



**ESCOLA TÉCNICA ESTADUAL  
DEP. ARY DE CAMARGO PEDROSO  
TÉCNICO EM AUTOMAÇÃO INDUSTRIAL**

Ivan Carlos Alves Souza

Leonardo Vieira Ferreira da Silva

Luis Fernando Dias Lopes

Orlando Gustavo Orlovski

Pedro Henrique Cirizola Alves Coelho

**ALIMENTADOR AUTOMÁTICO PARA  
*PETS* DOMÉSTICOS**

**Piracicaba**

**2023**

Ivan Carlos Alves Souza  
Leonardo Vieira Ferreira da Silva  
Luis Fernando Dias Lopes  
Orlando Gustavo Orlovski  
Pedro Henrique Cirizola Alves Coelho

**ALIMENTADOR AUTOMÁTICO PARA  
*PETS* DOMÉSTICOS**

Trabalho de Conclusão de Curso da Etec  
Deputado Ary de Camargo Pedroso,  
orientado pelo Prof. Adilson Antonio  
Rodrigues, apresentado como requisito  
parcial para obtenção do título de Técnico  
em Automação Industrial

**Piracicaba**

**2023**

## **DEDICATORIA**

Aos nossos familiares e a todos que nos ajudaram nessa trajetória.

## **AGRADECIMENTOS**

Gostaríamos de expressar nossa profunda gratidão aos professores Adilson Rodrigues e Matheus Altafini por sua orientação, apoio e contribuições ao longo do desenvolvimento deste trabalho. Suas orientações e críticas foram essenciais para aprimorar nosso trabalho até seu pleno resultado. Por fim, expressamos nossa gratidão aos nossos amigos, familiares e todos os que, de alguma forma, nos apoiaram e encorajaram ao longo desta jornada. Seu apoio moral e compreensão foram fundamentais para enfrentarmos os desafios e alcançarmos nossos objetivos.

## EPÍGRAFE

*“A grandeza de uma nação pode ser julgada pelo modo que seus animais são tratados”.*

MAHATMA GANDHI

## RESUMO

Este trabalho aborda o desenvolvimento de um alimentador automático de animais com ênfase na facilidade de uso com o objetivo de desenvolver um dispositivo que facilitasse a vida dos proprietários. Nesse contexto, o alimentador automático de animais emerge, sendo construído com a utilização de materiais reaproveitados e dispositivos eletrônicos. A metodologia aplicada ao projeto se baseou na utilização de componentes mecânicos e eletrônicos, dentre eles o Arduino, display LCD, motor de passo e sensor ultrassônico, com o conjunto desses aparelhos realizando a função de fazer o funcionamento da rosca transportadora. Foi elaborada uma programação com o intuito de construir uma interface entre o usuário e o dispositivo, permitindo assim seu controle por meio do Display LCD. O uso dessa metodologia alcançou resultados satisfatórios, possibilitando a criação de um protótipo funcional, que realizou o objetivo proposto no início do projeto.

**Palavras-Chave:** Alimentador. Sustentabilidade. Display. Automatização. Animais. Arduino.

## **ABSTRACT**

This work addresses the development of an automatic animal feeder with an emphasis on ease of use with the aim of developing a device that would make life easier for owners. In this context, the automatic animal feeder emerges, being built using reused materials and electronic devices. The methodology applied to the project was based on the use of mechanical and electronic components, including the Arduino, LCD display, stepper motor and ultrasonic sensor, with the set of these devices performing the function of operating the screw conveyor. Programming was developed with the aim of building an interface between the user and the device, thus allowing control through the LCD Display. The use of this methodology achieved satisfactory results, enabling the creation of a functional prototype, which achieved the objective proposed at the beginning of the project.

**Key-Words:** Feeder. Arduino. Screw thread. Display. Schedule.

## LISTA DE TABELAS

Tabela 1 - Cronograma do Projeto.....	9
Tabela 2 - Tabela de orçamento .....	15



## LISTAS DE FIGURAS

Figura 1 - Galinha .....	2
Figura 2 - Gráfico negativo.....	3
Figura 3 - Alimentador 1 .....	3
Figura 4 - Pintura rupestre .....	4
Figura 5 - Criação de gado .....	5
Figura 6 - Alimentador automático de bezerras.....	6
Figura 7 - Alimentador 2 .....	8
Figura 8 - Cachorro.....	11
Figura 9 - Diagrama de blocos .....	11
Figura 10 - Servo motor .....	12
Figura 11 - Sensor ultrassônico .....	12
Figura 12 - Módulo RTC .....	13
Figura 13 - Driver.....	13
Figura 14 - Display LCD.....	13
Figura 15 - Motor de passo .....	14
Figura 16 - Arduino UNO .....	14
Figura 17- Rosca helicoidal.....	16
Figura 18 - Representação da equação .....	17
Figura 19 - Protótipo da Rosca .....	18
Figura 20 - Cano principal.....	18
Figura 21 - Figura representativa de programação .....	21
Figura 22 - Onde inserir bibliotecas .....	22
Figura 23 - Modelo de estruturação da seção void setup .....	23
Figura 24 - void setup da programação do alimentador .....	23
Figura 25 - Exemplo de código utilizado na seção void loop .....	24
Figura 26 - Void loop do alimentador .....	25
Figura 27 - Momento da montagem da ligação .....	25
Figura 28 - Esquema do módulo RTC.....	26
Figura 29 - Momento de digitação do código .....	27
Figura 30 - Momento de encaixe do motor no eixo .....	28
Figura 31 - Resultado não definitivo do protótipo .....	28
Figura 32 - Resultado definitivo do protótipo .....	31

# SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>1</b>
<b>2. DESENVOLVIMENTO .....</b>	<b>2</b>
<b>Da selvageria a domesticação .....</b>	<b>4</b>
<b>Cronograma.....</b>	<b>9</b>
<b>O Projeto .....</b>	<b>10</b>
<b>Componentes .....</b>	<b>11</b>
<b>Orçamento .....</b>	<b>14</b>
<b>Rosca Infinita (Helicoidal).....</b>	<b>15</b>
<b>Produção da Rosca.....</b>	<b>17</b>
<b>Programação.....</b>	<b>20</b>
<b>Monitoramento.....</b>	<b>25</b>
<b>3. CONCLUSÃO .....</b>	<b>32</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>33</b>

## 1. INTRODUÇÃO

Com o tempo cada vez mais escasso e com jornadas de trabalho mais intensas, o tempo para cuidar de seu animal de estimação fica limitado, tornando importante encontrar maneiras de garantir que seu pet receba a atenção e cuidados de que necessita.

Nesse projeto pensamos em soluções inovadoras e eficientes para o cuidado de animais de estimação e decidimos desenvolver um dispositivo que facilitasse a vida dos proprietários. Nesse contexto, o alimentador automático de animais emerge como uma resposta promissora, oferecendo não apenas conveniência, mas também um compromisso com a sustentabilidade por meio do uso de materiais reaproveitados. Esta pesquisa explora o projeto e as vantagens de um alimentador automático, destacando sua facilidade de uso e o custo-benefício que o tornam uma opção atrativa para aqueles que desejam garantir o bem-estar de seus animais de estimação. Este trabalho investigará o processo de criação do alimentador automático, suas funcionalidades e como ele pode melhorar a qualidade de vida dos animais e de seus cuidadores.

## 2. DESENVOLVIMENTO

O ser humano sempre foi um ser social, sua sobrevivência em tempos antigos dependia da união de um grupo em prol de abrigo, alimentação e proteção. Com a evolução, o homem desenvolveu a agricultura e a pecuária, e passou a criar animais para sua alimentação e para auxiliar seu trabalho. Os animais, por estarem fora da natureza e sob os cuidados do ser humano, tinham sua alimentação controlada pelo mesmo. Por conta disso, com todas as revoluções tecnológicas ocorridas na história, o homem passou a buscar uma forma de facilitar a alimentação das criações, desenvolvendo técnicas e dispositivos para atender esse objetivo, como os alimentadores automáticos, apresentado nesta monografia, por exemplo.

**Figura 1 - Galinha**



Os alimentadores (ou comedouros) automáticos tem, em sua maioria, um custo relativamente elevado, principalmente os que de alguma forma são automatizados, especialmente para pessoas que não vivem da criação de animais. Portanto, nosso projeto visa tornar mais acessível o uso deste dispositivo para pessoas que apenas tenham seus animais como pets, pessoas que não trabalham com a criação dos mesmos. Por conta do material e da estrutura do nosso alimentador, seu preço acaba sendo menor. Seu baixo custo, design enxuto e sistema de funcionamento de fácil compreensão acaba tornando o projeto bem acessível para qualquer tipo de pessoa que deseje alimentar seus animais (em pequena escala).

**Figura 2 - Gráfico negativo**

Atualmente, existem diversos tipos de alimentadores, tanto mecânicos quanto automáticos, para os mais variados tipos de animais, sejam eles domésticos ou não. Podem ser confeccionados de vários materiais, mas geralmente são feitos de determinados metais.

Um grande exemplo quando se fala de alimentadores é a empresa de tecnologia *Rei dos Alimentadores*, de Uberlândia, Minas Gerais,

*“Somos especialistas no desenvolvimento de Mecanismos de Dosagem de Ração e Alimentadores Automáticos para os mais variados tipos de animais, desde animais domésticos, animais de criação (sítios e fazendas) ou animais criados para fins comerciais”* (REI DOS ALIMENTADORES, 2023, acessado em 02 de junho de 2023).

A empresa possui em seu portfólio diversos modelos de alimentadores, automáticos ou não, que alimentam animais como galinhas, cavalos, porcos, cachorros, peixes, vacas, entre outros, como o modelo a seguir:

**Figura 3 - Alimentador 1**

Num geral, é possível, a partir da análise de diversas empresas e modelos de alimentadores presentes no mercado, observar um custo considerável atrelado ao

determinado modelo desejado pelo cliente. Esse valor elevado, que está presente mesmo em aplicações simples, acaba por prejudicar pessoas comuns que possuem animais e querem praticidade e facilidade na hora de alimentá-los.

## Da selvageria a domesticação

Desde os primórdios, os animais foram importantes para os seres humanos, sendo parte quase fundamental da sua alimentação e, posteriormente, ajudantes essenciais para a sobrevivência do homem em uma comunidade. Enquanto a humanidade era formada exclusivamente por nômades, vivendo da caça de presas de pequeno a médio porte e da coleta de frutas, os animais eram não somente alimento, mas também ameaça a sobrevivência. Os humanos ainda estavam começando a caminhada em direção ao topo da cadeia alimentar, posição que ocupa até os dias atuais. Grandes predadores representavam um extremo perigo à espécie.

**Figura 4 - Pintura rupestre**



Mas tudo começou a mudar com a descoberta do fogo e o início do sedentarismo. A humanidade passou a se reunir em grupos cada vez maiores, formando grandes comunidades, que com o tempo se tornaram sociedades mais complexas, onde cada cidadão passou a desempenhar funções. Os mais fortes iam buscar alimentos, enquanto mulheres, crianças, idosos e doentes ficavam esperando, já que eram menos aptos ou não estavam em condições para realizar essa tarefa. O homem começava a subir na cadeia alimentar.

Em certo momento, o lobo foi domesticado, sendo um dos primeiros a se relacionar de forma amistosa com o homem. Estes canídeos começaram a se aproximar dos assentamentos humanos, já que eles conseguiriam comida mais facilmente, se alimentando dos restos das caças humanas. Os mais dóceis estabeleceram uma relação com os homens, fornecendo proteção em troca de alimento.

Com a revolução agrícola, o ser humano passou a cultivar seu próprio alimento, criando comunidades cada vez mais populosas. Mais animais foram domados, auxiliando no crescimento e desenvolvimento da agricultura. Posteriormente, o homem passou a “cultivar” animais, com grandes criações que forneceriam itens como alimento, couro, entre outros.

### **Figura 5 - Criação de gado**



**Criação de gado, Sec. XXI**

Existiam (e existem até hoje) muitas criações de animais ao redor do mundo, de porcos, vacas, galinhas, entre outras, servindo, sobretudo, para a alimentação. Porém, os animais que não faziam parte de tal grupo, como os cachorros por exemplo, eram utilizados para outras tarefas, como a caça e a guarda de residências e recintos.

Foi somente a partir da revolução industrial que a visão da sociedade sobre os animais foi modificada. Eles não eram mais apenas fontes de alimento ou renda ou “trabalhadores”. Desse ponto em diante os animais começaram a servir como companheiros, e passaram a integrar as famílias como um membro essencial. Logo, métodos e dispositivos foram desenvolvidos para cuidar da melhor forma possível destes

novos companheiros, como camas acolchoadas, brinquedos, rações especiais, alimentadores automáticos, entre outros.

Não se tem conhecimento de quando, exatamente, o alimentador automático foi inventado e construído pela primeira vez, mas se sabe que existe desde pelo menos 1980. Um alimentador automatizado para vacas foi criado nessa época, utilizado para auxiliar os funcionários de uma fazenda na Alemanha a terem mais praticidade na alimentação dos bovinos.

Foi comprovado a otimização do trabalho na fazenda. Os proprietários relataram que a média de tempo gasto pelos funcionários para manejar bezerras num sistema convencional é de 10 minutos/bezerra/dia, enquanto para bezerras criadas através de alimentadores automáticos, se gasta menos de um minuto/bezerra/dia. Esses pesquisadores estimaram que se paga o investimento no alimentador automático em dois ou três anos considerando a economia com mão de obra. Para esse cálculo foi considerado um rebanho de 200 vacas em lactação, com taxa de descarte de 35% e taxa de mortalidade de bezerras de 10%, necessitando, portanto, de 77 bezerras/ano. O equipamento utilizado no estudo foi o da Westfalia Systemat, cujo valor de aquisição foi de US\$ 12.000,00. Tal equipamento era composto por duas estações de alimentação que suportavam 25 bezerras cada uma (50 bezerras no total) e fornecia leite ou sucedâneo e concentrado. Portanto, 77 bezerras x oito semanas (tempo de aleitamento) x sete dias/semana x 10 minutos/dia (economia de mão de obra/bezerra/dia) = 719 horas/ano x US\$7/hora (preço da mão de obra considerado) = US\$ 5.033,00 de economia com mão de obra por ano. Logo,  $US\$ 12.000,00 \div US\$ 5.033,00 = 2,3$  anos para se pagar o investimento (Alimentador automático de bezerras: revisão, 2019, p. 5).

**Figura 6 - Alimentador automático de bezerras**

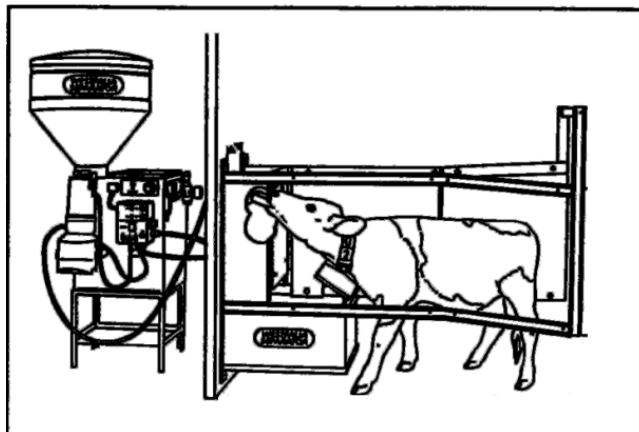


Ilustração do alimentador analisado. Fonte: Alimentador automático de bezerras: revisão, 2019, p. 4



Também citam a redução de mão de obra como uma vantagem dos alimentadores automáticos, mas alertam para a necessidade de gastar-se tempo monitorando o funcionamento do equipamento e sua higienização, além de avaliar consumo e saúde dos animais rotineiramente (Alimentador automático de bezerras: revisão, 2019, p. 5).

Os alimentadores automáticos apresentam muitas vantagens tais como: economia com mão de obra, redução do custo de alimentação, redução da idade ao desaleitamento com melhor manutenção do peso corporal, auxílio na identificação de doenças, menor taxa de mortalidade, redução da mamada cruzada, possibilidade de manter bezerras com diferentes necessidades no mesmo lote e permite que haja maior interação social entre as bezerras (Alimentador automático de bezerras: revisão, 2019, p. 7).

Este foi um experimento que provou a eficácia dos alimentadores automáticos quando aplicados em ambientes de trabalho, como fazendas. Em questão de menos de 2,5 anos o investimento se pagou, e não dura somente esse meio tempo, mostrando que um alimentador automático melhora o desempenho e lucros do proprietário.

Quando o alimentador é transferido para o âmbito doméstico, muda-se um pouco a situação, porém ainda existem vantagens quando comparado com um simples comedouro sem nenhum tipo de automatização.

Um exemplo de alimentador doméstico é o Alimentador Automático Pet Dogis Tech, que é um alimentador inteligente fabricado e comercializado pela empresa Dogis, fabricante de diversos produtos relacionados a automação inserida nos cuidados com o pet. O alimentador mencionado possui funções como câmera acoplada, programador de hora, interação por voz, conexão Wi-Fi, entre outras, como os próprios produtores citam em seu site:

- ➔ *Programador de Refeições: possui exclusiva função que programa as horas e a quantidade de porções que seu Pet precisa. Pode ser programado quantos horários e porções achar necessário durante o dia.*
- ➔ *Câmera de Vídeo: exclusiva câmera de vídeo Full HD com resolução de 720p, visão noturna, detecção de movimentos e gravação em alta resolução. Transmite ao vivo o ambiente e mostra em tempo real seu Pet.*
- ➔ *Áudio e Microfone: ele conta com microfone integrado de grande captação para ouvir o ambiente ao vivo e sistema de alto falantes que transmitem seus comandos de voz ao Pet de onde estiver.*

- ➔ *Gravador de Voz: Nas horas marcadas para refeição, poderá também gravar sua voz ou saudação para que seu Pet se familiarize ainda mais com o aparelho. Ela será reproduzida durante todo ciclo de alimentação.*
  - ➔ *Wi-Fi: conta com Wi-Fi integrado de 2,4 GHz que vai integrar seu celular com o alimentador onde você estiver.*
  - ➔ *Interação: tenha total controle e manuseie o aparelho a distância através dos sistemas Google Assistant© e Alexa© (aparelhos vendidos separadamente).*
- (DOGIS, acessado em 08 de outubro de 2023)

As funções citadas que estão presentes no modelo de alimentador apresentado são de uma utilidade considerável, porém, algumas, não são essenciais, sendo apenas um adereço que acaba por elevar o custo do produto. O preço imposto sobre o dispositivo, até o dia do acesso (08/10/2023), é de R\$839,00, valor que está um tanto além do razoável para o cidadão médio, que deseja apenas uma facilidade a mais na hora de alimentar seu pet.

**Figura 7 - Alimentador 2**



**Representação do Alimentador Automático Pet Dogis Tech.**

**Fonte: Dogis.com.br**

O projeto apresentado nesta monografia e desenvolvido pelo grupo possui certas semelhanças com o modelo da Dogis, em se tratando do hardware, como a utilização de componentes e conceitos presentes no Alimentador Dogis Tech. Um exemplo é o display LCD utilizado para o controle dos horários de liberação de comida e a programação utilizada, que possivelmente segue uma ideia parecida. Porém nosso dispositivo busca ser mais econômico e acessível para pessoas que apenas querem um auxílio na alimentação de seu pet, seja ele qual for.

Através de pesquisas de métodos para confeccionar nosso alimentador, conseguimos baratear e tornar mais acessível para as pessoas, reduzindo os custos. Utilizando o valor do Alimentador Automático Pet Dogis Tech como referência, reduzimos o custo em algo em torno de 75%, como será possível de observar no tópico de Orçamento, apresentado mais adiante.

## Cronograma

Abaixo segue um cronograma referente ao desenvolvimento do projeto, que contempla o período entre os meses de janeiro e dezembro de 2023, ano em que o planejamento e execução do projeto foram realizados.

**Tabela 1 - Cronograma do Projeto**

	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
Escolha do tema		X										
Planejamento		X	X	X					X			
Análise de métodos			X	X	X			X	X			
Análise de orçamento			X									
Coleta de dados			X									
Aquisição financeira			X	X								
Aquisição de componentes				X	X	X			X			
Divisão em equipes				X								
Montagem				X	X		X	X	X	X	X	
Análise de funcionamento					X	X		X				
Montagem da rosca					X	X	X		X			
Parte teórica				X	X	X	X	X		X	X	
Programação										X		

O tema foi escolhido durante o mês de fevereiro. Não foi uma etapa difícil, pois o grupo já tinha optado pelo projeto do comedouro automático no ano anterior. Esta fase foi somente para oficializar a ideia e fazer alguns ajustes.

Após a decisão do tema, iniciou o planejamento. Tal etapa do projeto durou praticamente o período todo, já que sempre era necessária uma nova abordagem diante de problemas que surgiam durante o processo, indo desde uma revisão dos meios e métodos até uma mudança de planos em busca de uma solução para determinados impasses.

Com o estudo e pesquisas realizados, decidimos métodos para a execução do planejamento, como formas de montagem, de arrecadar dinheiro, de comprar os componentes e materiais, entre outras.

Para conseguirmos o dinheiro necessário, realizamos uma ação para que o grupo viesse a ter uma ajuda nos custos do projeto, através de uma rifa realizada no ambiente escolar e familiar dos membros.

Os componentes foram adquiridos após a realização da rifa, utilizando o dinheiro levantado por ela, a partir de abril e durando pelos meses subsequentes.

A execução da parte prática iniciou no mês de abril, e, referindo-se a montagem do dispositivo, durou durante todo o período, sendo trabalhada todos os meses, já que o grupo se viu diante de diversos problemas, que necessitaram de uma solução, atrasando o desenvolvimento do projeto.

O grupo se dividiu em duas equipes, onde uma ficou responsável pela parte prática do projeto (que inclui montagem, análise de protótipo, revisão, execução e reexecução), e a outra se responsabilizou pela escrita desta monografia (fazendo pesquisas e descrevendo a realização do projeto).

Certos tópicos apresentados no cronograma serão abordados com mais detalhes adiante, como, por exemplo, a montagem da rosca e a programação.

## **O Projeto**

A ideia do projeto aqui apresentado surgiu a partir da necessidade de se possuir uma solução mais barata e acessível para um problema que acontece com frequência com donos de pequenos animais. Um exemplo que pode ser citado é a alimentação de animais durante viagens. Existem casos em que o animal não pode acompanhar o dono em determinadas viagens um pouco mais duradouras, portanto a alimentação do mesmo se torna um problema, precisando ficar por conta de vizinhos ou parentes, o que, por vezes, incomoda os agora responsáveis pela alimentação do bichinho.

O alimentador idealizado por nós (autores) visa facilitar essa alimentação, tirando esse incômodo que existe em quem fica responsável pelo *pet* alheio.

**Figura 8 - Cachorro**



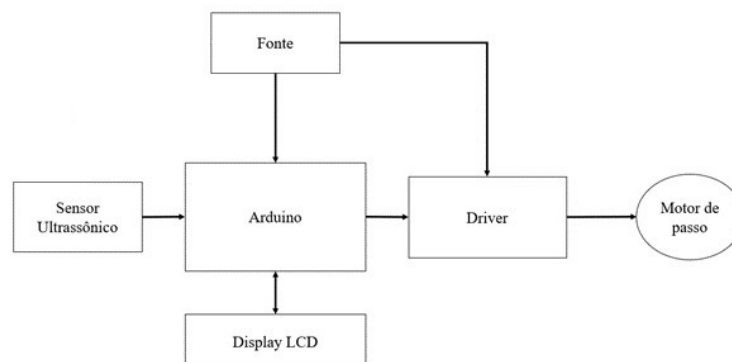
A ideia do projeto surgiu perto do final do ano anterior à escrita deste documento, e foi abraçada pelos membros do grupo. Apesar disso, durante um tempo, a ideia ficou apenas em nossas mentes, sendo colocada no papel, planejada e executada no começo deste ano.

O planejamento foi feito. Foram levantados os componentes que fariam parte do dispositivo, seus preços, sua programação e como cada um encaixaria de modo atendesse às demandas do projeto.

## Componentes

A seguir será explanado um rápido resumo dos principais componentes adquiridos e utilizados, no intuito de que o leitor esteja ciente do *hardware* que está presente no projeto.

**Figura 9 - Diagrama de blocos**



**Fonte: Próprios autores**

**Micro servo Tower Pro MG90S:** um servo motor com uma rotação de 360°, podendo ser programado para ter uma rotação precisa. Está presente no sistema da tampa do alimentador.

**Figura 10 - Servo motor**



**Sensor Ultrassônico HC-SR04:** envia sinais através do reconhecimento de determinadas distâncias. Está presente no sistema de aviso de armazenamento.

**Figura 11 - Sensor ultrassônico**



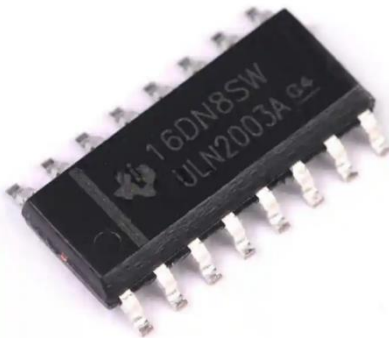
**Módulo RTC DS 1307 I2C:** temporizador programável. Presente no sistema de liberação de comida.

**Figura 12 - Módulo RTC**



**Driver Ci ULN2003A:** é responsável por executar a programação do motor de passo.

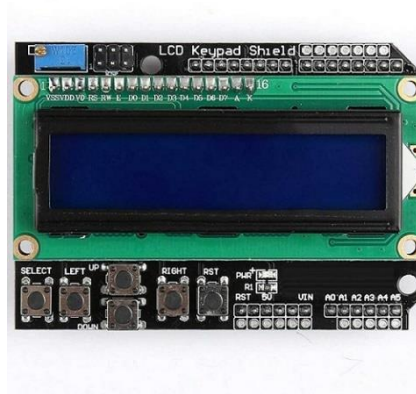
**Figura 13 - Driver**



**Rosca Infinita:** é o mecanismo que atua na liberação da comida, que gira durante o tempo programado e está presente no cano principal do dispositivo.

**Display LCD Keypad Shield:** funciona como o “menu” do projeto, sendo o componente responsável pela visualização e execução dos comandos.

**Figura 14 - Display LCD**



**Motor de passo 35BYJ46 12v:** é responsável por girar o eixo da rosca infinita, sendo assim um componente essencial no funcionamento.

**Figura 15 - Motor de passo**



**Arduino Uno R3:** microcontrolador que tem como função armazenar e executar a programação

**Figura 16 - Arduino UNO**



Os componentes apresentados acima são os essenciais que estão presentes no projeto, são os mais notáveis quando se olha o dispositivo de uma forma superficial.

Todos os itens utilizados são de extrema relevância para o funcionamento, porém aqui foram citados apenas os que implicam, de certa forma, uma função mecânica na liberação do alimento, logo, por consequência, itens como fiação elétrica, componentes de estrutura, entre outros, foram deixadas de lado

## **Orçamento**



Como tudo neste mundo, os componentes, em sua grande maioria, tiveram um custo, que foi suprido através da rifa realizada pelo grupo ou por investimento dos membros.

Abaixo segue uma tabela apresentando o custo de todos os materiais adquiridos, dos citados no tópico anterior até os não citados, incluindo doações e itens comprados para o teste do dispositivo.

**Tabela 2 - Tabela de orçamento**

<b>Item</b>	<b>Valor</b>	<b>Fornecedor</b>
Arduino UNO R3	76,5	Mercado Livre
Motor de passo 35BYJ46 12v	0	Doação
Driver Ci ULN2003A	2	Loja Eletromax
Módulo RTC DS 1307 I2C	16,49	Mercado Livre
Sensor Ultrassônico HC-SR04	15,6	Mercado Livre
Micro Servo Tower Pro MG90S	24,9	Mercado Livre
Cano PVC	7	Loja de construção
Lata de Achocolatado	0	Doação
Tábuas de madeira	0	Doação
Fios de ligação	10	Shoppe
Jumper	15	Loja Eletromax
Bateria 9V	6,49	Loja Eletromax
Super cola	10	Loja de construção
Lata de Tinta	0	Doação
Ração	0	Doação
Sachê de milho	4,99	Supermercado
Display LCD Keypad Shield	28	Shoppe

## **Rosca Infinita (Helicoidal)**

A Rosca Infinita (ou Rosca sem fim / Rosca Helicoidal) é uma parte fundamental do projeto, sendo ela o componente que atua na liberação do alimento.

**Figura 17- Rosca helicoidal**



Uma rápida definição sobre Rosca Sem Fim:

“As Roscas sem fim possuem cortes helicoidais, e são utilizadas em qualquer sistema onde se pretende precisão e torque.” (SOLUÇÕES INDUSTRIAIS, acessado em 21 de julho de 2023)

A Rosca Infinita surgiu na Grécia Antiga, em torno do ano 250 a.C, idealizada e construída por *Arquimedes de Siracusa* (por conta de seu criador, a Rosca também ficou conhecida como *parafuso de Arquimedes*) para transportar água nos sistemas de irrigação da época. Tal invenção se mostrou eficiente, conseguindo realizar o transporte de diversos materiais, desde líquidos, como água (que foi o motivo de sua criação), até sólidos, como grãos (principal produto transportado atualmente). Por conta dessa eficiência, a invenção sobreviveu ao tempo, sendo utilizada até os dias de hoje em indústrias, fazendas e até mesmo em ambientes domésticos. A ideia de Arquimedes foi estudada e aprimorada por vários cientistas através da história, originando outros dispositivos inspirados e que seguem a premissa base do *parafuso*.

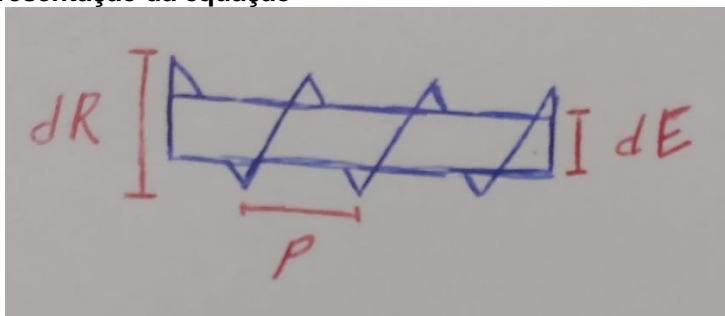
A rosca utilizada no projeto foi construída pelos membros do grupo, utilizando materiais encontrados em casa e em ambientes de ferro-velho.

Uma Rosca Helicoidal é construída a partir de cálculos que determinam suas medidas. Seu cálculo é dividido em duas etapas: O cálculo do diâmetro externo e o cálculo do diâmetro interno.

O diâmetro externo é determinado pela fórmula  $dR = dE + \frac{\sqrt{dE^2 \cdot \pi^2 + p^2}}{\pi}$ , sendo  $dR$  o diâmetro da rosca,  $dE$  o diâmetro do eixo e  $p$  o passe.

Já o diâmetro interno é calculado após o externo, pelo motivo de não ser possível calculá-lo antes. Seu resultado é obtido através da fórmula  $dEX = (dR - dE)$ , sendo  $dEX$  o diâmetro externo.

**Figura 18 - Representação da equação**



**Fonte: próprios autores**

As medidas foram alcançadas usando como base o diâmetro do eixo usado para a confecção da rosca.

## Produção da Rosca

A rosca infinita que foi utilizada para o projeto foi construída a partir de materiais reaproveitados, com o intuito de reduzir custos. Os materiais utilizados para a construção foram: um cano, utilizado na confecção do eixo, e partes de lata, que serviram na construção da helicóide.

A confecção da rosca foi realizada nas seguintes etapas: o cálculo para a posição da helicóide e comprimento necessário do eixo, em que o grupo calculou as medidas através de uma projeção em uma folha de papel, e logo após transpôs as medidas para a lata, abrindo-a pelo meio e cortando de maneira que pudesse ser modelada em um formato helicoidal, fazendo uso de uma tesoura para chapa cedida por um dos membros, e a montagem, que apresentou uma certa dificuldade em sua realização, por exigir uma precisão para que sua função no projeto fosse executada conforme o planejado; nesta etapa foi preciso a utilização de cola instantânea e cola quente. Para auxiliar na fixação da lata em torno do eixo, foram cortados vários pedaços de madeira com 4 cm de comprimento (medida referente ao tamanho do passo) e 1,5 cm de largura e posicionados para fixar o eixo durante o processo e para evitar erros na colagem, para que a lata permanecesse na posição correta, conforme suas medidas. A lata foi fixada no eixo com a cola instantânea e, após seu posicionamento, suas bordas foram seladas com a cola quente, para garantir sua fixação no cano PVC. Com o término de todo este processo, os

pedaços de madeira foram utilizados de maneira a dar suporte para a rosca, mantendo-a imóvel durante a secagem da cola, para evitar complicações.

**Figura 19 - Protótipo da Rosca**



**Fonte: autores, 2023**

A rosca infinita, já pronta, foi inserida no cano principal que compõe o projeto. Contudo, ela não encaixou da forma correta, ficando, de certo modo, com o mesmo diâmetro do cano, impossibilitando que a mesma girasse e fizesse a liberação da comida. Para resolver este problema, foi necessário que as bordas externas da lata fossem recortadas, para que pudesse girar sem dificuldades. Foi feito um acabamento na helicoide com uma lixa de aço, facilitando mais sua atuação e melhorando esteticamente.

**Figura 20 - Cano principal**



**Fonte: autores, 2023**

## **Problemas na montagem**

O grupo enfrentou certos problemas na pós-produção da rosca, sendo necessário 2 remontagens do componente.

### **A primeira rosca**

A rosca apresentada acima foi a primeira tentativa de produção de uma rosca helicoidal, porém, por problemas decorrentes da montagem, a rosca ficou enguiçada no cano principal, impossibilitando a mesma de girar conforme deveria. Mesmo com a ajustada realizada anteriormente, que fez com que aparentemente a rosca girasse da maneira certa, ela acabou ficando presa no cano principal.

Outro problema que acabou ocorrendo foi um erro na medida do tamanho da helicoide em relação ao cano principal, que foi um fator contribuinte para que a rosca não conseguisse girar. O grupo realizou um teste utilizando bolinhas de papel para simular os grãos de milho e ração (que são o objetivo final do projeto

### **A segunda rosca**

Diferente da primeira rosca o problema da segunda estava no espaçamento entre a hélice e o cano principal, ela acabou ficando presa no cano principal fazendo com que entrasse em contato contínuo com o cano, assim impossibilitando a rosca de girar e transportar as bolinhas de papel que simulavam os grãos. A solução desse problema refazer a rosca, consertando suas medidas, em especial o tamanho da helicoide com relação ao cano principal.

### **A terceira rosca**

Esta foi a tentativa de montagem da rosca que teve sucesso, já que ela pode girar da maneira adequada quando dentro do cano principal, sem entrar em contato com as paredes do cano, permitindo a liberação dos grãos (bolinhas de papel) de maneira adequada e sem mais problemas. Em aparência, ficou extremamente semelhante ao primeiro modelo, apresentado nas imagens acima, já que foi utilizado o mesmo material, sendo alterado apenas suas medidas. Portanto, este último modelo foi o utilizado para a montagem do projeto final como um todo.

## Programação

A programação é uma parte essencial do nosso projeto, sendo um pilar extremamente importante para o funcionamento do dispositivo, já que, sem a programação, toda a parte automatizada do projeto não funcionaria, fazendo com que todo o propósito fosse perdido.

A programação serve para controlar o Arduino e comandá-lo, fazendo com que realize as funções básicas do dispositivo, como controle de horário, controle de dosagem e alerta sobre o armazenamento de comida.

A linguagem utilizada para programar o Arduino é uma linguagem de programação homônima, mas baseada nas linguagens C e C++.

Primeiramente, o que é uma linguagem de programação?

*“É por onde o hardware (máquina) e o programador se comunicam. É uma linguagem formal que funciona por meio de uma série de instruções, símbolos, palavras-chave, regras semânticas e sintáticas.*

*A linguagem de programação permite que um programador crie programas a partir de um conjunto de ordens, ações consecutivas, dados e algoritmos.*

*Esse conjunto faz o controle do comportamento físico e lógico de uma máquina. Existem diversas linguagens, pois também existem diversas formas de transmitir um mesmo comando para alcançar um mesmo objetivo.*

*Por meio da linguagem de programação, é possível definir uma série de especificidades. Entre elas, quais ações o software deve executar, de acordo com cada circunstância variável, quais dados ele deve operar e como esses dados devem ser transmitidos e armazenados.”* (KENZIE, acessado em 08 de outubro de 2023)

Para o entendimento da linguagem utilizada pelo Arduino, é importante que se tenha conhecimento das linguagens C e C++, em que a linguagem usada pelo Arduino é baseada.

O que é a linguagem de programação C e C++?

*“A linguagem C é uma linguagem de programação de alto nível que foi criada nos anos 1970 para desenvolvimento de sistemas operacionais. Ela é amplamente utilizada em sistemas embarcados.”* (EMBARCADOS, acessado em 08 de outubro de 2023)

**Figura 21 - Figura representativa de programação**

```
17 string sInput;  
18 int iLength, iN;  
19 double dblTemp;  
20 bool again = true;  
21  
22 while (again) {  
23     iN = -1;  
24     again = false;  
25     getline(cin, sInput);  
26     system("cls");  
27     stringstream(sInput) >> dblTemp;  
28     iLength = sInput.length();  
29     if (iLength < 4) {  
30         again = true;  
31         continue;  
32     } else if (sInput[iLength - 3] != '.') {  
33         again = true;  
34         continue;  
35     } while (++iN < iLength) {  
36         if (isdigit(sInput[iN])) {  
37             continue;  
38         } else if (iN == (iLength - 3)) {  
39             continue;  
40         }  
41     }  
42 }
```

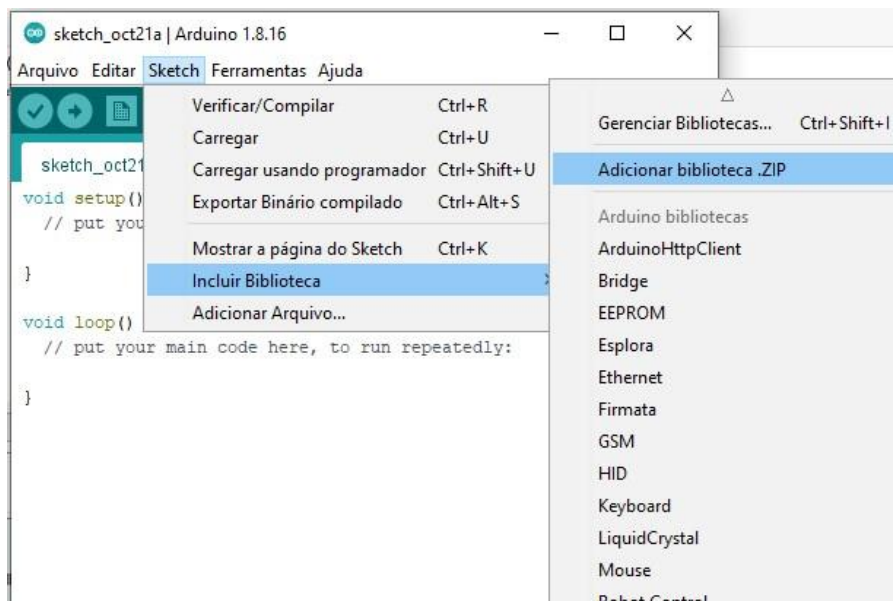
**Fonte: Idocode.com.br**

*“C++ é uma linguagem de programação open source orientada a objetos, utilizada tanto como linguagem de máquina quanto para softwares empresariais, acadêmicos, jogos, edição de imagens, fotos e vídeos.*

*Essas características permitem que ela seja considerada uma linguagem tanto de alto quanto de baixo nível, isso sendo possível pelas bibliotecas e funções que ela possui, o que permite que sistemas desenvolvidos nessa linguagem sejam de alto desempenho, estáveis e seguros.”* (Blog Trybe, acessado em 08 de outubro de 2023)

A programação que foi feita para o nosso projeto permite que as funções sejam executadas da maneira correta. Para que os periféricos respondam da maneira adequada é necessário que sejam instaladas e utilizadas bibliotecas no Arduino. Bibliotecas nada mais são que uma *“coleção de códigos que facilitam a conexão a um sensor, display, módulo, etc. Por exemplo, a biblioteca LiquidCrystal facilita a comunicação com displays LCD de caracteres.”* (ARDUINO.CC, acessado em 08 de outubro de 2023)

**Figura 22 - Onde inserir bibliotecas**



**Fonte: Portal Vida de Silício**

As bibliotecas utilizadas na programação do nosso alimentador automático foram as seguintes:

**LiquidCrystal:** *Permite a comunicação com displays alfanuméricos de cristal líquido (LCDs). Esta biblioteca permite que uma placa Arduino controle monitores LiquidCrystal (LCDs) baseados no chipset Hitachi HD44780 (ou compatível), encontrado na maioria dos LCDs baseados em texto. A biblioteca funciona no modo de 4 ou 8 bits (ou seja, usando 4 ou 8 linhas de dados além das linhas de controle rs, enable e, opcionalmente, rw).* (ARDUINO.CC, acessado em 08 de outubro de 2023)

**RTClib:** é uma biblioteca que permite que você trabalhe com módulos RTC. Esses módulos são usados para manter o controle preciso do tempo, independentemente de o Arduino estar ligado ou desligado. Eles são úteis em projetos que precisam de horários específicos, como alarmes, registros de tempo ou controle de eventos baseados em tempo. A biblioteca RTClib fornece uma maneira fácil de configurar, ler e controlar módulos RTC compatíveis com Arduino. Ela suporta uma variedade de módulos RTC, incluindo o DS1307, RTC utilizado neste projeto.

A programação do Arduino é composta por duas partes principais, que são o *void setup* e o *void loop*.



O *void setup* pode ser explicado de uma maneira rápida e simplificada como “*uma seção obrigatória de um programa para Arduino. A declaração da função Setup é realizada da seguinte forma:*”

**Figura 23 - Modelo de estruturação da seção void setup**

```
void setup()
{
  // Linhas de código do setup
}
```

**Fonte: CIRCUITAR, acessado em 17 de outubro de 2023**

*Qualquer código que estiver dentro do setup ( ), ou seja, entre chaves ( { } ) é executado uma única vez no início do seu programa.*

*Essa função é útil para configurar o Arduino, é aqui por exemplo que você coloca as configurações iniciais, como por exemplo se um LED começa desligado ou ligado, quais são os pinos de entrada e saída, entre outras coisas.”* (CIRCUITAR, acessado em 17 de outubro de 2023). Em se tratando do nosso projeto, esta seção vai receber as instruções que vão configurar os pinos do Arduino de maneira que funcione e atenda suas funções, que farão com que os componentes comandados pelo microcontrolador realizem suas designações e para que os comandos das bibliotecas sejam configurados e executados como deveriam.

**Figura 24 - void setup da programação do alimentador**

```
void setup() {

  myStepper.setSpeed(50);
  epon();
  lcd.begin(16,2);
  rtc.begin();
  //rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  lcd.setCursor(0,0);
  lcd.print("Alimentador 3000");
  pinMode(pino_rele, OUTPUT);
  digitalWrite(pino_rele, HIGH);
  delay(3000);
}
```

**Fonte: autores, 2023**

Acima estão inseridos os comandos que dão ordens ao Display LCD e ao módulo RTC, sendo eles:

- ➔ `lcd.begin`: “Inicializa a interface para a tela LCD e especifica as dimensões (largura e altura) da tela. `Begin ()` precisa ser inserido antes de qualquer outro comando da biblioteca LCD.” (ARDUINO.CC, acessado em 17 de outubro de 2023)
- ➔ `rtc.begin`: “Inicializa o RTC. `Begin ()` precisa ser inserido antes de qualquer outro método da biblioteca RTC.” (ARDUINO.CC, acessado em 17 de outubro de 2023)
- ➔ `lcd.setCursor`: “Posicione o cursor LCD; isto é, defina o local em que o texto subsequente escrito no LCD será exibido.” (ARDUINO.CC, acessado em 17 de outubro de 2023)
- ➔ `lcd.print`: “Imprime texto no LCD.” (ARDUINO.CC, acessado em 17 de outubro de 2023)
- ➔ `delay`: “Pausa o programa por uma quantidade especificada de tempo (em milissegundos). Cada segundo equivale a 1000 milissegundos.” (ARDUINO.CC, acessado em 17 de outubro de 2023)

O `void loop`, por sua vez, “faz precisamente o que o seu nome sugere, e repete-se consecutivamente enquanto a placa estiver ligada, permitindo o seu programa mudar e responder a essas mudanças. Use-a para controlar ativamente uma placa Arduino.” (ARDUINO.CC, acessado em 17 de outubro de 2023)

**Figura 25 - Exemplo de código utilizado na seção `void loop`**

```
void loop() {  
  if (digitalRead(buttonPin) == HIGH) {  
    Serial.write('H');  
  }  
  else {  
    Serial.write('L');  
  }  
  
  delay(1000);  
}
```

**Fonte: ARDUINO.CC, acessado em 17 de outubro de 2023**

**Figura 26 - Void loop do alimentador**

```
void loop() {  
  Epson();  
  KeyboardRead();  
  Keyboard();  
  Despejo1();  
  Despejo2();  
  switch(menu_num) {  
    case 1: menu1(); break;  
    case 2: menu2(); break;  
    case 3: menu3(); break;  
    case 4: menu4(); break;  
  }  
}
```

**Fonte: autores, 2023**

Essas são duas das partes indispensáveis do programa utilizado no alimentador. A programação não se resume apenas a isso, mas, por conta de ser extensa estará anexada no fim do documento de forma completa.

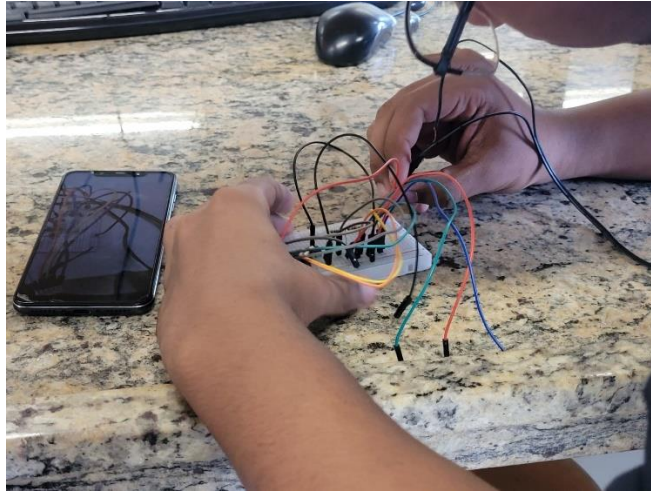
## Monitoramento

Após o término da rosca helicoidal, finalização da programação e aquisição de todos os componentes necessários, se iniciou o processo de montagem do protótipo do Alimentador Automático, realizado no dia 20 de outubro de 2023.

O grupo se reuniu no laboratório para que a montagem do projeto como um todo fosse realizada. Esta ainda não foi a montagem definitiva, porém trouxe a confirmação do funcionamento do dispositivo.

Utilizando uma *proto-board*, a ligação elétrica foi realizada, conectando o Arduino, o display LCD, o módulo RTC e o motor de passo que gira o eixo da rosca helicoidal. O primeiro passo foi a ligação dos fios na *proto-board* e no Arduino, que teve a fiação conectada nas suas entradas digitais e analógicas.

**Figura 27 - Momento da montagem da ligação**

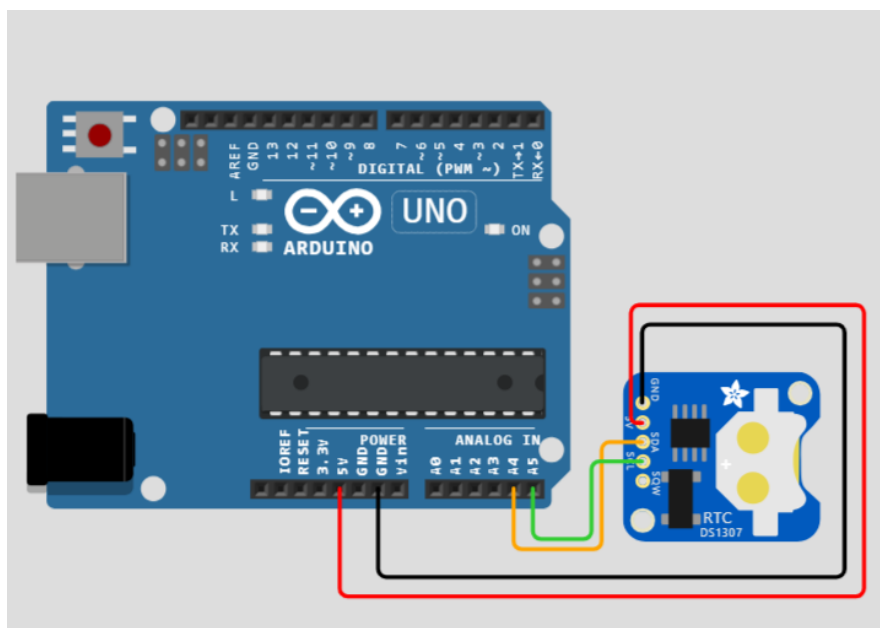


**Fonte: autores, 2023**

O Arduino foi conectado ao computador através do cabo USB que vem juntamente com o microcontrolador quando comprado, e o display foi conectado encaixando-o nas entradas do Arduino.

O módulo RTC foi ligado diretamente ao Arduino, utilizando suas entradas analógicas, além das entradas de alimentação (VCC e GND).

**Figura 28 - Esquema do módulo RTC**



**Fonte: autores, 2023**

O motor de passo foi conectado aos demais através de seus fios de alimentação, utilizando a *protoboard* como um intermediário entre o microcontrolador e o motor.

Após a realização desses processos, com o display, Arduino, fiações, módulo RTC e motor de passo conectados, era preciso fazer a parte da programação, para que o grupo pudesse testar o programa e corrigir os possíveis erros. O código foi digitado no programa do Arduino com sucesso, faltando agora o teste.

Três membros do grupo ficaram responsáveis por esta parte da execução, com o grupo por completo revisando o código após seu término.

**Figura 29 - Momento de digitação do código**



**Fonte: autores, 2023**

Com o código terminado e todas as ligações feitas, o grupo testou para verificar o funcionamento das partes do dispositivo. Foi testado apenas o motor solto, sem estar conectado a rosca. No início dos testes o motor de passo não estava girando conforme deveria, apresentando uma velocidade de rotação insignificante, ficando quase parado. Logo, após uma análise de todo o sistema, o grupo descobriu que o problema estava na linha do código que determinava a velocidade do motor; o valor digitado estava bem abaixo do valor necessário para fazer o motor girar. Durante as simulações, este mesmo valor funcionou, porém, na prática, acabou por deixar o motor muito lento, e caso ele

estivesse conectado a rosca, não teria força suficiente para girar o eixo. Foi necessária uma mudança no valor da velocidade, aumentando-a gradativamente até chegar ao número correto.

Após o motor estar girando na velocidade certa, o grupo encaixou o motor no eixo da rosca com o auxílio de parafusos e apoiou-a em uma caixa de sapato. Com tudo pronto, começou o teste “final”, para verificar o funcionamento como um todo e conferir se o motor teria força suficiente para girar o eixo da rosca helicoidal.

**Figura 30 - Momento de encaixe do motor no eixo**

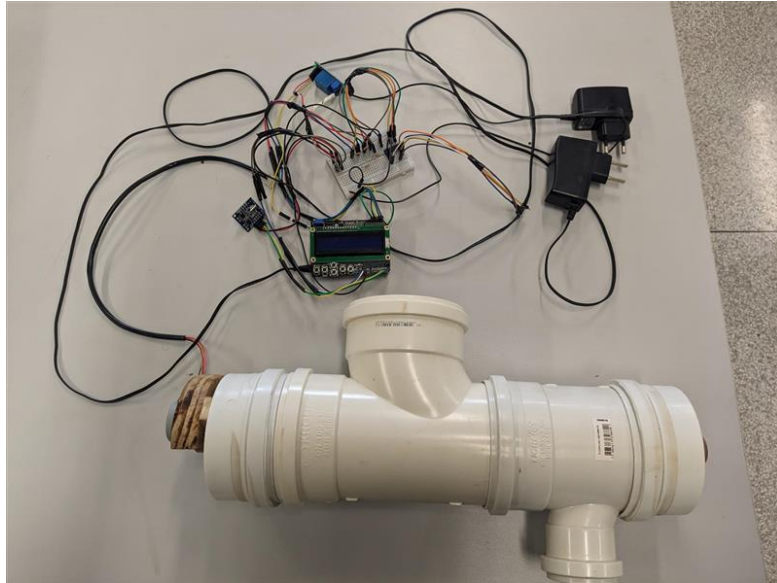


**Fonte: autores, 2023**

## **Resultado**

O protótipo se mostrou funcional, atendendo o objetivo proposto. O display realizou suas funções e conseguiu configurar a hora conforme o desejado para a liberação da comida. Todas as partes funcionaram, o motor, o Arduino, a programação e a rosca, e o protótipo foi um sucesso.

**Figura 31 - Resultado não definitivo do protótipo**

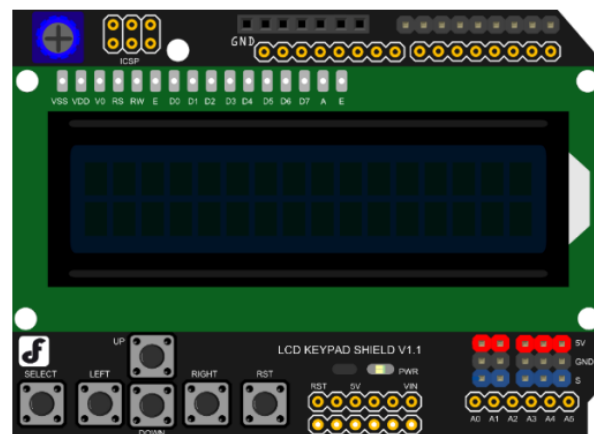


Fonte: autores, 2023

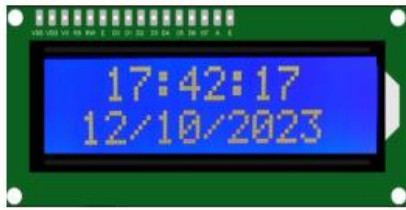
## Manual do display

Abaixo segue um manual referente a utilização do display, exemplificando as telas exibidas durante a configuração do horário de liberação da comida.

Painel de Controle



- |                                               |                                               |
|-----------------------------------------------|-----------------------------------------------|
| 1. <b>SELECT</b><br>Confirma a ação           | 4. <b>UP</b><br>Ajusta o horário para cima    |
| 2. <b>LEFT</b><br>Move para o menu à esquerda | 5. <b>DOWN</b><br>Ajusta o horário para baixo |
| 3. <b>RIGHT</b><br>Move para o menu à direita | 6. <b>RST</b><br>Reinicia o arduino           |

**Menus:**

Informa a data



Configura a hora



Define o horário do primeiro despejo



Define o horário do segundo despejo

**Imprevistos**

Após o funcionamento do protótipo, aconteceu um imprevisto. Na hora de guardar a rosca para continuar os trabalhos posteriormente, ela acabou caindo de uma altura pequena, estando dentro do cano principal, o que resultou em uma entortada na helicoide, causando problemas semelhantes aos vistos na primeira e segunda rosca. Com isso, foram necessários ajustes para que a rosca voltasse a funcionar corretamente.

**Protótipo completo**

Abaixo estão fotos do protótipo montado por completo, finalizado.



Figura 32 - Resultado definitivo do protótipo



Esta foi a estrutura final desenvolvida pelo grupo, sendo totalmente funcional e atendendo as expectativas.

### 3. CONCLUSÃO

O desenvolvimento do alimentador automático de animais resultou em um protótipo funcional que conseguiu cumprir os objetivos estabelecidos desde a fase inicial do projeto. O desempenho do protótipo foi validado mediante testes, e simulações demonstrando consistência e confiabilidade na dispensação controlada de alimentos aos animais de estimação.

A eficácia do dispositivo em prover alimentação de maneira automatizada corroborou com a premissa inicial, confirmando a capacidade do alimentador em oferecer praticidade aos cuidadores de animais. Além disso, a utilização de materiais reaproveitados não apenas se mostrou viável, mas também destacou a importância da sustentabilidade como parte integrante do projeto.

A conclusão bem-sucedida deste trabalho reforça não somente a funcionalidade do protótipo, mas também enfatiza a relevância de abordagens inovadoras e sustentáveis no contexto do cuidado com animais de estimação. O êxito do alimentador automático de animais representa não apenas uma solução prática, mas também um compromisso com a responsabilidade ambiental e a melhoria da qualidade de vida dos animais e de seus cuidadores.

Em resumo, a eficácia do protótipo reforça a realização dos objetivos estabelecidos na introdução deste estudo. Este projeto não apenas marca o término de um ciclo de desenvolvimento, mas também representa uma contribuição significativa para a comunidade de cuidadores de animais, enfatizando a importância da inovação sustentável na criação de soluções práticas e acessíveis.

## REFERÊNCIAS BIBLIOGRÁFICAS

**Arduino uno R3.** < <https://www.makehero.com/produto/placa-uno-r3-cabo-usb-para-arduino/> > Acesso em 07 jun. 2023.

### **Biblioteca de programação utilizada**

<<https://www.arduino.cc/reference/en/libraries/liquidcrystal/begin/>> Acesso em 17 out. 2023

### **Display LCD Keypad Shield.**

<https://www.diversoeletronica.com.br/loja/produto/shielddisplaylcdkeypad16x02comtecladotac>> Acesso em 05 jun. 2023.

**Dogis.** <<https://www.dogis.com.br/alimentadores>>. Acesso em 8 out. 2023.

**Driver Ci ULN2003A.** < <https://www.recicomp.com.br/produtos/uln2003a-ci-driver/>>

**Fig.4** Joseane Pereira publicado em 25/03/2019 Disponível em : <<https://aventurasnahistoria.uol.com.br/noticias/reportagem/revolucao-neolitica-foi-mesmo-tao-revolucionaria.phtml>> Acesso em 01 set. 2023.

**Fig.5** publicada em 27 jul. 2021 disponível <https://ruralcentro.com.br/noticias/pecuaria-confira-os-principais-conceitos-da-atividade-na-criacao-de-animais-85742> Acesso em 01 set. 2023

**Kenzie** <<https://kenzie.com.br/blog/linguagem-de-programacao/>> Acesso em 20 ago. 2023.

Marcos Lopes, Gustavo Silva, Luiz Marcos Simões Filho, **Alimentador automático de bezerras: Revisão** Disponível: <https://doi.org/10.31533/pubvet.v13n2a270.1-9> Acesso em 08 ago. 2023.

**Módulo RTC DS 1307 I2C.** <<https://www.piscaled.com.br/modulo-relogio-rtc-ds1307-com-bateria>> Acesso em 05 jun. 2023.

**Motor de passo 35BYJ46 12v** < <https://www.multcomercial.com.br/motor-de-passo-com-reduc-o-1-85-dc-12v-35byj46.html> > Acesso em 07 jun. 2023.

**Rei dos Alimentadores.** <<https://www.reidosalimentadores.com.br/produtos/> > Acesso em 23 jun. 2023.

**Rosca Helicoidal** < <https://www.casarinitubos.com.br/roscas-transportadoras-helicoidais> > Acesso em 20 jun. 2023

**Sensor Ultrassônico HC-SR04.** <<https://www.megaeshop.com.br/arduino/sensor-ultrassonico-de-distancia-hc-sr04-modulo-arduino> > Acesso em 05 jun. 2023.

**Servo Motor.** < <https://curtocircuito.com.br/servo-motor-mg90s-towerpro.html> > Acesso em 05 jun. 2023.

**Solução Industrial** <<https://www.solucoesindustriais.com.br/rosca-sem-fim> > Acesso em 20 jun. 2023.

## **Anexos**

[https://github.com/LuwhisGod/programa-o-do-arduino/blob/main/meu\\_ardunas21.ino](https://github.com/LuwhisGod/programa-o-do-arduino/blob/main/meu_ardunas21.ino)