN3</u>	<u>Pino D4 do Arduino</u>	<u>Comando digital</u>	<u>5 V (sinal)</u>	
<u>14</u>	<u>Relé 4</u>	<u>IN4</u>	<u>Pino D5 do Arduino</u>	<u>Comando digital</u>	<u>5 V (sinal)</u>	
<u>15</u>	<u>Relé 5</u>	<u>IN5</u>	<u>Pino D6 do Arduino</u>	<u>Comando digital</u>	<u>5 V (sinal)</u>	
<u>16</u>	<u>Módulo Relé</u>	<u>VCC</u>	<u>5 V Arduino</u>	<u>Alimentação lógica do módulo</u>	<u>+5 V</u>	
<u>17</u>	<u>Módulo Relé</u>	<u>GND</u>	<u>GND Arduino</u>	<u>Referência comum</u>	<u>0 V</u>	

<u>18</u>	<u>Fonte DC 12 V</u>	<u>+12 V</u>	<u>Bombas (via COM dos relés)</u>	<u>Alimentação principal das bombas</u>	<u>+12 V</u>	<u>3 A mínimo</u>
<u>19</u>	<u>Fonte DC 12 V</u>	<u>GND</u>	<u>Arduino GND e Bombas</u>	<u>Referência comum do sistema</u>	<u>0 V</u>	<u>Unificar GND</u>
<u>20</u>	<u>Arduino UNO/Nano</u>	<u>Vin</u>	<u>(opcional, se usar 12 V)</u>	<u>Alimentação do microcontrolador</u>	<u>7–12 V</u>	<u>ou via USB 5 V</u>

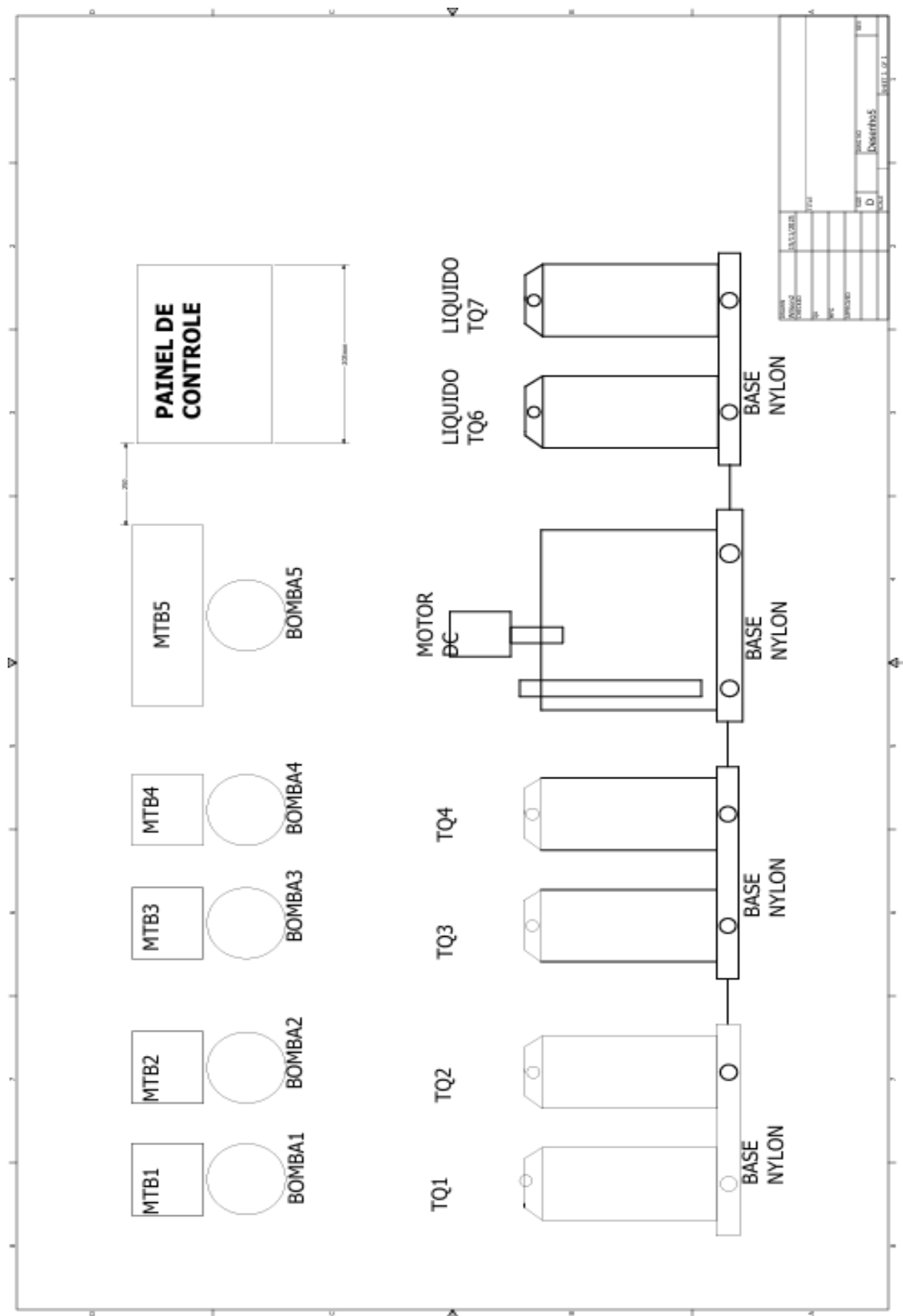
Fonte: Autores, 2025.

APÊNDICE C - ENDEREÇAMENTO E FUNÇÃO DE CADA RELÉ

<u>Relé</u>	<u>Endereço (Pino Arduino)</u>	<u>Bomba Associada</u>	<u>Estado Ativo</u>
<u>Relé 1</u>	<u>D2</u>	<u>Bomba 1</u>	<u>LOW (ativo em nível baixo)</u>
<u>Relé 2</u>	<u>D3</u>	<u>Bomba 2</u>	<u>LOW</u>
<u>Relé 3</u>	<u>D4</u>	<u>Bomba 3</u>	<u>LOW</u>
<u>Relé 4</u>	<u>D5</u>	<u>Bomba 4</u>	<u>LOW</u>
<u>Relé 5</u>	<u>D6</u>	<u>Bomba 5</u>	<u>LOW</u>

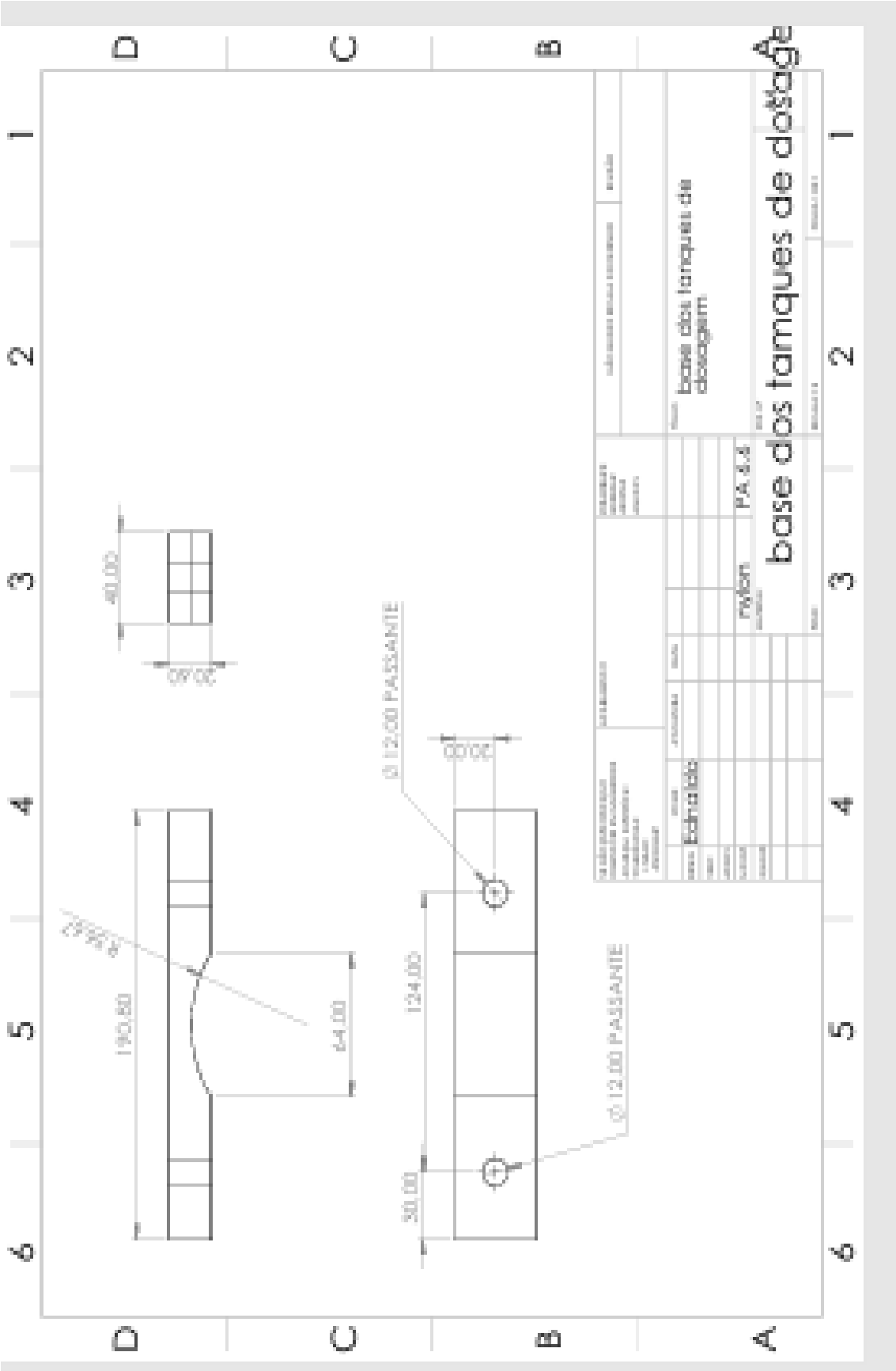
Fonte: Autores, 2025.

APÊNDICE D - PAINEL DO PROJETO



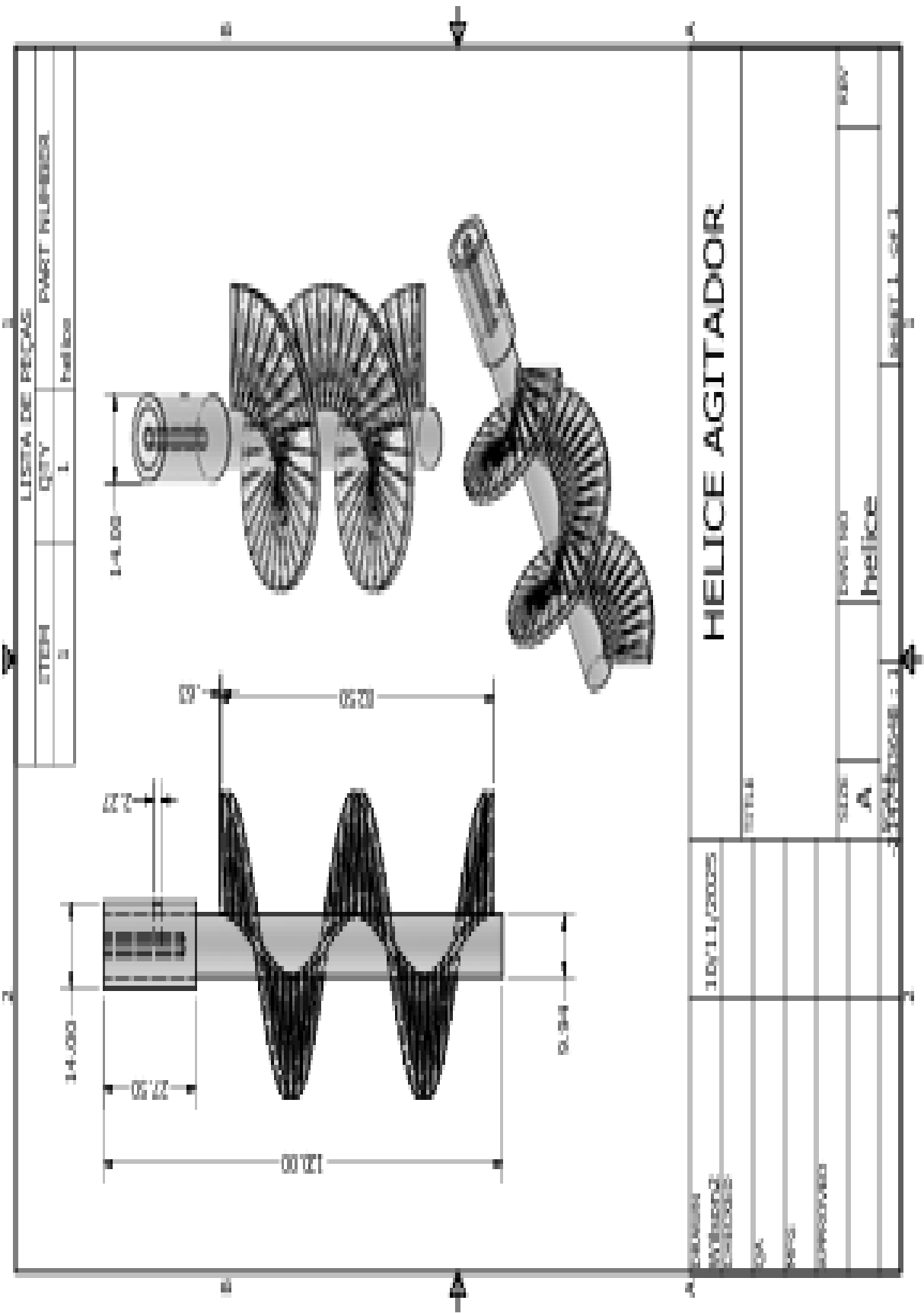
Fonte: Autores, 2025.

APÊNDICE E - BASE DOS TANQUES DE DOSAGEM



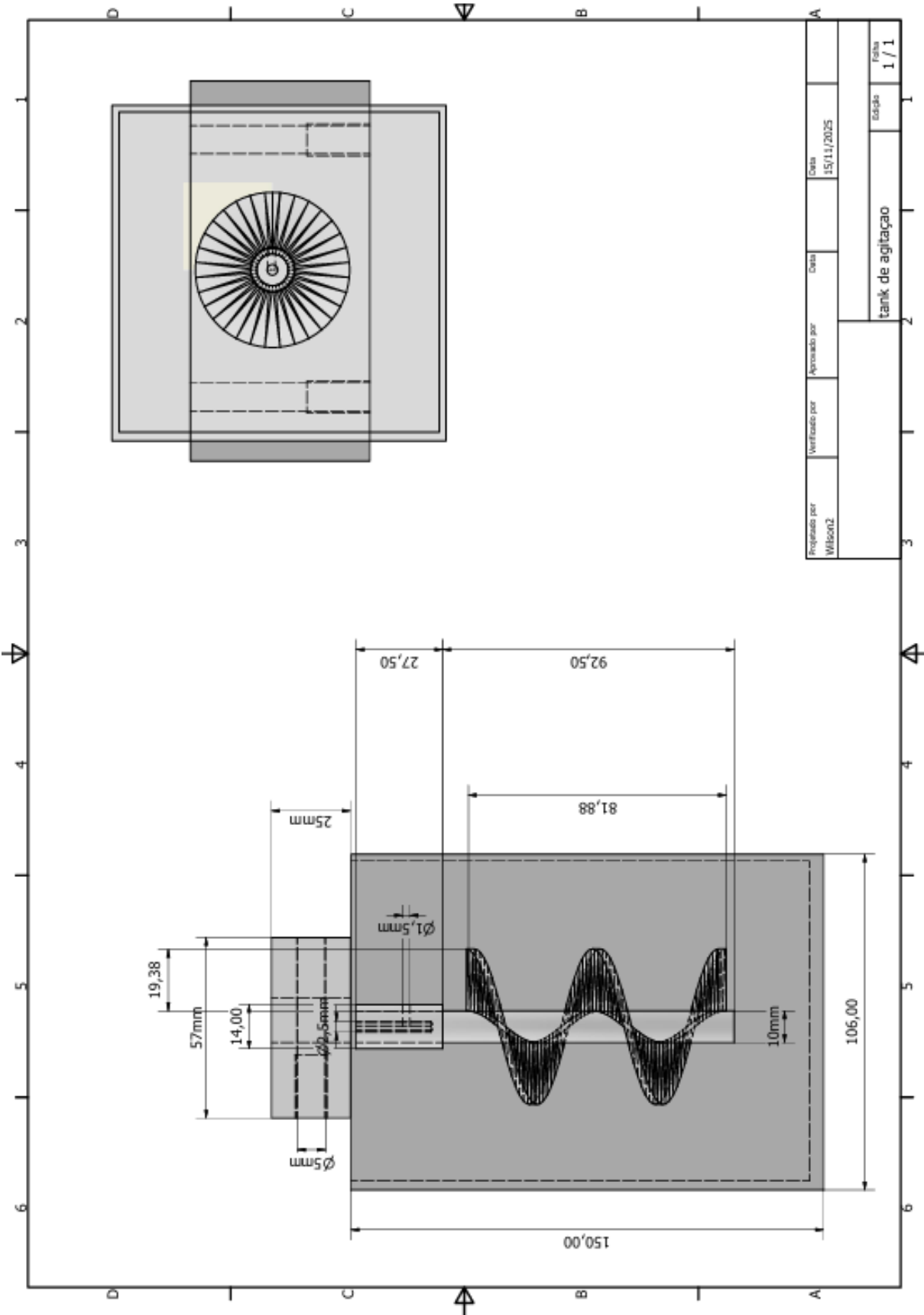
Fonte: Autores, 2025.

APÊNDICE G - HELICE DE AGITAÇÃO



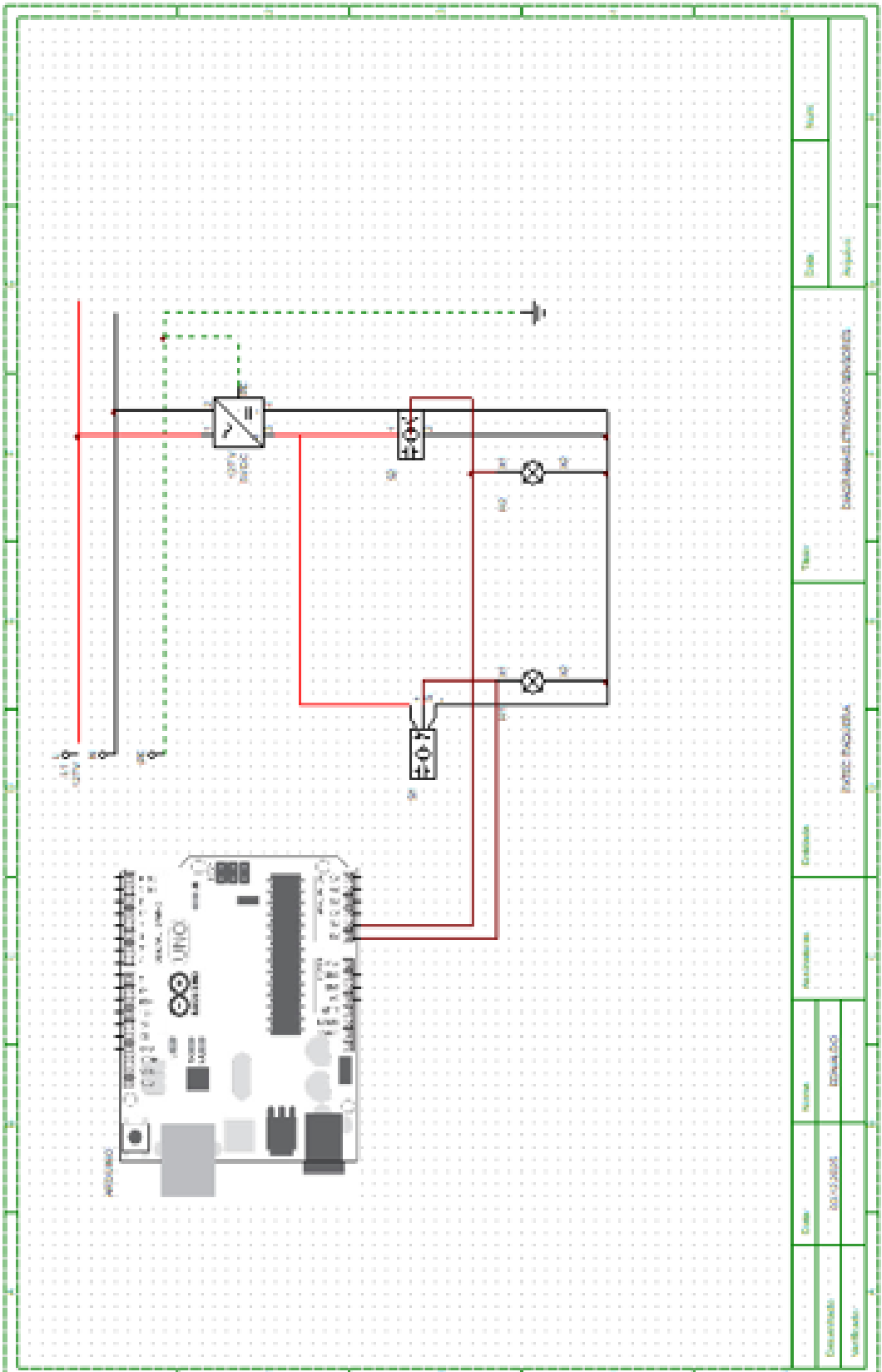
Fonte: Autores, 2025.

APÊNDICE H - TANQUE DE AGITAÇÃO



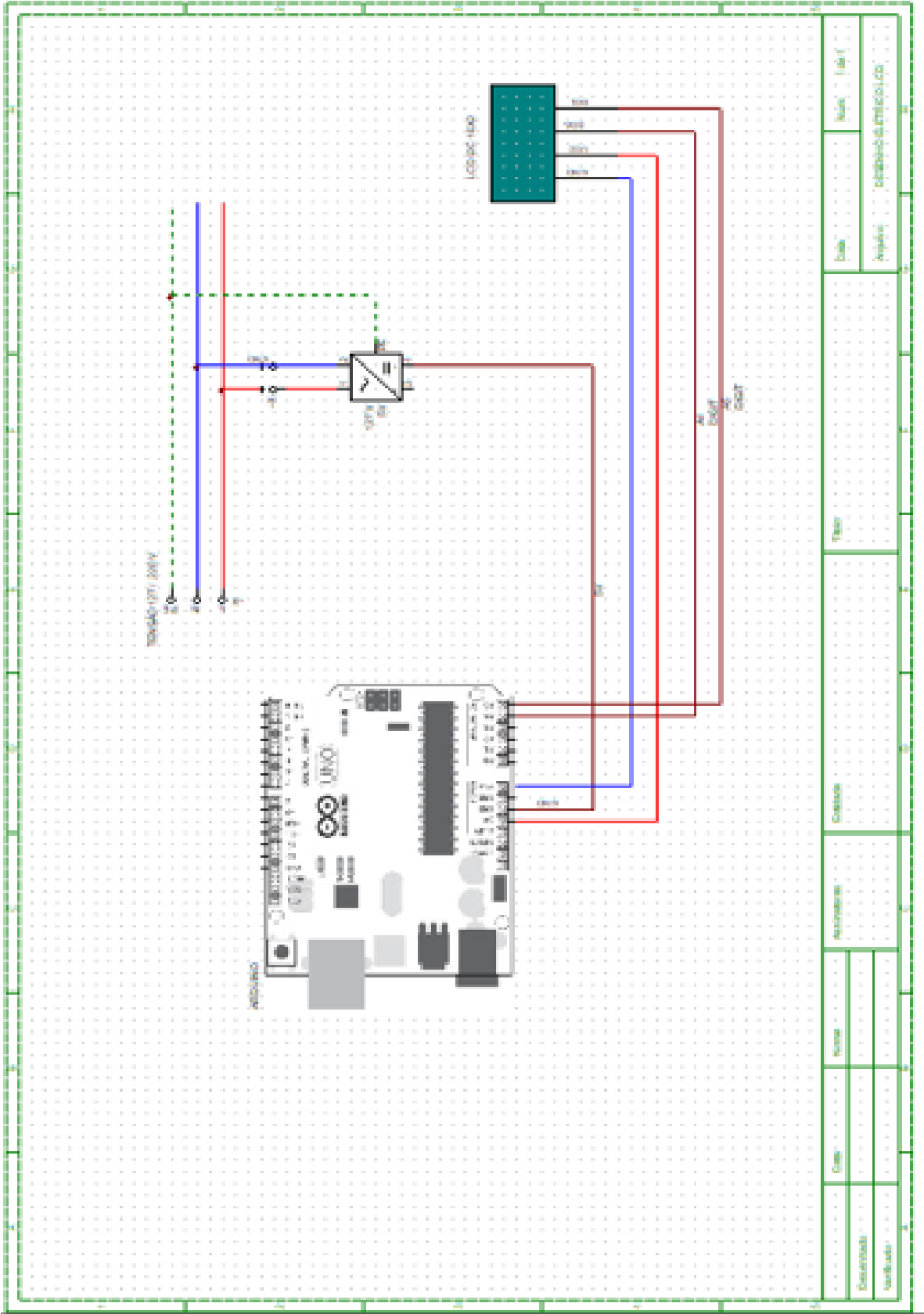
Fonte: Autores, 2025.

APÊNDICE I - DIAGRAMA ELÉTRICO FUNCIONAMENTO DOS SENSORES DE NIVEL, ELABORADO NO CADE_SIMU

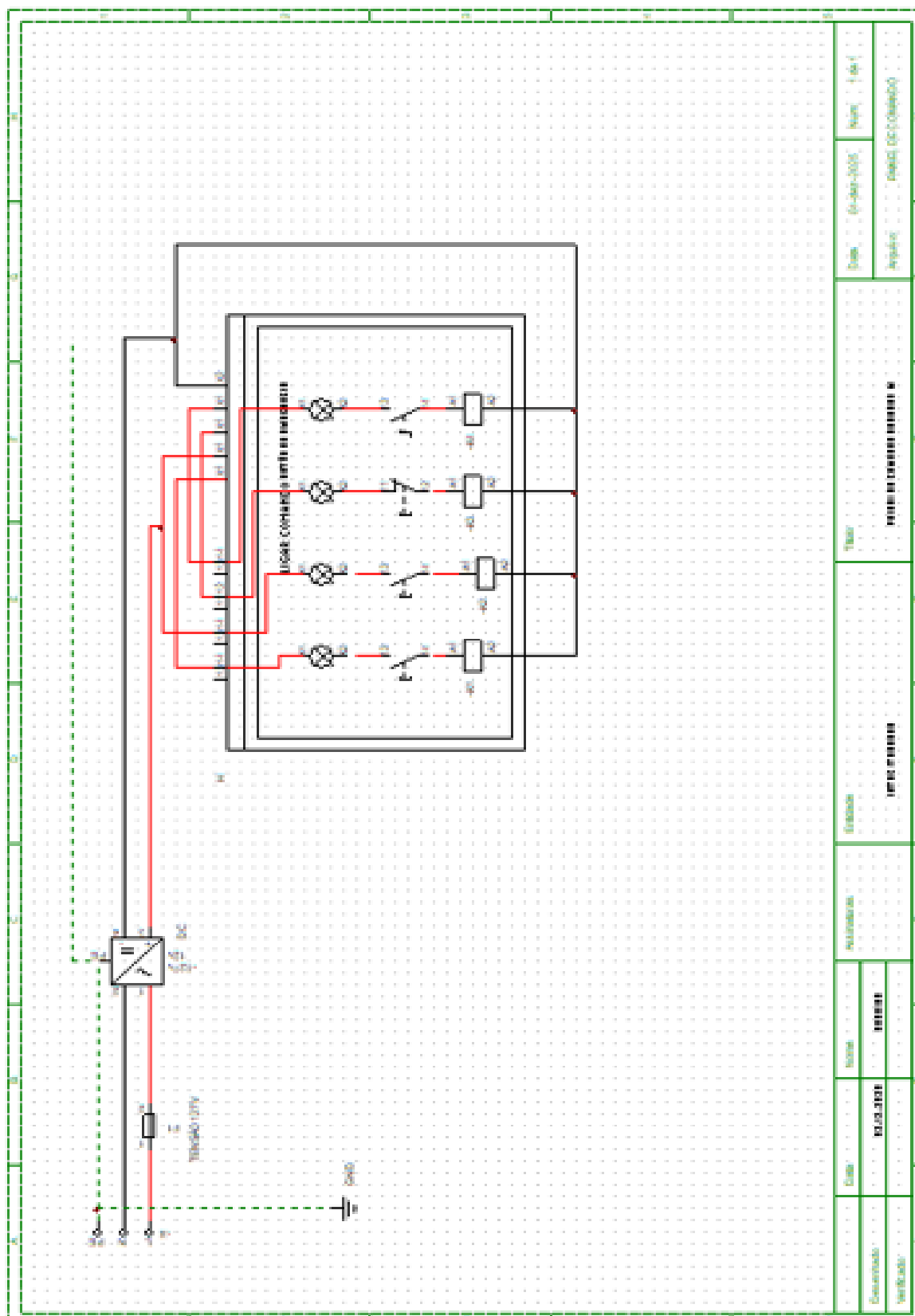


Fonte: Autores, 2025.

APÊNDICE J - DIAGRAMA ELÉTRICO LCD I2C LÓGICA ARDUINO

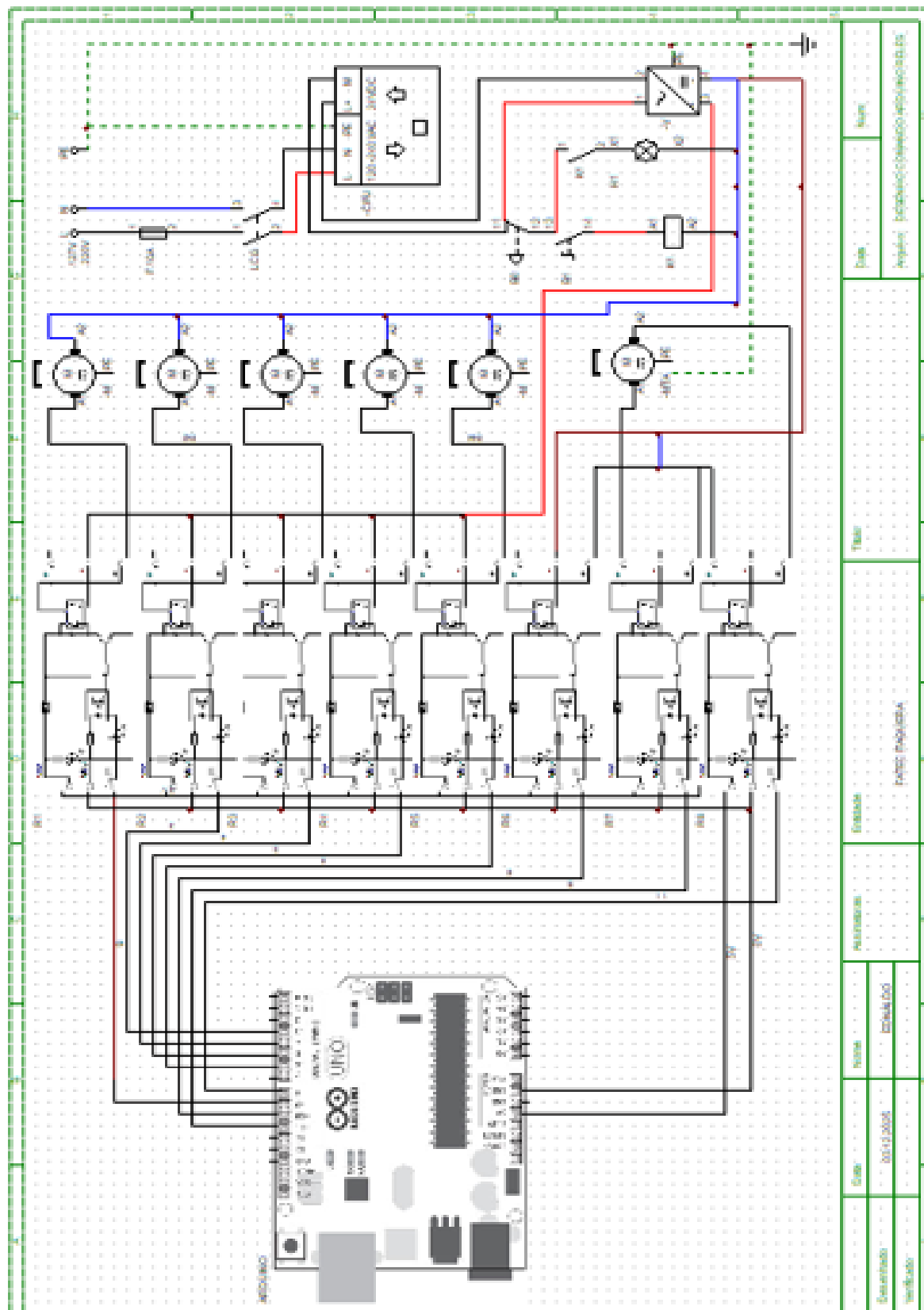


**APÊNDICE K - DIAGRAMA ELÉTRICO PAINEL DE COMANDO
DESENVOLVIDO NO, SOFTWARE CADE_SIMU**



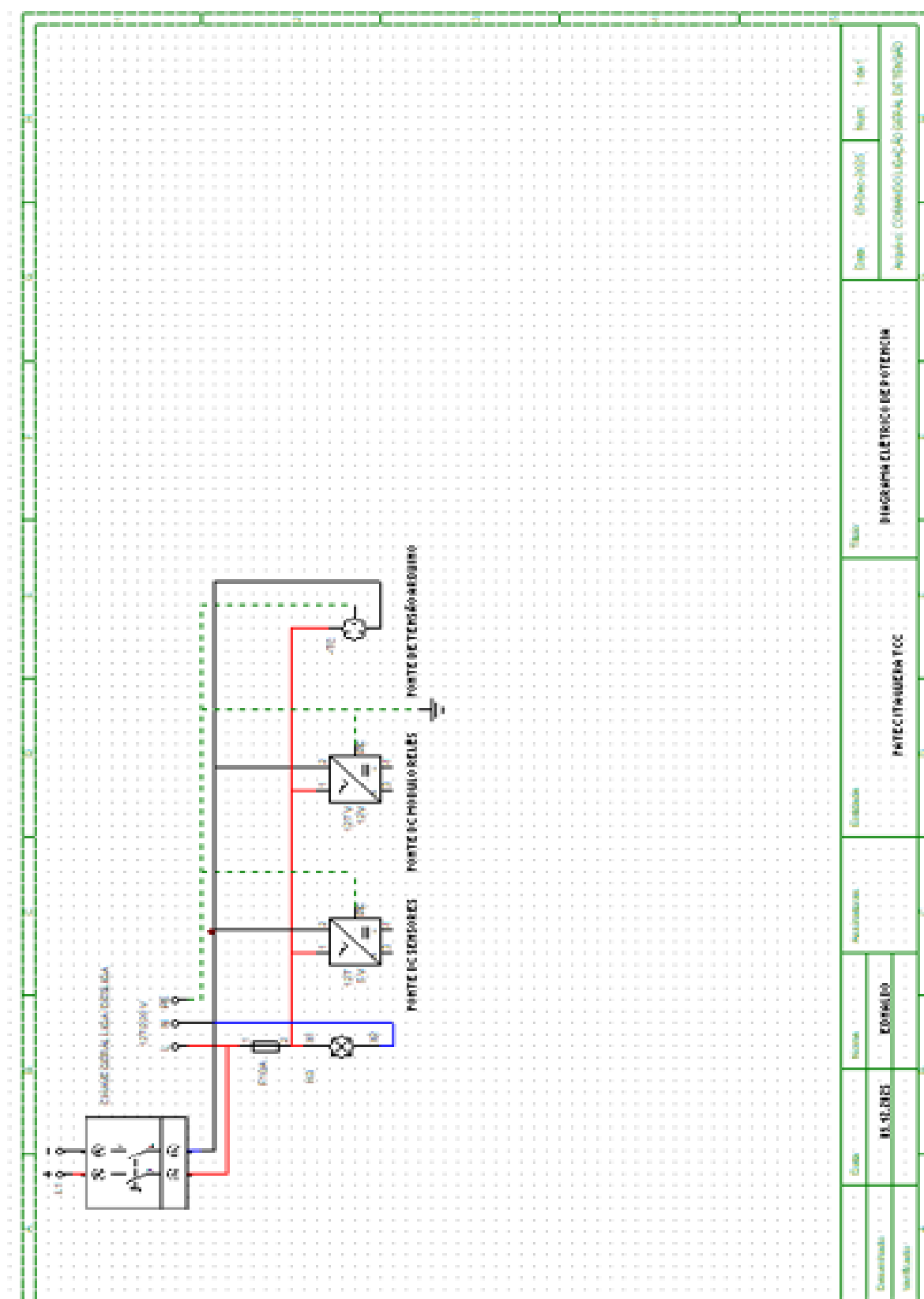
Fonte: Autores, 2025.

**APÊNDICE L - DIAGRAMA DO CIRCUITO ELÉTRONICO ARDUINO
LÓGICA BLOCO DE RELÉS 8 CANAIS, DESENVOLVIDO NO SOFTWARE
CADE_SIMU**



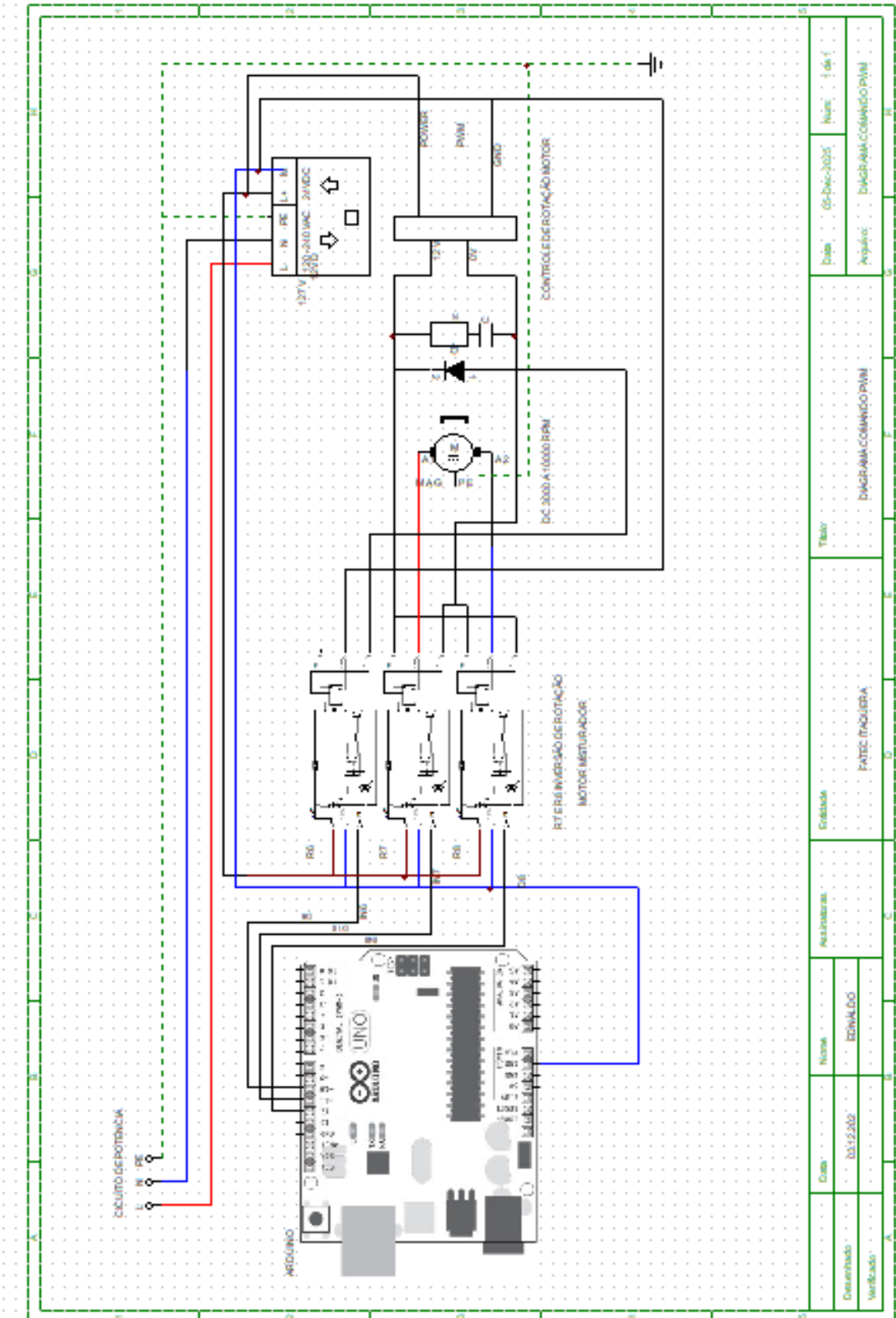
Fonte: Autores, 2025.

APÊNDICE M - DIAGRAMA ELÉTRICO DE POTÊNCIA, PARA LIGAÇÃO DO PAINEL DE COMANDO DESENVOLVIDO NO CADE_SIMU



Fonte: Autores, 2025.

APÊNDICE O – DIAGRAMA DE COMANDO CONTROLE DE VELOCIDADE VIA PWM E CHAVEAMENTO DE RELES R7 E R8, DSENVOLVIDO DO SOFTWER CADESIMU



Fonte: Autores, 2025.

APÊNDICE P - FICHA TÉCNICA DOS PARAFUSOS FICHA TÉCNICA – PARAFUSOS MÉTRICOS M3 / M10 / M12

PARAFUSO MÉTRICO M3

Normas aplicáveis: ABNT NBR ISO 261, NBR ISO 965-1/2, NBR ISO 898-1

Parâmetro	Valor
Diâmetro nominal (d)	3,0 mm
Passo	0,5 mm
Furo de passagem	3,4 mm
Chave sextavada (DIN 912)	2,5 mm
Comprimentos usuais	6 a 40 mm

Características Mecânicas (Classe 8.8)

\\	Valor
Resistência à tração mínima	800 MPa
Limite de escoamento	640 MPa
Alongamento mínimo	12%
Torque recomendado	≈ 0,6 N·m

Aplicações: Montagens leves, eletrônica, fixações de precisão.

PARAFUSO MÉTRICO M10

Normas aplicáveis: ABNT NBR ISO 261, NBR ISO 965-1/2, NBR ISO 898-1, DIN 931/933, NBR 6323

Parâmetro	Valor
Diâmetro nominal (d)	10 mm
Passo	1,5 mm
Furo de passagem	11 mm
Chave sextavada (DIN 933)	17 mm
Comprimentos usuais	20 a 120 mm

Características Mecânicas (Classe 8.8)

Propriedade	Valor
Resistência à tração mínima	800 MPa
Limite de escoamento	640 MPa
Alongamento mínimo	12%
Torque recomendado (seco)	49–50 N·m
Torque recomendado (lubrificado)	≈ 35 N·m

Aplicações: Fixações industriais, suportes, máquinas, estruturas metálicas.

PARAFUSO MÉTRICO M12

Normas aplicáveis: ABNT NBR ISO 261, NBR ISO 965-1/2, NBR ISO 898-1, ISO 3506-1,

DIN 931/933

Parâmetro	Valor
Diâmetro nominal (d)	12 mm
Passo	1,75 mm
Furo de passagem	13 mm
Chave sextavada	19 mm
Comprimentos usuais	30 a 200 mm

Propriedade	Valor
Resistência à tração mínima	1000 MPa
Limite de escoamento	900 MPa
Alongamento mínimo	9%
Torque recomendado (seco)	100–110 N·m
Torque recomendado (lubrificado)	» 75–80 N·m

Características Mecânicas (Classe 10.9)

Aplicações: Estruturas pesadas, fixações sujeitas a altos esforços e vibrações.

Referências Técnicas

ABNT NBR ISO 261:2016 – Roscas métricas ISO – Dimensões básicas.

ABNT NBR ISO 965-1/2 – Tolerâncias de roscas métricas ISO.

ABNT NBR ISO 898-1:2023 – Propriedades mecânicas dos fixadores de aço carbono e liga.

ISO 3506-1:2020 – Aço inoxidável para fixadores.

DIN 931 / DIN 933 – Parafusos com cabeça sextavada.

ABNT NBR 6323:2017 – Revestimento de zinco por imersão a quente.

CRV Industrial – Tabelas de torque e equivalência (Brasil, 2023).

APÊNDICE Q - FICHA TÉCNICA DAS PORCAS M10, M12

Pesquisa Completa – Porcas M10 e M12

Este documento apresenta um estudo técnico completo sobre as porcas de rosca métrica M10 e M12, com base em normas internacionais ISO, DIN e ABNT. São abordadas suas características dimensionais, propriedades mecânicas, aplicações, materiais, acabamentos e valores de torque orientativos. As informações seguem referências seguras e amplamente reconhecidas no setor da mecânica e engenharia industrial.

Porca M10

Características gerais:

Rosca métrica M10 com passo grosso de 1,5 mm, utilizada em fixações médias. Apresenta boa resistência mecânica e ampla disponibilidade no mercado industrial e automotivo.

Parâmetro	Símbolo	Valor (mm)
Diâmetro nominal	M	10
Passo da rosca	P	1,5
Largura entre faces	s	17
Altura da porca (espessura)	m	8

Materiais e acabamentos:

Aço carbono, aço inoxidável (A2/304 e A4/316), latão e porcas autotravantes com inserto de nylon (DIN 985).

Acabamentos comuns: zincagem branca, galvanização, niquelagem, negro-óxido e passivação (inox).

Propriedades mecânicas:

As propriedades seguem a norma ISO 898-2, que define as classes de resistência e ensaios de dureza e carga de prova. A porca M10 deve ser compatível com a classe do parafuso (por exemplo, 8.8 ou 10.9).

Tabela de torque (M10):

Classe do Parafuso	Torque de Aperto (N·m)	Observações
6.8	30 – 35	Aplicações leves
8.8	40 – 50	Padrão industrial
10.9	55 – 65	Alta resistência
12.9	70 – 80	Uso severo, verificar lubrificação

Aplicações mecânicas:

Usada em montagens estruturais, equipamentos leves, fixação de suportes metálicos, máquinas, móveis e componentes automotivos. Possui equilíbrio entre resistência e peso.

Normas e referências:

ISO 4032 – Porcas hexagonais com rosca métrica (dimensões)

DIN 934 – Porcas hexagonais padrão

ISO 4035 – Porcas hexagonais finas

ISO 898-2 – Propriedades mecânicas das porcas de aço

DIN 985 – Porcas autotravantes com inserto de nylon

ABNT NBR ISO 4032 – Versão brasileira da norma ISO 4032

Porca M12

Características gerais:

Rosca métrica M12 com passo grosso de 1,75 mm. Indicada para fixações de média a alta resistência, com ampla aplicação em estruturas metálicas e montagens industriais.

Parâmetro	Símbolo	Valor (mm)
Diâmetro nominal	M	12
Passo da rosca	P	1,75
Largura entre faces	s	19
Altura da porca (espessura)	m	10

Materiais e acabamentos:

Aço carbono, aço inoxidável (A2/304 e A4/316), latão e porcas com inserto de nylon (autotravantes).

Acabamentos usuais: zincagem, galvanização, niquelagem, negro-óxido e passivação.

Propriedades mecânicas:

De acordo com a norma ISO 898-2. Deve ser combinada com parafusos de classe compatível (8.8, 10.9 ou 12.9). As propriedades incluem dureza mínima, carga de prova e ensaio de resistência à tração.

Tabela de torque (M12):

Classe do Parafuso	Torque de Aperto (N·m)	Observações
6.8	55 – 65	Aplicações médias
8.8	70 – 90	Padrão estrutural
10.9	100 – 115	Alta resistência
12.9	120 – 130	Uso severo, verificar atrito e lubrificação

Aplicações mecânicas:

Empregada em montagens estruturais pesadas, conexões metálicas, fixações automotivas de chassis, máquinas industriais e estruturas que exigem alto torque e segurança **mecânica**.

Normas e referências:

ISO 4032 – Porcas hexagonais com rosca métrica (dimensões)

DIN 934 – Porca hexagonal padrão

ISO 4035 – Porcas finas

ISO 898-2 – Propriedades mecânicas das porcas

DIN 985 – Porcas autotravantes com inserto de nylon

ABNT NBR ISO 4032 – Versão brasileira da norma ISO 4032.

Fonte: Catálogos Técnicos, MG_Parafusos,20925

APÊNDICE R - FICHA TÉCNICA DOS PARAFUSOS

Validação da Lógica do Código Modificado (v3.0)

O código foi modificado para atender aos requisitos do usuário, incluindo a remoção completa da funcionalidade de emergência. A lógica da Máquina de Estados Não-Blocante foi revisada para incorporar os novos tempos e a sequência de eventos.

Sequência de Estados e Transições

Sequência de Estados e Transições

Estado Atual	Condição de Transição	Próximo Estado	Ações de Saída do Estado
PARADO	Botão INICIO (PINO 12) pressionado E Sensor S1 (PINO A0) NÃO ativado (Tanque Vazio)	DOSANDO	<code>bombaAtual = 1 ; iniciarProximaDosagem()</code>
DOSANDO	<code>millis() - tempoAnterior</code> <code>>=</code> <code>TEMPO_DOSAGEM_MS</code> (4s)	PAUSA_DOSAGEM (se <code>bombaAtual < 4</code>) ou AGUARDANDO_CHEIO (se <code>bombaAtual == 4</code>)	<code>desligarBomba(bombaAtual)</code>
PAUSA_DOSAGEM	<code>millis() - tempoAnterior</code> <code>>=</code> <code>TEMPO_PAUSA_ENTRE_DOSAGEM_MS</code> (10s)	DOSANDO	<code>bombaAtual++ ; iniciarProximaDosagem()</code>
AGUARDANDO_CHEIO	Sensor S2 (PINO A1) ativado (LOW)	AGITANDO	<code>iniciarAgitacao()</code>

TEMPO_ENCHIMENTO_TANQUE_7_MS (2s)

Parâmetros de Tempo (Constantes)

Constante	Valor	Descrição
TEMPO_DOSAGEM_MS	4000L	Duração de funcionamento das Bombas 1 a 4.
TEMPO_PAUSA_ENTRE_DOSAGEM_MS	10000L	Intervalo entre o funcionamento das Bombas 1 a 4.
TEMPO_AGITACAO_MS	15000L	Duração de funcionamento do Motor de Mistura.
TEMPO_PAUSA_POS_AGITACAO_MS	5000L	Intervalo após a mistura.
TEMPO_ENCHIMENTO_TANQUE_7_MS	2000L	Duração de funcionamento da Bomba 5.
VELOCIDADE_MOTOR_PWM	180	Velocidade do Motor de Mistura (0-255).

AGITANDO	<code>millis() - tempoAnterior</code> <code>>=</code> <code>TEMPO_AGITACAO_MS</code> (15s)	PAUSA_POS_AGITACAO	<code>desligarMotorAgitacao()</code>
PAUSA_POS_AGITACAO	<code>millis() - tempoAnterior</code> <code>>=</code> <code>TEMPO_PAUSA_POS_AGITACAO_MS</code> (5s)	ENCHENDO_TANQUE_7	<code>iniciarEnchimentoTanque7()</code>
ENCHENDO_TANQUE_7	<code>millis() - tempoAnterior</code> <code>>=</code>	PARADO	<code>desligarBomba5()</code>

Remoção da Emergência

Todas as referências ao pino 2 (função Intervalo após a mistura. Duração de funcionamento da Bomba 5. Velocidade do Motor de Mistura (0-255). PINO_BOTAO_EMERGENCIA), à variável ISR_Emergencia() , ao estado emergenciaAtivada , à EMERGENCIA e à função gerenciarEstadoEmergencia() foram removidas ou comentadas, garantindo a exclusão completa da funcionalidade de

emergência do código. A lógica está consistente com os requisitos mais recentes

Fonte: Catálogos Técnicos.