



UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE FÍSICA E ESCOLA DE ENGENHARIA

Desenvolvimento e prototipagem de um sistema de alimentação automática para pets

Laura Martinelli Braunstein

Orientador: Prof. Rogério Luis Maltez

Porto Alegre, 19 de Agosto de 2024.

SUMÁRIO

1. Identificação do Projeto.....	2
1.1 Declaração do Objetivo do Projeto.....	2
1.2 Público-alvo.....	2
1.3 Outras Partes Interessadas e Impactos.....	3
1.4 Lista de Necessidades do Usuário.....	3
2. Desenvolvimento de Especificações.....	4
2.1 Descrição do Contexto de Uso.....	4
2.2 Avaliação Comparativa.....	5
2.3 Lista de Especificações.....	8
3. Projeto Conceitual.....	9
4. Projeto Detalhado.....	9
4.1 Projeto Mecânico.....	9
4.1.1 Prototipagem do sistema dispensador de ração.....	10
4.1.2 Estrutura.....	10
4.2 Projeto Eletrônico.....	13
4.2.1 Hardware.....	13
4.2.1.1 Motor de Passo de 5 V e Módulo de Controle (Driver ULN2003).....	13
4.2.1.2 Protótipo de sistema com sensor de peso com amplificação.....	15
4.2.1.2.1 Célula de carga de 5 kg.....	15
4.2.1.2.2 Módulo de Amplificação HX711.....	17
4.2.1.3 Interface com o usuário (a).....	19
4.2.1.4 Fonte DC.....	21
4.2.1.5 Montagem do circuito.....	21
4.2.1.6 Componentes Eletrônicos.....	23
4.2.2 Interface com o (a) usuário (a).....	24
4.2.2.1 Navegação.....	25
4.2.2.2 Configurações.....	27
4.2.2.3 Funcionamento do Timer.....	28
4.2.3 Software.....	29
5. Implementação.....	31
5.1 Montagem da estrutura final.....	31
5.2 Verificação.....	32
5.3 Validação.....	32
6. Conclusão.....	32
Apêndices.....	32
Apêndice A: Código principal.....	32
Apêndice B: Manual do (a) usuário (a).....	37

1. Identificação do Projeto

1.1 Declaração do Objetivo do Projeto

Este projeto é feito com o objetivo de concluir o curso de bacharelado em Engenharia Física na UFRGS através de um projeto que integra hardware e software, seguindo o modelo EPICS¹ para projetos em engenharia. O mercado de produtos e serviços relacionados a animais de estimação tem apresentado um crescimento significativo nos últimos anos. Os proprietários de pets estão cada vez mais dispostos a investir em soluções inovadoras que proporcionem conforto, saúde e bem-estar para seus animais de estimação. Nesse contexto, um projeto de alimentação automática de pets surge como uma oportunidade comercial promissora.

O Brasil, de acordo com a Associação Brasileira da Indústria de Produtos para animais de Estimação², é reconhecido como o segundo país com a maior população de cães, gatos e aves canoras e ornamentais em todo o mundo, e ocupa o terceiro lugar em termos de população total de animais de estimação. Atualmente, o país conta com cerca de 54,2 milhões de cães, 23,9 milhões de gatos, 19,1 milhões de peixes, 39,8 milhões de aves e mais 2,3 milhões de outros animais, totalizando aproximadamente 139,3 milhões de pets. Esses números evidenciam a força e o potencial do setor pet na economia brasileira.

O setor pet é composto por diversas indústrias e participantes da cadeia de distribuição, incluindo alimentos para animais de estimação (Pet Food), medicamentos veterinários (Pet Vet) e cuidados de saúde e higiene para pets (Pet Care). Atualmente, o mercado pet já representa 0,36% do Produto Interno Bruto (PIB) brasileiro. Em 2018, a indústria de produtos para animais de estimação registrou um faturamento de R\$ 20,3 bilhões, um aumento significativo em relação aos R\$ 3,3 bilhões de 2006.

O interesse no mercado de pets é impulsionado pelo fato de que os proprietários estão dispostos a investir em produtos de alta qualidade e soluções convenientes para cuidar de seus animais de estimação. O segmento alimentação de pets é o que mais movimenta esse mercado, nesse âmbito é feito o projeto de um produto para alimentação automática de pets. Os donos de pets muitas vezes têm horários ocupados e nem sempre conseguem garantir que seus animais de estimação sejam alimentados adequadamente ao longo do dia. Uma solução automatizada que ofereça comodidade e segurança pode se tornar extremamente atraente para esse público. Dentro desse espírito, apresentamos esse projeto visando o desenvolvimento de um produto para alimentação automática de pets, como cães e gatos.

1.2 Público-alvo

O projeto de alimentação automática para pets tem como público-alvo proprietários ocupados que enfrentam dificuldades em alimentar seus animais de estimação devido à falta de tempo ou ausências frequentes. Além disso, atrai pessoas que desejam estabelecer uma rotina consistente para seus pets, promovendo saúde e equilíbrio nutricional. Também é direcionado a proprietários que valorizam a praticidade e conveniência de um sistema automatizado, eliminando a necessidade de ajustar horários manualmente.

¹ Modelo de relatório retirado de <https://engineering.purdue.edu/EPICS/purdue/current-students/team-documents/design-documents>

² Informações retiradas da ABINPET, Associação Brasileira da Indústria de Produtos para animais de Estimação (site: <https://abinpet.org.br/>)

1.3 Outras Partes Interessadas e Impactos

Além dos clientes diretos, que são os proprietários de animais de estimação em busca de um sistema de alimentação automática, outros grupos também podem ser afetados pelo projeto. Isso inclui:

1. Pontos de comercialização de produtos para pets.
2. Veterinários e profissionais de cuidados com animais: Os veterinários e outros profissionais de cuidados com animais têm um interesse direto no sucesso do projeto, pois ele pode contribuir para a saúde e o bem-estar dos animais. Eles podem recomendar esse tipo de sistema aos proprietários como uma forma de garantir a alimentação adequada de seus pets, especialmente em casos de necessidades dietéticas específicas ou condições de saúde.
3. Provedores de serviços de cuidados para animais: Empresas ou profissionais que oferecem serviços de cuidados para animais, como creches, hotéis, pet sitters e dog walkers, também podem ser impactados pelo projeto. Um sistema automatizado de alimentação e hidratação pode complementar esses serviços e facilitar a rotina de cuidados com os animais sob sua responsabilidade.
4. Profissionais de manutenção e assistência técnica: Os profissionais responsáveis pela manutenção e assistência técnica dos sistemas podem se beneficiar do projeto, já que a implementação de um sistema automatizado pode exigir serviços de instalação, reparo e suporte técnico ao longo do tempo.
5. Empresas de design para o desenho final do equipamento.
6. Indústria de plástico para produção das partes e do casco do equipamento comercial.
7. Parceiros de TI para a produção de um site.
8. Empresas de pagamento online.

1.4 Lista de Necessidades do Usuário

Os donos dos pets são parte interessada fundamental, pois desejam proporcionar o melhor cuidado para seus amados animais de estimação. Através desta lista, compreendemos suas necessidades específicas, como praticidade, conveniência, confiabilidade e facilidade de uso. Eles desejam um sistema que automatize a alimentação de seus cães e gatos, garantindo a comida adequada de seus pets mesmo quando não estão em casa.

Necessidade	Usuário	Necessidade do Usuário
1	Pet.	Deve ter porções de comida adequadas ao seu porte.
2	Pet.	Deve ter uma quantidade limitada de porções ao dia.
3	Pet.	Não deve acumular comida no pote ao longo do dia.
4	Tutor do pet.	Ter um reservatório suficientemente grande para não ter que fazer uma reposição diária de ração.

Necessidade	Usuário	Necessidade do Usuário
5	Tutor do pet.	Que o pote e reservatório de comida seja removível e fácil de lavar.
6	Tutor do pet.	Poder controlar a quantidade de comida e o tempo de funcionamento do produto.

Tabela 1 - Lista de Necessidades do Usuário.

2. Desenvolvimento de Especificações

2.1 Descrição do Contexto de Uso

Este projeto visa permitir que tutores forneçam comida fresca e limpa automaticamente para cães e gatos. Eles poderão ajustar a quantidade de ração por porção e definir os intervalos de tempo para dispensação, garantindo uma alimentação controlada e adequada.

- **Limitações de Espaço:** O projeto será compacto, com maior altura do que largura, e uma caixa de armazenamento de comida em cima para otimizar o espaço. Deve-se considerar o espaço necessário para armazenar o sistema e a comida, garantindo facilidade de armazenamento e ocupando o mínimo de espaço possível. Materiais seguros e vedação adequada são essenciais para manter a higiene dos alimentos. Considerações sobre uma superfície estável e acessibilidade para limpeza e manutenção também são importantes.
- **Manutenção:** A manutenção do equipamento pode ser realizada pelos próprios donos, com orientações do manual do sistema. Pode ser necessário suporte ao cliente para problemas técnicos ou dúvidas adicionais.
- **Condições Ambientais:** O projeto é destinado a uso interno ou externo, mas deve ser protegido contra chuva ou sol intenso. Partes eletrônicas não terão contato com umidade e água; os recipientes de comida e armazenamento serão destacáveis para limpeza. Sensores estarão protegidos contra acesso dos animais.
- **Segurança e Durabilidade:** O sistema deve ser bem projetado para evitar lesões ou danos aos animais, usando materiais seguros e duráveis. Deve resistir ao uso diário e às condições ambientais normais, minimizando a necessidade de substituições ou reparos.
- **Fatores Sociais e Comportamentais:** Considerações sobre preferências alimentares específicas de cães e gatos, bem como necessidades individuais, são importantes. Adaptações podem ser feitas para diferentes tamanhos de animais, dietas específicas ou restrições alimentares.
- **Limitações Tecnológicas:** As limitações podem incluir a precisão e confiabilidade do sensor de peso.

2.2 Avaliação Comparativa

No mercado atual, existem soluções disponíveis para alimentar os pets na ausência dos donos. Uma opção simples e acessível é o "Comedouro Duplo Automático para Cachorros e Bebedouro para Gatos" (Fig. 1),

facilmente encontrado em lojas online. Esse produto é projetado com praticidade em mente.

O funcionamento desse comedouro, visto na figura 1, é descomplicado: o reservatório de água e comida fica posicionado acima dos potes de alimentação. Há uma pequena fenda em que a comida passa a medida que o animal consome para a queda de mais comida ao longo do tempo. Esse processo é controlado pela gravidade e estimula o pet a buscar sua alimentação quando sentir necessidade.

Há pontos positivos e negativos a serem considerados. O "Comedouro Duplo Automático" possui um preço acessível (R\$84,99), sendo uma opção econômica para os donos de pets. Além disso, não requer eletricidade para funcionar, o que é conveniente. No entanto, existem alguns pontos negativos a serem observados. A ração pode emperrar na fenda, o que pode exigir que o dono agite o equipamento para destravar. Além disso, a quantidade de comida não é regulada, o que pode levar o animal a comer continuamente até que a ração se esgote.



Figura 1 - Comedouro Duplo Automático³

É possível encontrar no mercado outros produtos com uma proposta similar, porém com preços mais elevados. Um exemplo é o "Alimentador Automático 2 em 1 Água e Ração Sortido - Meu Pet", disponível por 211,22 reais na loja Magazineluiza, visto na figura 2:

³ Fonte: divulgação do produto na internet.

https://www.amazon.com.br/Comedouro-Duplo-Automatico-Cachorros-Bebedouro/dp/B08WJG21DP/ref=asc_df_B08WJG21DP/?tag=googleshopp00-20&linkCode=df0&hvadid=405257279186&hvpos=&hvnetw=g&hvrnd=1183724149118295012&hvnone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmld=&hvlocint=&hvlocphy=1001686&hvtargetid=pla-1591448473910&pssc=1



Figura 2 - Alimentador Automático 2 em 1 Água e Ração Sortido - Meu Pet⁴

Uma solução inovadora, embora com um ponto de atenção em relação à reposição, é o "Alimentador Automático para Cachorros e Gatos - Pet Feeder", visto na figura 3. Esse produto, disponível por 299,90 reais, oferece uma abordagem única, permitindo que o dono programe a abertura da portinha em horários pré-definidos para acessar porções de comida manualmente colocadas no dispositivo.

O "Pet Feeder" apresenta seis grades diferentes, nas quais é possível colocar a comida em porções separadas. Além disso, o sistema permite que o usuário defina até seis horários diferentes para alimentar o animal. Essa funcionalidade programável é especialmente útil para garantir a alimentação regular e controlada, mesmo quando o dono não está presente.

No entanto, é importante considerar que a reposição manual das porções de comida é necessária periodicamente para manter o funcionamento contínuo do dispositivo. Portanto, esse fator deve ser levado em conta ao avaliar a conveniência e a praticidade desse produto.



Figura 3 - Alimentador Automático para Cachorros e Gatos - Pet Feeder⁵

Existem soluções no mercado com preços mais elevados, porém, que oferecem resultados mais interessantes. Um exemplo é o "Alimentador Automático Cães Gatos Pets Programável Animus", visto na figura 4, disponível na Amazon por 379 reais na seção de Comedouros, Bebedouros e Acessórios de Pet Shop. Esse

⁴ Fonte: divulgação do produto na internet.

https://www.magazineluiza.com.br/alimentador-automatgico-2-em-1-agua-e-racao-sortido-meu-pet/p/fdhjf7a2ck/pe/ouan/?seller_id=lojaestiloconforto2&utm_source=google&utm_medium=pla&utm_campaign=&partner_id=70403&gclid=CjwKCAjw5remBhBiEiwAxL2M97iM4p2ouRZ6oJKSyXje4tcAH_00ysR8ePyDw49xyEQ0tIP_whVGdRoCj4gQAvD_BwE

⁵ Fonte: divulgação do produto na internet.

https://petmagia.com.br/products/alimentador-automatgico-para-cachorros-e-gatos-pet-feeder?variant=43795186483429¤cy=BRL&utm_medium=product_sync&utm_source=google&utm_content=sag_organic&utm_campaign=sag_organic&utm_source=google&utm_medium=roas&utm_campaign=pmax-cachorros&utm_content=17795489228&gad=1&gclid=CjwKCAjw5remBhBiEiwAxL2M97iM4p2ouRZ6oJKSyXje4tcAH_00ysR8ePyDw49xyEQ0tIP_whVGdRoCj4gQAvD_BwE

produto se aproxima do funcionamento desejado para o projeto, permitindo escolher entre diferentes tipos pré-definidos de porções de ração e intervalos de tempo para alimentação. Contudo, o valor desse produto é considerado alto para o mercado, tornando-o menos acessível. Além disso, ele não verifica se o pote se encontra vazio antes de repor uma nova porção. Essa seria uma função a ser empregada nesse projeto.



Figura 4 - Alimentador Automático Cães Gatos Pets Programável - Animus⁶

Existem outras opções no mercado que seguem a mesma lógica de automação, porém, apresentam um preço mais elevado. Um exemplo é o "Mufern Alimentadores automáticos para cães, visto na figura 5, com comedouros cronometrados para gatos, controle de tamanho de porção programável de 4 L para 1 a 6 refeições por dia", disponível na Amazon por aproximadamente 429 reais. Esse produto oferece recursos avançados, como a capacidade de programar o tamanho da porção de comida e definir de 1 a 6 refeições por dia. Com um reservatório de 4 litros, ele fornece uma opção conveniente para garantir a alimentação regular e controlada do pet.



Figura 5 - Alimentadores automáticos para cães - Mufern⁷

Após análise das opções disponíveis no mercado, podemos concluir que existe espaço para uma solução tecnológica programável de alimentação de pets por um preço acessível.

Dentre todas as opções existentes, a mais atrativa para o público em geral acaba sendo a primeira opção mencionada, principalmente devido ao seu preço mais baixo e à capacidade de oferecer água e comida ao pet

⁶ Fonte: divulgação do produto na internet. <https://www.amazon.com.br/Alimentador-Autom%C3%A1tico-Gatos-Program%C3%A1vel-Animus/dp/B09L79FFGV>

⁷Fonte: divulgação do produto na internet.

<https://www.amazon.com.br/Mufern-Alimentador-Autom%C3%A1tico-Animais-Estima%C3%A7%C3%A3o/dp/B0C7YY1P5X>

em um único dispositivo. No entanto, essa solução ainda apresenta limitações em termos de programabilidade em tempo e quantidade.

Diante dessa análise, a proposta deste projeto é desenvolver uma solução que permita a programação da alimentação dos pets em termos de horários e quantidades e que seja acessível. O objetivo é oferecer uma solução mais acessível do que as atualmente disponíveis no mercado, proporcionando aos donos de pets uma opção conveniente e bastante completa para a alimentação de seus animais de estimação.

2.3 Lista de Especificações.

Para cada necessidade do usuário, serão listadas as especificações necessárias para atender a esse requisito.

Necessidade #	Necessidade do Usuário	Especificação
1	Deve ter porções de comida adequadas ao seu porte.	Permitir a seleção de porções de ração entre 0 g e 400 g, ajustável em incrementos de 1g.
2	Deve ter uma quantidade limitada de porções ao dia.	Permitir a configuração de até 24 porções de ração por dia, com intervalos ajustáveis entre 1 e 24 horas.
3	Não deve acumular comida no pote ao longo do dia.	Detectar quando a ração no pote excede 20 g e interromper a liberação de novas porções até que o peso da ração seja inferior a 20 g.
4	Ter um reservatório suficientemente grande para não precisar de reposições diárias de ração.	Um reservatório com capacidade para abrigar 1 kg de ração.
5	Que o pote e reservatório de comida seja removível e fácil de lavar.	Pote e reservatório removíveis com sistema de encaixe simples, permitindo fácil retirada para limpeza.
6	Ter um produto com display mostrando a quantidade de comida e de tempo configurados para controle.	Display digital que exibe a quantidade de comida (em gramas) e o tempo configurado (em horas), com capacidade para ajustar esses valores usando uma interface de três botões.

Tabela 2 - Especificações das Necessidades do Usuário

3. Projeto Conceitual

Nessa fase do projeto, após falar com pessoas interessadas no produto para entender exatamente quais são as suas necessidades, e depois de criar uma lista de especificações, estamos prontos para começar a trazer a criatividade da montagem do produto à tona. É nesse momento em que serão exploradas ideias e soluções ao problema levantado, ou mini-soluções para solucionar partes do problema - o que é chamado de diversão. Para que mais tarde as ideias sejam escolhidas com cuidado e integradas para criar a solução final - ponto chamado de conversão.

O método escolhido para obtenção de ideias é o *Brainstorming*. Essa técnica é amplamente utilizada para promover a geração de ideias de forma criativa e colaborativa. Nessa abordagem, um grupo de indivíduos se reúne e, sob a orientação de um facilitador, compartilha livremente todas as ideias que surgem sobre um determinado tópico ou problema. Não há críticas ou avaliações nessa etapa, e o objetivo é incentivar a produção de um grande volume de ideias, mesmo que aparentemente sem conexão. Esse método é conhecido por estimular a participação ativa de todos os membros, favorecendo a diversidade de perspectivas e, assim, aumentando as chances de encontrar soluções inovadoras. A eficácia do *Brainstorming* tem sido amplamente comprovada em diversas áreas e é citada em estudos acadêmicos e pesquisas como uma abordagem valiosa para estimular a criatividade e o pensamento divergente⁸.

Nesse caso específico o *Brainstorming* foi adaptado para trazer ideias de um indivíduo, o qual escreve esse trabalho, a partir dos estudos previamente descritos. As ideias serão listadas abaixo:

- Reservatório de comida: Ele deve ser adequado ao tipo de alimento e ter capacidade suficiente para suprir as necessidades do animal por um determinado período.
- Sensores de peso: Esses sensores seriam conectados ao sistema de controle para agir em conjunto com o sistema de acionamento da liberação de comida na quantidade e tempos desejados.
- Sistema de controle: Um microcontrolador ou uma placa de desenvolvimento poderiam ser utilizados como o sistema de controle central do projeto. Esse sistema de controle receberia os dados dos sensores de peso e controlaria os mecanismos de liberação de comida. Ele também poderia incluir recursos como um temporizador programável para definir os horários de alimentação.
- Mecanismos de liberação: Seriam dispositivos acionados pelo sistema de controle para liberar a quantidade apropriada de comida. Por exemplo, um alimentador poderia ser ativado para liberar uma porção específica de comida. Esses mecanismos seriam conectados ao sistema de controle para receber os comandos de liberação.
- Fonte de energia: O sistema de alimentação automática precisaria de uma fonte de energia para funcionar. Poderia ser utilizada uma fonte de energia elétrica, como uma tomada.

⁸ Osborn, A. F. *Applied Imagination: Principles and Procedures of Creative Thinking*; Paulus, P. B. *Group Creativity: Innovation through Collaboration*; Isaksen, S. G. et al. *Creative Approaches to Problem Solving: A Framework for Innovation and Change*

4. Projeto Detalhado

Nesta seção, será apresentado o Projeto Detalhado do sistema dispensador de ração, abrangendo tanto os aspectos mecânicos quanto eletrônicos. Inicialmente, o Projeto Mecânico será explorado, incluindo a prototipagem do sistema dispensador e a estrutura física do equipamento. Em seguida, o Projeto Eletrônico será abordado em profundidade, detalhando o hardware utilizado, como o motor de passo e o sensor de peso, além de apresentar a interface com o usuário, fontes de alimentação e a montagem do circuito. A seção também incluirá uma visão geral das funcionalidades do software desenvolvido para controlar o sistema. Cada subseção fornecerá uma análise detalhada dos componentes e processos envolvidos.

4.1 Projeto Mecânico

O Projeto Mecânico foca nos elementos estruturais e na concepção física da máquina. Nesta seção, serão descritos os componentes que compõem a estrutura mecânica do dispositivo, incluindo a integração dos mecanismos responsáveis pelo armazenamento e liberação da ração.

4.1.1 Prototipagem do sistema dispensador de ração

Nesse primeiro protótipo, a ideia era validar como seria possível ter uma passagem de liberação de comida que abrisse e fechasse de forma programável para que a ração saísse no horário e quantidade desejados.

A ração é armazenada em uma garrafa plástica moldada para obter formato quadrado que se encaixa em uma tampa. O formato da tampa foi feito sob medida em software, cujas peças foram posteriormente cortadas em uma impressora a laser, conforme pode ser visto na figura 6.



Figura 6 - Vista frontal do protótipo montado.

4.1.2 Estrutura

Para a realização da estrutura do projeto, dividirei em duas partes: O sistema de liberação de ração com o reservatório de ração (parte superior da máquina) e o sistema de pesagem com o pote de ração (parte inferior da máquina).

A começar pela parte superior do projeto, foram utilizados os seguintes componentes:

N.º	Item	Quantidade	Proveniência	Material
1	Garrafa plástica moldada para obter formato quadrado.	1	Comprada em super mercado.	Plástico.
2	Caixa que envolve a tampa da garrafa e faz o encaixe com o restante da estrutura..	2	Projetado em software CorelDraw e cortado em uma cortadora a laser.	Acrílico.
3	Caixa que envolve as engrenagens e os tubos	1	Projetado em software CorelDraw	MDF.

N.º	Item	Quantidade	Proveniência	Material
	pelos quais a ração passa.		e cortado em uma cortadora a laser.	
4	Engrenagens.	4	Projetado em software CorelDraw e cortado em uma cortadora a laser.	Acrílico.
5	Espaçadores.	9	Projetado em software CorelDraw e cortado em uma cortadora a laser.	MDF.
6	Canos que guiam os tubos de borracha.	2	Adquiridos em um ferro velho.	Ferro.
7	Tubos de borracha.	2	Adquiridos em um ferro velho.	Borracha.

Tabela 3 - Materiais utilizados pelo sistema de liberação de ração com o reservatório de ração.

Nas figuras 7 a 9, podem ser vistas as dimensões das peças exclusivamente desenvolvidas para o projeto e enumeradas de acordo com a tabela 3.

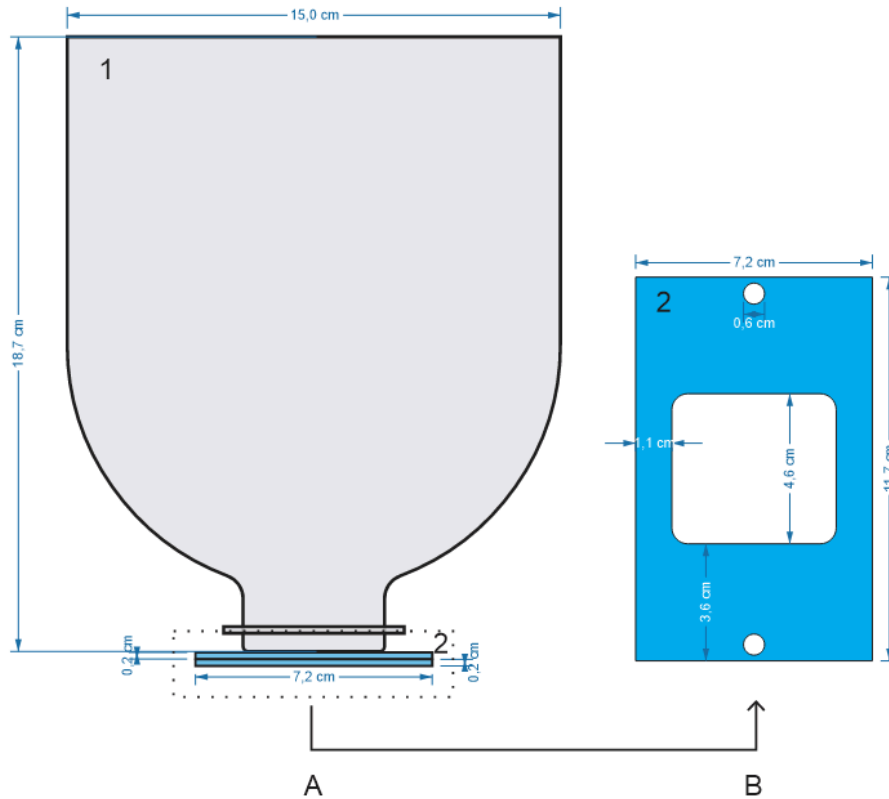


Figura 7 - A) Armazenador de ração feito a partir de garrafa de água mineral de 5 L, com base de acrílico de 2 mm de espessura. B) Tampa feita sob medida para o armazenador de ração usando chapas cortadas de acrílico de 2 mm de espessura.

A estrutura que liga a parte superior à parte inferior é composta de duas chapas de mdf de 3 mm de espessura e de uma rampa curva pela qual a ração cai no pote (feita de plástico), conforme visto na figura 8.

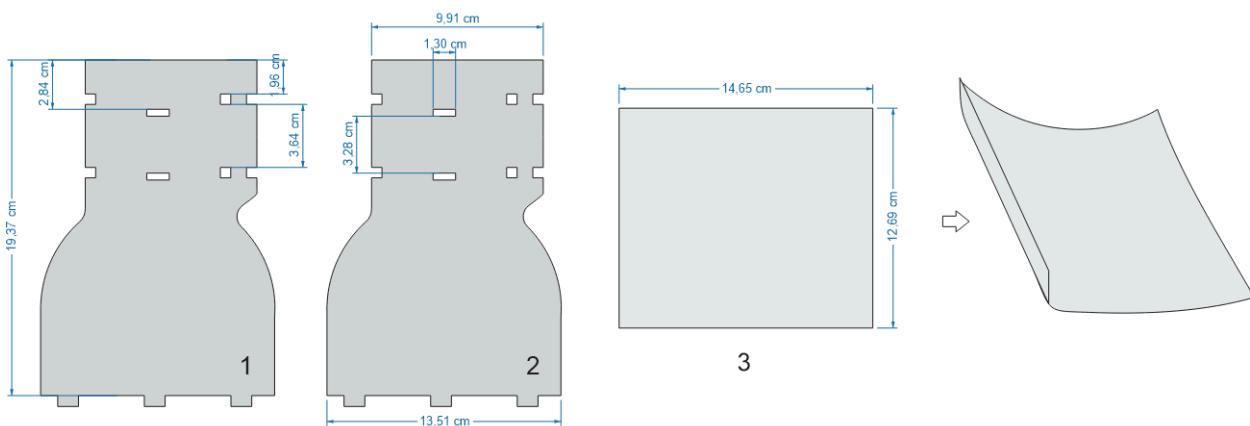


Figura 8 - Peças desenvolvidas para a integração da parte superior do projeto com a base. 1 e 2: as laterais que se unem à caixa de engrenagens superior com a base encaixável na caixa inferior onde terá o pote e o circuito. 3: a rampa pela qual a ração desce no pote.

Na base do projeto, foi cortada uma caixa em mdf usando seis chapas de 3 mm de espessura, conforme visto na figura 9.

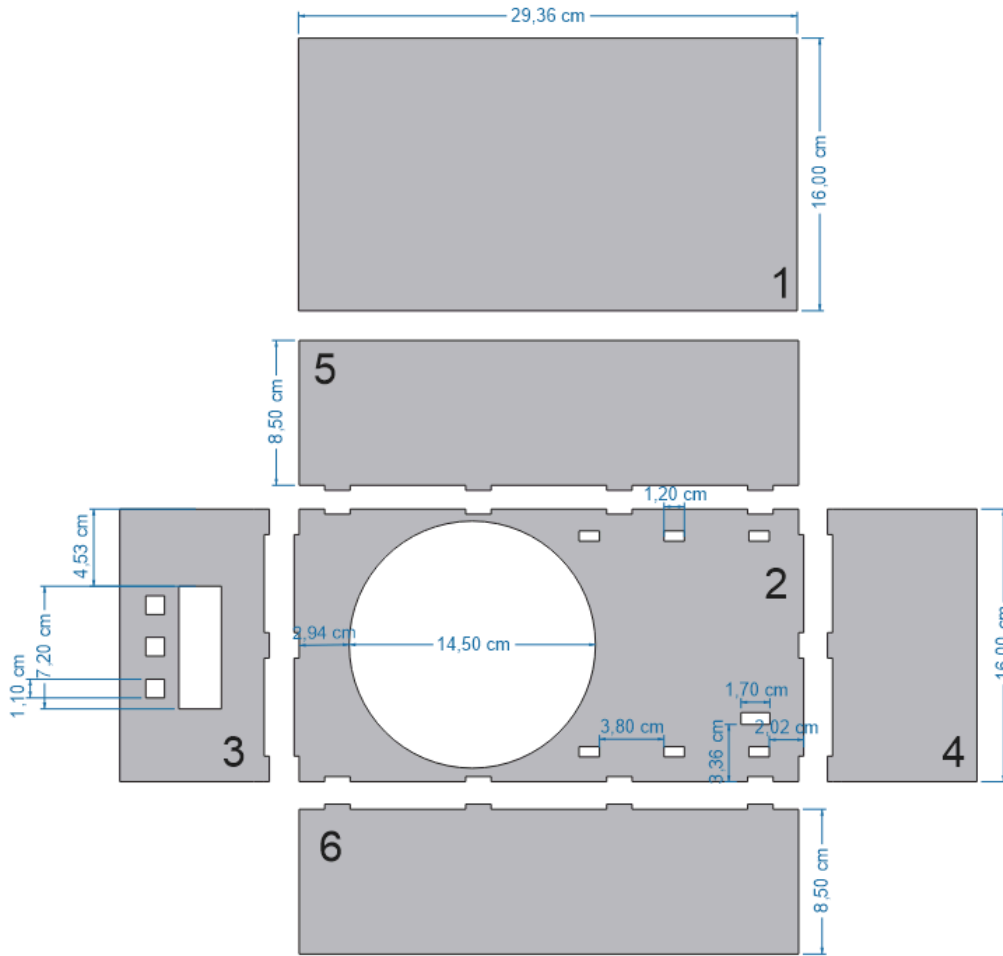


Figura 9 - 1: Base da caixa. 2: Parte superior da caixa, com furo para o pote de ração, 6 furos idênticos para conectar a caixa da estrutura superior e um furo adicional para passagem de fios para dentro da caixa. 3: Superfície frontal da caixa que contará com o LED e 3 botões. 4: Superfície posterior da caixa. 5 e 6: Laterais direita e esquerda da caixa.

A estrutura que armazena o sistema de engrenagens pelo qual a ração será liberada foi cortada com chapas de MDF e acrílico, conforme visto na figura 10.

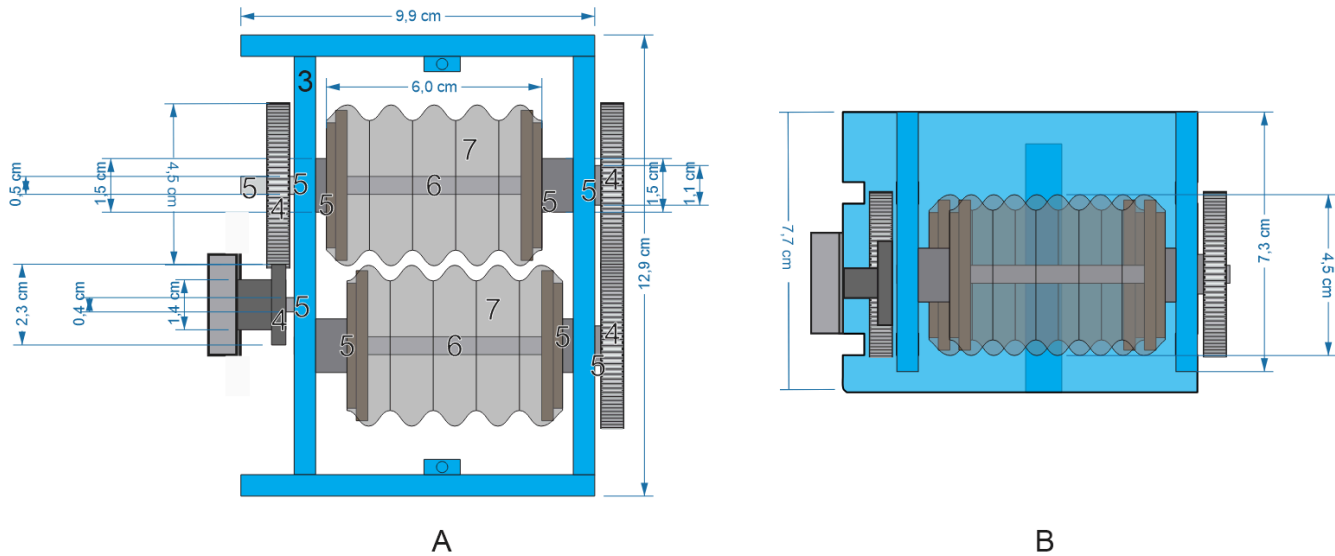


Figura 10 - Sistema de engrenagens pelo qual a ração será liberada. A) Vista Superior: motor de passo encontra-se acoplado à esquerda, na parte externa da estrutura, responsável por acionar as engrenagens responsáveis por girar um eixo axial de um tubo flexível em sentido horário e o outro anti-horário, forçando a passagem de grãos de ração pelo espaço entre eles. B) Vista Lateral.

4.2 Projeto Eletrônico

O Projeto Eletrônico abordará os componentes e circuitos responsáveis por controlar e operar a máquina. Nesta seção, serão detalhados o hardware utilizado, incluindo o motor de passo, sensores, módulos de controle, e a interface com o (a) usuário (a). Além disso, será descrita a montagem do circuito e a integração dos componentes eletrônicos.

4.2.1 Hardware

Nesta seção serão detalhados os componentes eletrônicos cruciais para o funcionamento da máquina que opera de acordo com a quantidade e o tempo configurados pelo usuário. Esta seção abordará o motor de passo e o módulo de controle, o sistema de pesagem com a célula de carga e o módulo amplificador HX711, além da interface com o usuário composta por um display LCD e botões de controle.

4.2.1.1 Motor de Passo de 5 V e Módulo de Controle (Driver ULN2003)

Para a realização dessa etapa, foi feito um mecanismo de engrenagens controladas por um motor de passo de quatro bobinas alimentados por 5 V e por um módulo de controle (driver ULN2003), conforme visto na figura 11. O motor de passo empregado possui quatro terminais Fases, e um COMUM (terminal ao qual um extremo de cada uma das quatro bobinas estão conectadas entre si). Esse motor funciona convertendo pulsos elétricos em movimentos mecânicos discretos, conforme visto na figura 12. Cada pulso faz com que o motor avance um passo, o que permite o controle preciso da posição e velocidade do rotor.



Figura 12 - Motor de Passo 5 V e Módulo de Controle (Driver ULN2003).⁹

A figura 12 esquematiza o funcionamento básico do motor de passo. A parte estacionária do motor, chamada de estator, tem 8 polos que são espaçados entre 45 graus. A seção que se move do motor, chamada de rotor, é feita de um material ferromagnético que tem 6 dentes espaçados entre 60 graus. Para que o rotor gire um passo, a corrente é aplicada, ao mesmo tempo, através de dois pares de pólos opostos para se tornarem magnetizados. Isso faz com que os dentes do rotor se alinhem com os pólos, como mostrado na figura 12. Para fazer com que o rotor gire 15 graus em sentido horário da sua posição, a corrente através do par de bobina do par 1 é removida e mandada através do par de bobinas 2. E para fazer com que o rotor gire mais 15 graus em sentido horário dessa posição, a corrente é removida do par 2 e enviada ao par 3. O processo continua dessa forma. Para fazer com que o rotor gire em sentido anti-horário, a sequência de disparo do par de bobinas é revertida.

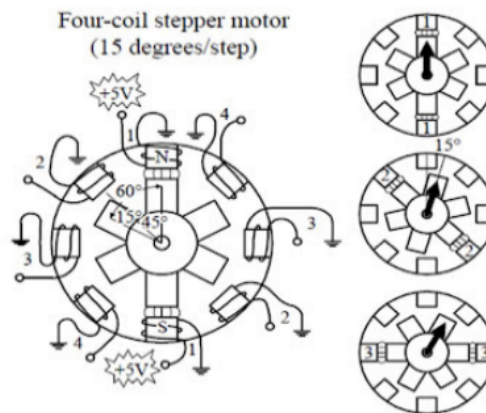


Figura 12 - Funcionamento de um Stepper Motor com 4 bobinas (15 graus/ passo).¹⁰

⁹ Figura retirada do site:

[https://www.eletrogate.com/motor-de-passo-modulo-de-controle-driver-uln2003?utm_source=Site&utm_medium=GoogleMerchant&utm_campaign=GoogleMerchant&utm_source=google&utm_medium=cpc&utm_campaign=\[MC4\]_\[G\]_\[PMax\]_Categorias&utm_content=&utm_term=&gad_source=1&gclid=CjwKCAjw_Na1BhAlEiwAM-dm7CSwW6wFxnH MJ2Rd4HLsYB1c1-3yOoxdAOHEW8B8Tj0XKR-TI9F4xoC56kQAvD_BwE](https://www.eletrogate.com/motor-de-passo-modulo-de-controle-driver-uln2003?utm_source=Site&utm_medium=GoogleMerchant&utm_campaign=GoogleMerchant&utm_source=google&utm_medium=cpc&utm_campaign=[MC4]_[G]_[PMax]_Categorias&utm_content=&utm_term=&gad_source=1&gclid=CjwKCAjw_Na1BhAlEiwAM-dm7CSwW6wFxnH MJ2Rd4HLsYB1c1-3yOoxdAOHEW8B8Tj0XKR-TI9F4xoC56kQAvD_BwE)

¹⁰ Figura e explicações retiradas do Livro 'Practical Electronics for Inventors'- Capítulo 15: 'Motors' - Página 939.

O motor empregado é compatível com o driver ULN2003 para controle de motores de passo. Ele possui as seguintes pinagens principais:

1. **IN1 a IN4:** Entradas de controle (conectadas ao microcontrolador).
2. **OUT1 a OUT4:** Saídas para as fases do motor de passo.
3. **GND:** Terra (conectado ao terra do sistema).
4. **VCC:** Alimentação de 5 V.

O ULN2003 é um driver Darlington de alta tensão e alta corrente. Ele atua como um amplificador, permitindo que correntes mais altas do motor de passo sejam controladas por correntes menores provenientes das portas do microcontrolador. Quando um sinal é enviado para um dos pinos de entrada (IN1 a IN4), o ULN2003 ativa a correspondente saída (OUT1 a OUT4), energizando a fase respectiva do motor de passo. Quando as fases do motor são energizadas em uma sequência específica, o motor de passo gira em movimentos discretos. A sequência e a frequência dos pulsos enviados determinam a direção, velocidade e posição do motor de passo.

No projeto, o motor de passo de 5 V, controlado pelo driver ULN2003, gira um conjunto de engrenagens do mecanismo liberador de ração, esquematizado na figura 10. O microcontrolador envia pulsos de controle para o ULN2003, que então energiza as fases do motor de passo em sequência, resultando em uma rotação controlada das engrenagens, que permite liberar uma quantidade precisa de ração. O mecanismo foi projetado para liberar ração em quantidades precisas: quando o motor é acionado, dois tubos de borracha flexíveis e corrugados (que encaixam-se um ao outro, ver figura 10) giram permitindo que a ração passe entre eles. Ao desligar o motor, as engrenagens param de funcionar e o sistema trava, impedindo a passagem de ração. As peças foram desenhadas e cortadas sob medida para o projeto, conforme a figura 10.

4.2.1.2 Protótipo de sistema com sensor de peso com amplificação

Após a liberação da ração, um sensor de peso mede a quantidade dispensada e envia um sinal amplificado para processamento. O usuário define a quantidade desejada e o intervalo entre as liberações através de botões e uma interface LED com menus interativos. Quando o peso medido atinge o valor configurado, o sistema interrompe a liberação, garantindo precisão. Serão explicados todos os componentes eletrônicos usados nessa etapa e suas interações no circuito.

4.2.1.2.1 Célula de carga de 5 kg

Para este protótipo, foi utilizado um célula de carga com capacidade de até 5 kg, conforme ilustrado na figura 13. Chamados também de sensores de peso, eles são estruturas metálicas as quais são fixados extensômetros, também conhecidas como *strain gauges*¹¹. Esses extensômetros são resistores sensíveis à deformação que alteram sua resistência elétrica quando sujeitos a tensões mecânicas, e estão fixados na região

¹¹ Explicações sobre os *strain gauges* retiradas de:

<https://ensur.com.br/extensometria-strain-gauge-o-que-e-quando-utilizar/#:~:text=O%20extens%C3%B4metro%20ou%20strain%20gauge,de%20um%20equipamento%20ou%20m%C3%A1quina.>

mais frágil da estrutura metálica, conforme apresentado na figura 13, acima e abaixo dos orifícios centrais. Quando a estrutura metálica da célula de carga sofre deformação, seja por compressão ou tração, os extensômetros registram essa deformação como uma variação na resistência. Para medir essa variação de maneira precisa, os extensômetros são conectados em uma configuração de ponte de Wheatstone, permitindo a detecção de pequenas mudanças na resistência, que são então convertidas em um sinal elétrico proporcional ao peso aplicado sobre a célula de carga.

A base de montagem e o prato da balança são rígidos, e são colocados também espaçadores rígidos entre elas e a célula de carga. O objetivo é que qualquer massa sobre o prato da balança resulte um peso que ocasione um estiramento/compressão das regiões delgadas do bloco de alumínio, sobre as quais estão alojados os resistores (*strain gauges*). Sem espaçadores, a carga seria transferida diretamente do prato da balança para a base fixa, sem afetar a célula de carga.

O sensor é composto por dois pares de resistências em série, cujo valor sofre uma pequeníssima alteração em função do tensionamento a que é submetido o suporte metálico à qual elas estão fisicamente acopladas. Essas resistências são montadas de maneira formar uma ponte de Wheatstone.

A ponte de Wheatstone¹² (figura 13) é o método mais empregado para realizar medições na variação da célula de carga. Por meio dela, é possível determinar valores de resistência desconhecida, sendo particularmente útil para a mensuração de resistências muito pequenas, alcançando a escala de mili-ohms. O circuito da ponte consiste em duas disposições série-paralelo de dois resistores de $1\text{ k}\Omega$. Quando os valores são equilibrados em todos os resistores da ponte, a diferença de potencial no centro da ponte, medida entre as disposições dos resistores, deve ser igual a zero.

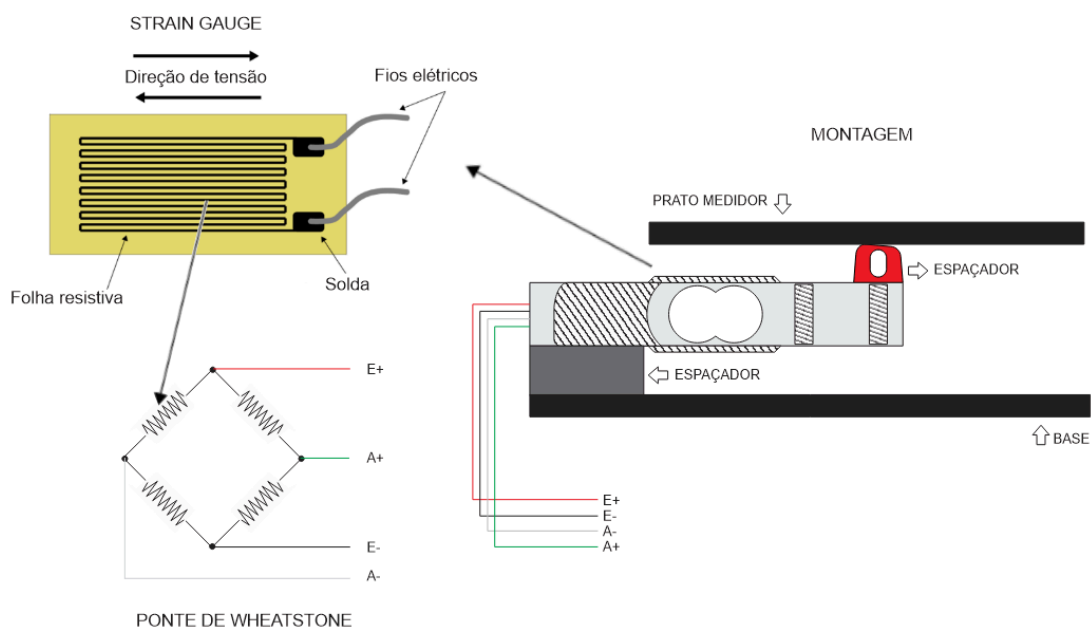


Figura 13 - Célula de Carga e extensômetros conectados como uma Ponte de Wheatstone.¹³

¹² Explicações sobre a Ponte de Wheatstone retiradas de: <https://embarcados.com.br/ponte-de-wheatstone/>.

¹³ Explicações sobre os extensômetros e a sua relação com a Ponte de Wheatstone retiradas de: <https://www.futek.com/strain-gauge-load-cell>.

Essa montagem é muito eficiente para se detectar pequeníssimas variações de resistência entre dois caminhos resistivos, o V_{out} é uma ddp que surge quando há desequilíbrio entre esses caminhos e uma tensão é aplicada sobre a ponte.

Em testes realizados, com uma pressão bastante forte manualmente aplicada, se percebeu apenas uma redução de décimos de Ohm no conjunto das resistências do sensor, e há necessidade de amplificar o sinal que surge quando há desequilíbrio entre os dois caminhos resistivos. Dessa forma, foi necessário definir um sistema de amplificação para o projeto.

4.2.1.2.2 Módulo de Amplificação HX711

Foi escolhido utilizar o Módulo comercial de Amplificação HX711, invés de um circuito de amplificação próprio. O módulo HX711 é conhecido por ser um amplificador estável e preciso, inclusive possui uma biblioteca para pronta implementação em Arduino, de maneira que minimizou o tempo em testes de operação.

O módulo HX711 é um conversor analógico-digital (ADC) de 24 bits amplamente utilizado em aplicações de medição de peso e força, como balanças eletrônicas (conforme ilustrado na figura 14). Ele é frequentemente emparelhado com células de carga, que convertem uma força aplicada em um sinal elétrico. Ele usa um chip de mesmo nome, fabricado pela AVIA semiconductor, que pode detectar pequenos sinais de entrada, de amplitudes de até 20, 40 ou 80 mV, conforme o ganho programado. O sinal amplificado é convertido para um número de 24 bits que é informado a uma taxa de 10 ou 80 Hz por uma comunicação de dois fios a um processador principal, os pinos DT (ou DOUT) e SCK. O processador principal também pode programar o ganho do chip através destes mesmos dois fios.

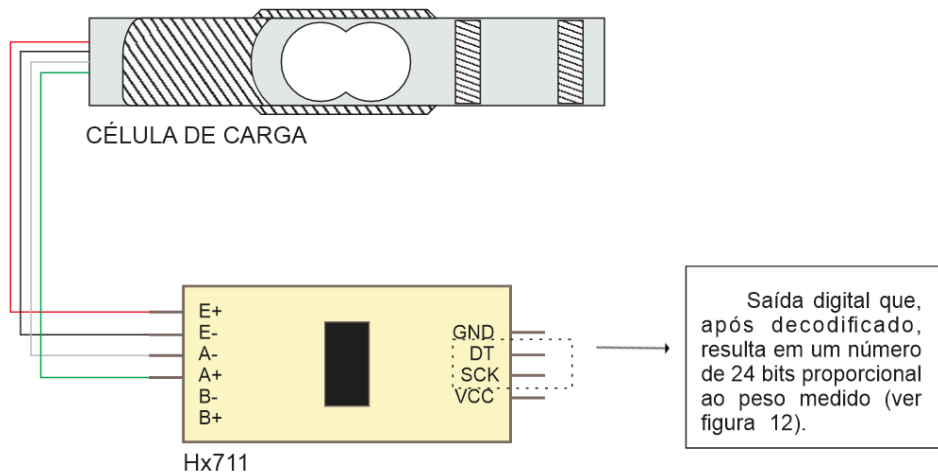


Figura 14 - Conexões entre a Célula de Peso e o Amplificador Hx711.

A célula de carga é conectada ao amplificador de célula de carga HX711 usando quatro fios. Esses quatro fios são vermelho, preto, branco e verde. Abaixo os detalhes de conexão:

- O fio VERMELHO está conectado ao E +.
- O fio PRETO está conectado ao E-.

- O fio BRANCO está conectado a A-.
- O fio VERDE está conectado a A+.

Para entender como os pinos DT (ou DOUT) e SCK operam, no que segue será explorado o algoritmo de comunicação do HX711 com um microcontrolador¹⁴. O HX711 possui os seguintes módulos eletrônicos internos principais: um amplificador de instrumentação que amplifica o sinal analógico proveniente da célula de carga com ganhos de 32, 64 ou 128 (nesse projeto foi selecionado o ganho 128); um filtro de ruído que reduz interferências de ruídos da alimentação de 50 ou 60 Hz sobre o sinal amplificado; um conversor analógico-digital (ADC) de 24 bits que converte o sinal analógico filtrado em um sinal digital de alta resolução; e um controlador de comunicação que gerencia a interface de comunicação com o microcontrolador, utilizando os pinos DT e SCK.

O pino DT (ou DOUT) do módulo HX711 é a linha de dados usada para transmitir os dados digitais do HX711 para um microcontrolador. Após a conversão A/D, o HX711 coloca o resultado digital no pino DT. Fisicamente, quando o HX711 tem dados prontos para serem lidos, o pino DT é colocado em nível baixo (0V); quando não há dados prontos, o pino DT permanece em nível alto (3.3V ou 5V, dependendo da alimentação).

O pino SCK do módulo HX711 é a linha de sincronia que envia pulsos positivos em série (25, 26 ou 27 pulsos, conforme o ganho requerido: 128, 32 ou 64, respectivamente). Este pino é utilizado para sincronizar a transferência de dados entre o HX711 e o microcontrolador, que gera pulsos de "clock" no pino SCK para ler dados quase simultaneamente enviados pelo HX711 ao pino DT. Fisicamente, a transição de baixa para alta (e de volta para baixa) em cada pulso de "clock" faz com que o HX711 envie um bit de dados no pino DT. Após 24 pulsos, são enviados todos os 24 bits do número que corresponde à intensidade do sinal amplificado. O primeiro bit enviado é o mais significativo (MSB, bit 23) e o último é o menos significativo (LSB, bit 0) do número.

No ciclo de leitura do HX711, o módulo realiza uma conversão A/D do sinal analógico e, ao completar a conversão, o pino DT é colocado em nível baixo para indicar que os dados estão prontos. O microcontrolador detecta o nível baixo no pino DT e começa a gerar pulsos no pino SCK. A cada pulso de clock no pino SCK, um bit do dado convertido é enviado pelo pino DT, e esse processo continua até que todos os 24 bits do dado sejam lidos. Após a leitura dos 24 bits, o microcontrolador gera um pulso adicional no pino SCK para resetar o pino DT, colocando-o de volta em nível alto e permitindo que o HX711 inicie uma nova conversão.

Os diagramas de temporização (figura 15) mostram o comportamento dos pinos DT e SCK, com o eixo X representando o tempo e o eixo Y representando o nível lógico dos pinos (alto ou baixo). Esses diagramas ajudam a visualizar como os sinais nos pinos DT e SCK variam ao longo do tempo durante o processo de leitura e conversão do HX711. Um exemplo de ciclo de temporização pode ser visto abaixo:

1. Inatividade:
 - DT está em nível alto, indicando que não há dados prontos.
 - SCK está em nível baixo.
2. Prontidão:
 - DT muda para nível baixo, indicando que os dados estão prontos.

¹⁴ Explicações sobre o funcionamento do Amplificador Hx711 retiradas do seu Datasheet disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1132222/AVIA/HX711.html>.

3. Leitura:
 - O microcontrolador começa a gerar pulsos no pino SCK.
 - A cada pulso de SCK, um bit é transmitido no pino DT.
4. Finalização:
 - Após 24 pulsos, os dados foram completamente lidos.
 - Um pulso adicional é gerado no pino SCK para resetar DT para nível alto.

Resumindo, os pinos DT e SCK do módulo HX711 funcionam como uma interface de comunicação serial simples. O pino DT é usado para transmitir dados digitais convertidos, e o pino SCK é utilizado para sincronizar essa transmissão. A correta leitura dos dados envolve detectar quando DT está pronto (nível baixo), gerar pulsos de “clock” no SCK, e ler os bits de dados a cada pulso, conforme pode ser visto na figura 15.

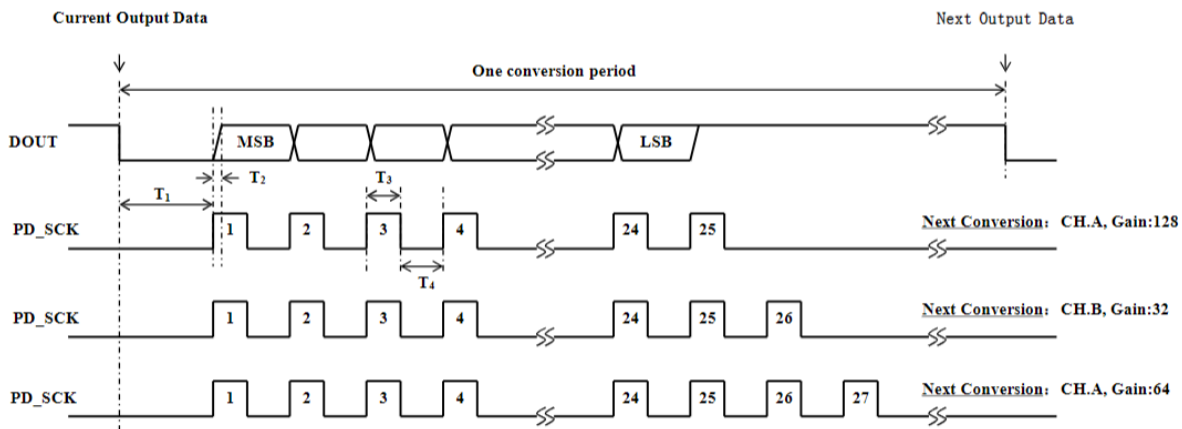


Figura 15 - Saída de dados, entrada e tempo de seleção de ganho e controle.¹⁵

Dessa forma, foi utilizado o Módulo HX711 de amplificação no projeto, pois ele amplifica a baixa saída elétrica das células de carga e, em seguida, esse sinal amplificado e convertido digitalmente, é lido por pinos digitais de um Arduino, e decodificado por uma biblioteca (<HX711.h>) de onde é derivado o valor do peso medido.

4.2.1.3 Interface com o usuário (a)

Um display LCD de duas colunas e 16 linhas, com módulo I2C acoplado (ilustrados na figura 16), foi utilizado para apresentar as opções para o usuário bem como mostrar as modificações efetuadas ou opções escolhidas. O módulo I2C foi o utilizado devido à facilidade que ele oferece, em termos de conexões do display, bastando apenas uma conexão de dois fios para estabelecer a comunicação com o microcontrolador Arduino

¹⁵ Figura retirada do Datasheet do Módulo de Amplificação HX711 disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1132222/AVIA/HX711.html>.

UNO.

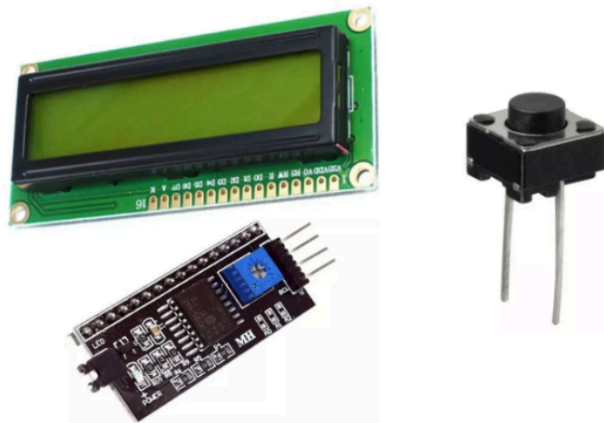
O display LCD 16x2 possui 16 pinos, enquanto o módulo serial I2C possui 4 pinos principais. O módulo I2C é projetado para ser montado diretamente na parte traseira do display LCD, facilitando as conexões:

- VSS (LCD) ao GND (I2C): Terra.
- VDD (LCD) ao VCC (I2C): Alimentação (5V).
- VO (LCD): Ajuste de contraste (geralmente controlado pelo módulo I2C).
- RS (LCD) ao RS (I2C): Seleção de registro.
- RW (LCD): Conectado ao GND diretamente no módulo I2C.
- E (LCD) ao E (I2C): Habilitação.
- D4 a D7 (LCD) aos pinos de dados do I2C: Transferência de dados.

O módulo I2C se comunica com o microcontrolador através de apenas dois pinos de dados, além da alimentação e terra:

- GND (I2C) ao GND (Microcontrolador): Terra.
- VCC (I2C) ao 5V (Microcontrolador): Alimentação.
- SDA (I2C) ao SDA (Microcontrolador): Linha de dados I2C.
- SCL (I2C) ao SCL (Microcontrolador): Linha de sincronização ou clock.

O microcontrolador envia dados através do protocolo I2C para o módulo I2C, que os converte para o padrão do Driver do display LCD¹⁶. As mensagens são enviadas por comandos definidos pela biblioteca “LiquidCrystal_I2C.h” para impressão de mensagens no mesmo. A biblioteca que gerencia o protocolo I2C no Arduino chama-se wire. O módulo I2C converte esses dados e os transmite para o display LCD 16x2, que exibe as informações de forma clara e precisa. Essa configuração reduz a complexidade da fiação e facilita a integração do display no sistema, garantindo uma interface de usuário eficiente e prática.



¹⁶ Explicação do funcionamento do Módulo Display LCD 16x2 com o Módulo Serial I2C retirada de: <https://www.blogdarobotica.com/2022/05/02/como-utilizar-o-display-lcd-16x02-com-modulo-i2c-no-arduino/>.

Figura 16 - Módulo Display LCD 16x2, Módulo Serial I2C Para Display LCD 16x2 e botão de contato¹⁷.

Para o controle das funcionalidades do equipamento, são utilizados três botões de contato, os quais estarão ligados aos pinos digitais 8, 9 e 10 do arduino:

- Botão da esquerda e botão da direita.
- Botão de seleção.

Para identificar o aperto nos botões usou-se uma biblioteca chamada de EventButton.h. Esta biblioteca permite verificar se eles estão sendo clicados ou pressionados por um tempo, demais detalhes da implementação serão explorados na seção 4.2.2 do relatório, bem como a descrição detalhada do menu que o usuário terá para escolher as funcionalidades, como os componentes interagem e como os botões são usados na seleção.

4.2.1.4 Fonte DC

No teste de funcionamento da balança e do circuito, uma bateria de 9 V estava sendo suficiente. Porém, no teste de funcionamento de longa duração entre as deposições de comida, a bateria recarregável de 9 V não foi adequada para o funcionamento do projeto. Desta forma, o teste de funcionamento de longa duração foi executado com um carregador de celular de 5 V. Isso não significa que a bateria esteja descartada do projeto, porém ela deve ser substituída por uma bateria de lítio de maior capacidade, como as encontradas em celulares. Testes ainda teriam que ser realizados para a viabilidade desse tipo de fonte de energia e para caracterizar a capacidade de armazenamento da célula de energia necessária ao projeto.

4.2.1.5 Montagem do circuito

Conforme ilustrado nas figuras 17 e 18, o projeto utiliza um Arduino Uno como microcontrolador para gerenciar todo o processo. Quando o motor de passo é acionado, a ração cai e a célula de carga detecta o peso, fornecendo uma tensão elétrica analógica ao Módulo Amplificador de Carga HX711. Este módulo amplifica e converte digitalmente a saída da célula de carga, enviando o valor amplificado para o Arduino. O Arduino converte os dados em valores de peso em gramas com o auxílio da biblioteca HX711.h incorporada ao código. O operador ajusta a quantidade de ração e os horários de liberação através de três botões, podendo visualizar as opções e os resultados no LCD 16x2, que utiliza o módulo serial I2C para simplificar a fiação.

¹⁷ Imagem do Módulo Display LCD 16x2 e do Módulo Serial I2C Para Display LCD 16x2 retirada de: <https://www.casadarobotica.com/display/display/display-lcd-amarelo-16x2-modulo-i2c-serial>. Imagem do botão de contato retirada de: https://www.msseletronica.com.br/detalhes/chave-tactil-6x6x4-3mm-2-terminais-kfc-a06-micro-chave-micro-switch_pid1127.html.

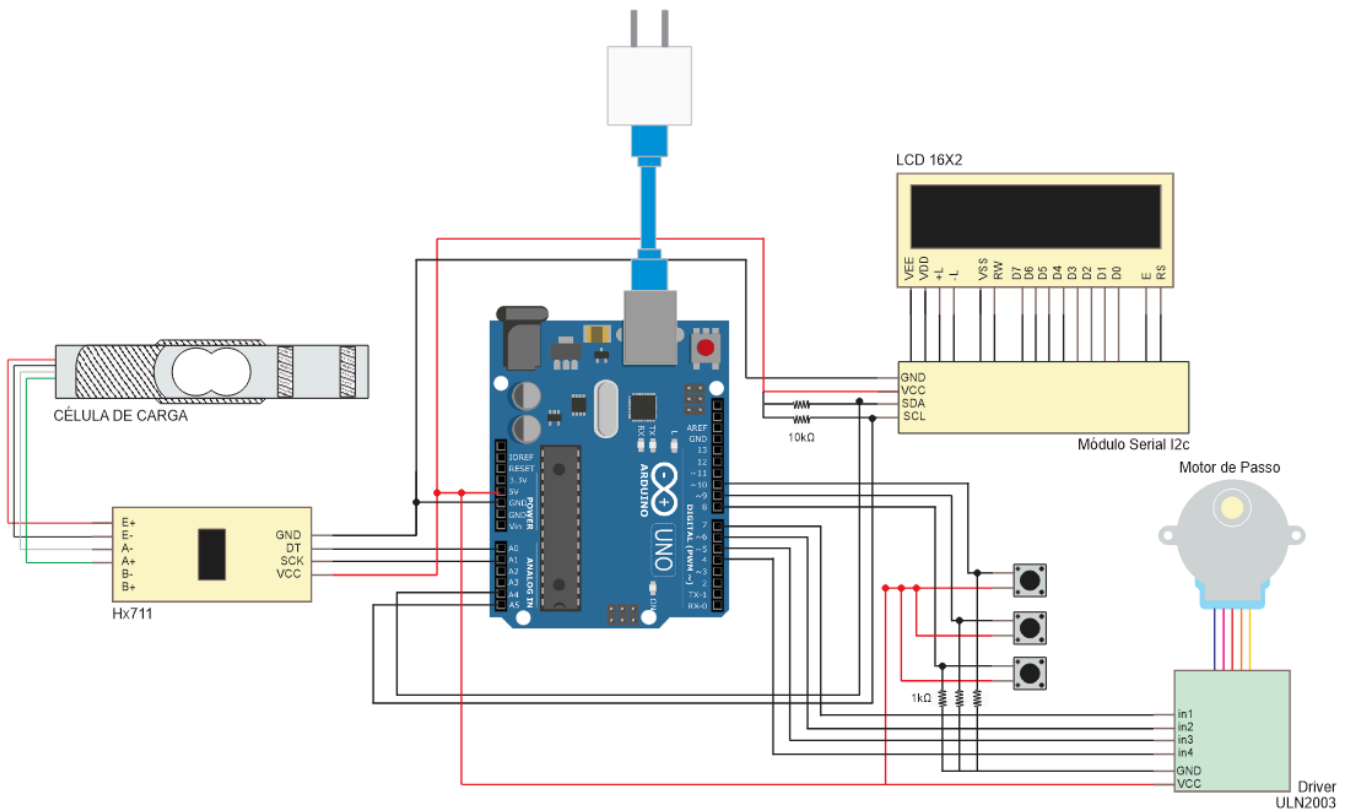


Figura 17 - Montagem de prototipagem usando um Arduino Uno como microcontrolador e os componentes descritos nas seções anteriores: Célula de Peso, Amplificador Hx711 e Módulo LCD 16X2 com módulo I2C acoplado.

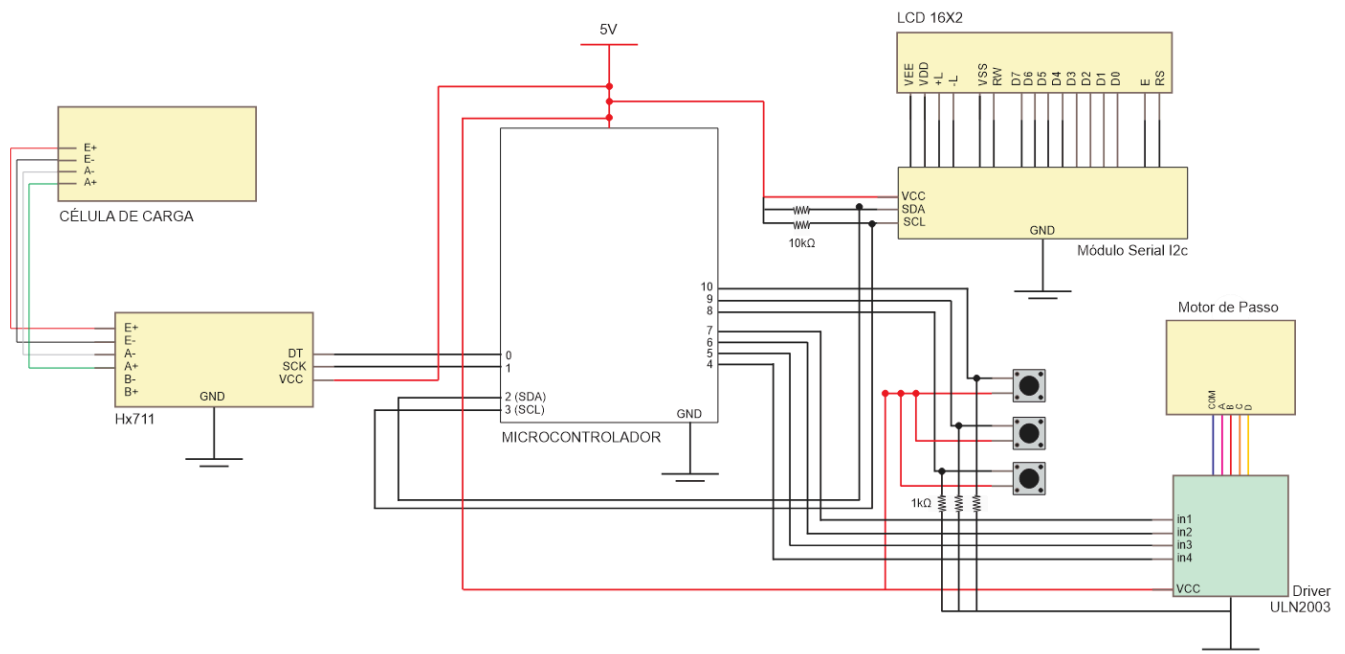


Figura 18 - Esquema de circuito.

4.2.1.6 Componentes Eletrônicos

Para a concepção do protótipo de teste deste projeto, serão listados abaixo todos os componentes eletrônicos utilizados e os valores de mercado.

N.º	Item	Quantidade	Valor Unitário	Site	Valor (R\$)
1a	Arduíno Uno	1	52,16	https://www.eletrogate.com	52,16
1b	Microcontrolador* atmega328P	1	11,78	https://pt.aliexpress.com	11,78
2	Fios	40	0,34	https://produto.mercadolivr.com.br	13,90
3a	Carregador de celular	1	26,62	https://www.amazon.com.br	26,62
3b	Bateria de lítio 4000 mAh recarregável**	1	37,89	https://pt.aliexpress.com	37,89
4	Sensor de Peso + Módulo Conversor Amplificador HX711	1	25,00	https://produto.mercadolivr.com.br	25,00
5	Módulo LCD I2C	1	9,40	https://www.eletrogate.com	9,40
6	LCD 16X2	1	17,96	https://www.eletrogate.com	17,96
7	Stepper Motor	1	13,78	https://www.a2robotics.com.br/	13,78
8	Botão de contato	50	0,38	https://produto.mercadolivr.com.br	1,14
Total (R\$) com item 1a e 3a***					159,96
Total (R\$) com item 1b e 3a					119,58
Total (R\$)					130,85

N.º	Item	Quantidade	Valor Unitário	Site	Valor (R\$)
	com item 1b e 3b				
	Total (R\$) com item 1a e 3b				171,23

Tabela 4 - Lista de Componentes eletrônicos.

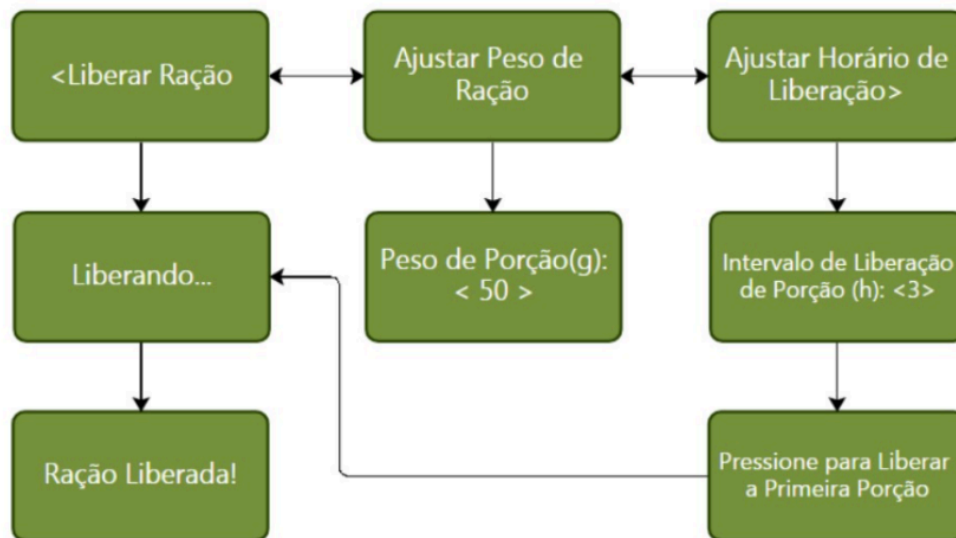
* Item opcional a ser testado, inserido no orçamento por ser uma opção de menor custo a fins de comparação.

**Item opcional a ser testado conforme explicado na seção 4.2.1.4.

***Para este trabalho, foram utilizados os itens 1a e 3a.

4.2.2 Interface com o (a) usuário (a)

Conforme visto no fluxograma 1 e na figura 19, o usuário pode navegar por 3 diferentes opções de configuração e função. Sendo essas funções navegadas pelos botões da esquerda e da direita, e selecionadas pelo botão central. As opções são: a liberação imediata de uma porção de ração, ajuste do peso de ração a ser liberado e o ajuste e começo de um ciclo de liberação de porções que irá se repetir até que outro horário seja definido.



Fluxograma 1 - Mapa de navegação do (a) usuário (a).

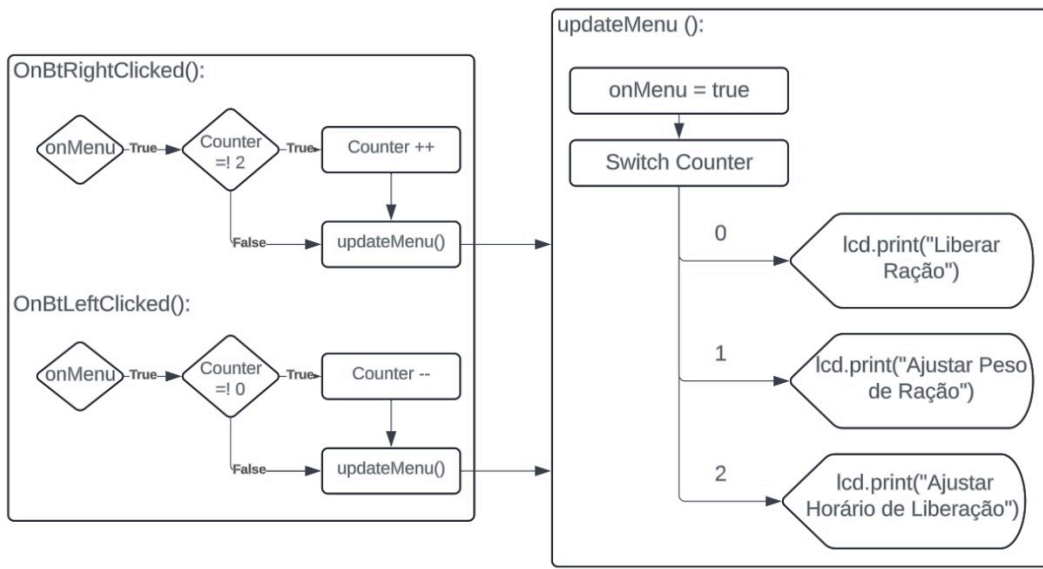


Figura 19 - Protótipo final da interface, a qual está localizada na base da estrutura (ver figura 6).

4.2.2.1 Navegação

A navegação da aplicação é feita a partir da detecção de cliques dos Botões de Contato e sua atribuição a duas funções, `OnbtRightClicked()` e `OnbtleftClicked()`, que mudam um valor de *index*, chamado *Counter*, e acionam uma terceira função, `updateMenu()`, que atualiza o LCD com a opção correspondente ao *index* selecionado.

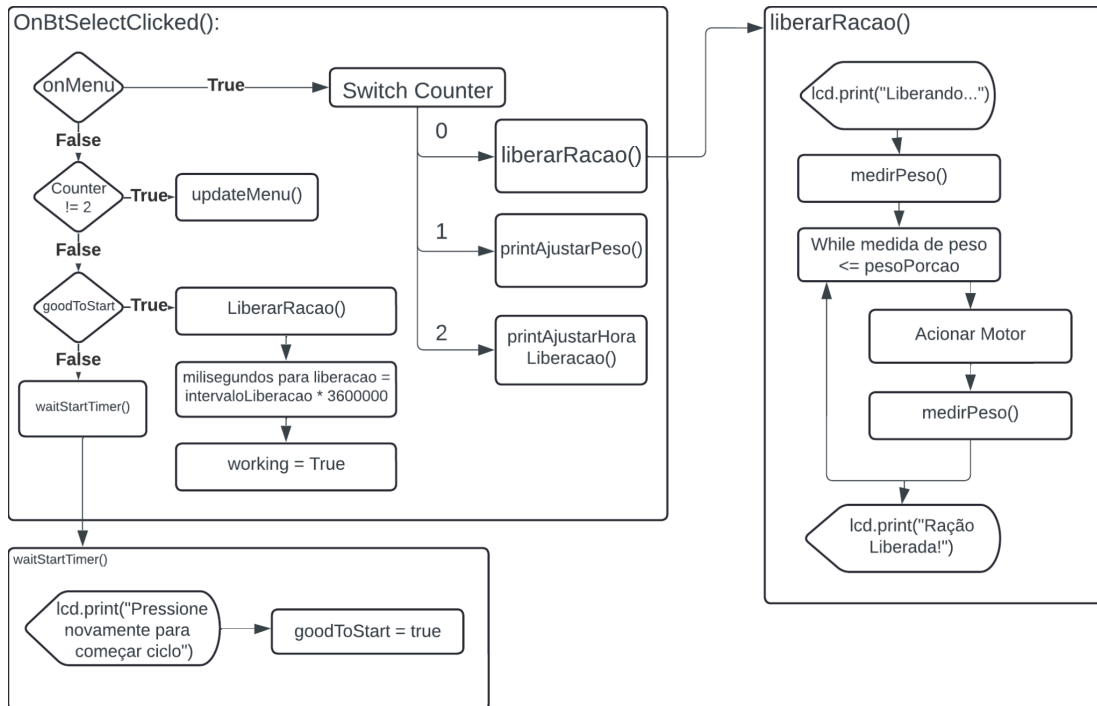
É importante ressaltar que nenhuma troca de opção é permitida quando a variável booleana `onMenu` é falsa, pois os mesmos botões usados para a troca de opção no menu são depois usados para a mudança de variáveis como o peso da porção e a quantidade de horas por ciclo de liberação.



Fluxograma 2 - Funções de cliques direito e esquerdo dos botões e função de atualização de menu .

O quadro à esquerda do fluxograma 3, representa o funcionamento dos botões esquerdo e direito, cuja função é selecionar uma entre as três opções apresentadas no menu. O quadro à direita no fluxograma 3, indica o que acontece ao apertar o botão central, cuja função é executar a opção selecionada do menu. Sob o apertado do botão central, a função `OnBtSelectClicked()` é acionada, se o usuário estiver na opção no menu, sinalizado pela variável `onMenu`, uma função correspondente ao seu *index Counter* será chamada. Abaixo está a função `liberarRacao()` como exemplo.

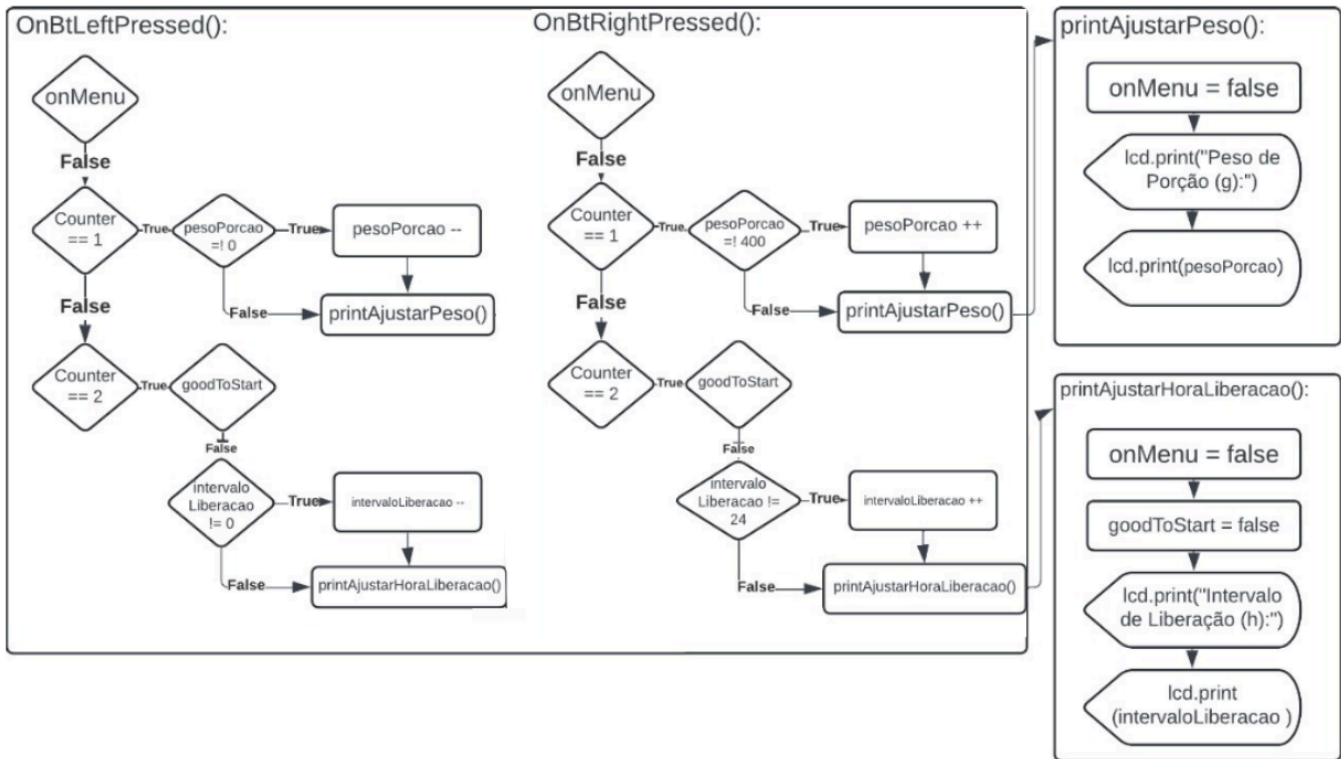
Um outro cenário para o acionamento do botão do meio, é quando o usuário estiver satisfeito com a variável editada e quiser voltar para o menu ou prosseguir com um novo ciclo de liberação. Neste caso, `onMenu` será falsa, o que levará para o acionamento da função `updateMenu()` novamente. Caso o usuário esteja na opção "Ajustar Hora de Liberação", caracterizada pelo *Counter* 2, e apertar o botão para selecionar a função `waitStartTimer` será ativada, mudando o valor da booleana `goodToStart` para *true*. Quando o botão for novamente pressionado, a ração será liberada e o timer para a liberação da próxima porção será ativado.



Fluxograma 3 - Funções de seleção de clique de botão, função de liberação de ração e timer.

4.2.2.2 Configurações

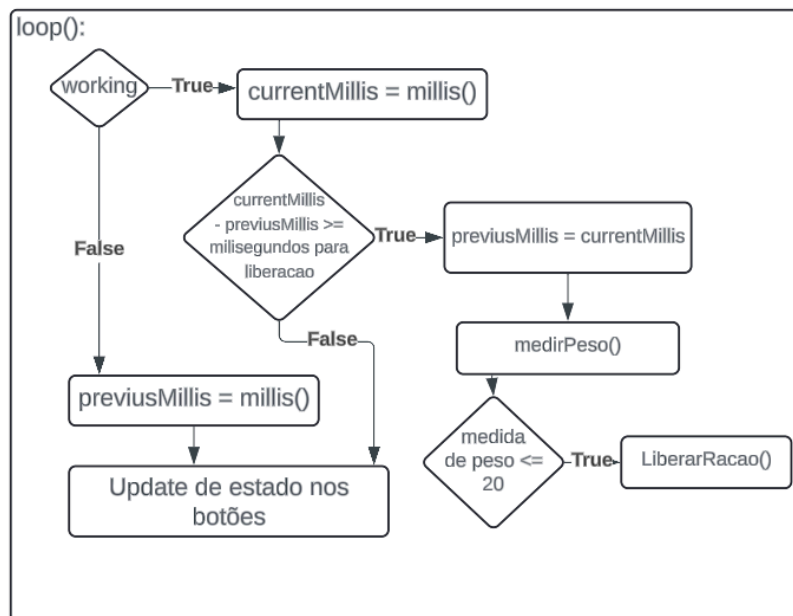
Quando o usuário selecionar uma opção com a possibilidade da edição de alguma variável, como o ajuste de peso ou horário de liberação, as funções `OnBtRightPressed()` e `OnBtLeftPressed()` são usadas para aumentar ou diminuir o parâmetro em ajuste (tempo em horas ou peso em gramas). Para implementar essa mudança de atribuição, inicialmente o código verifica se o usuário está fora do menu principal (corresponde à situação `onMenu = false` no diagrama abaixo), e a partir desse momento os botões aumentam (botão da direita) ou reduzem (botão da esquerda) o parâmetro.



Fluxograma 4 - Funções de edição de variável para ajuste de peso e hora de liberação .

4.2.2.3 Funcionamento do *Timer*

Para o funcionamento do timer a função nativa `millis()` é usada para recuperar a contagem de tempo desde que o microcontrolador foi iniciado. Quando a booleana `working for true`, o timer começa.



Fluxograma 5 - Funcionamento do *Timer*.

4.2.3 Software

Para a implementação do código do projeto, foi utilizado Arduino IDE na versão 2.1.1. As bibliotecas empregadas foram <HX711.h>, para a comunicação com o módulo de amplificação HX711, responsável por processar os sinais da célula de carga e fornecer medições de peso precisas; <LiquidCrystal_I2C.h>, para facilitar a interação com o display LCD 16x2 via interface I2C, permitindo a exibição de informações de forma clara e eficiente; <Stepper.h>, para controlar o motor de passo que libera a ração no pote, garantindo que o motor funcione até que o peso desejado seja alcançado e pare quando o peso requerido pelo usuário for atingido; e <EventButton.h>, para gerenciar os acionamentos dos três botões de contato (do tipo Chave Tátil 6x6x5mm com 2 terminais), permitindo a detecção de eventos de cliques para realização de ações correspondentes no código.

Abaixo, segue uma lista das principais funções de cada biblioteca:

1. <HX711.h>

- a. `begin(uint8_t data_pin, uint8_t clock_pin)`: Inicialização da balança com os pinos especificados para os sinais de dados e clock.
- b. `set_scale(float scale)`: Define a escala de calibração para a balança.
- c. `tare()`: Zera a balança, desconsiderando o peso atual da estrutura.
- d. `get_units(uint8_t times)`: Retorna a média de um número especificado de medições, em unidades de peso calibradas.
- e. `power_up()`: Liga o sensor de peso.
- f. `power_down()`: Desliga o sensor de peso.

2. <EventButton.h>

- a. `EventButton(uint8_t pin)`: Construtor que inicializa um botão no pino especificado.
- b. `setClickHandler(void (*handler)(EventButton&))`: Define a função que será chamada quando o botão for clicado.
- c. `setPressedHandler(void (*handler)(EventButton&))`: Define a função que será chamada quando o botão for pressionado.
- d. `update()`: Atualiza o estado do botão, verificando se foi pressionado ou clicado.

3. <LiquidCrystal_I2C.h>

- a. `LiquidCrystal_I2C(uint8_t lcd_addr, uint8_t lcd_cols, uint8_t lcd_rows)`: Construtor que inicializa o display LCD com o endereço I2C e o número de colunas e linhas especificadas.
- b. `init()`: Inicializa o display LCD.
- c. `backlight()`: Liga a luz de fundo do display.
- d. `clear()`: Limpa o display.
- e. `setCursor(uint8_t col, uint8_t row)`: Define a posição do cursor no display.
- f. `print(const char* str)`: Imprime uma string na posição atual do cursor.

4. <Stepper.h>

- a. `Stepper(int number_of_steps, int motor_pin_1, int motor_pin_2, int motor_pin_3, int motor_pin_4)`: Construtor que inicializa o motor de passo com o número de passos por revolução e os pinos utilizados.
- b. `setSpeed(long whatSpeed)`: Define a velocidade do motor de passo em rotações por minuto (RPM).
- c. `step(int steps_to_move)`: Move o motor de passo pelo número especificado de passos.

5. Funções criadas para o código:

- a. `onbtRightClicked(EventButton& eb)`: Função chamada quando o botão direito é clicado. Atualiza o menu.
- b. `onbtLeftClicked(EventButton& eb)`: Função chamada quando o botão esquerdo é clicado. Atualiza o menu.
- c. `onbtRightPressed(EventButton& eb)`: Função chamada quando o botão direito é pressionado. Ajusta o peso da porção ou o intervalo de liberação.
- d. `onbtLeftPressed(EventButton& eb)`: Função chamada quando o botão esquerdo é pressionado. Ajusta o peso da porção ou o intervalo de liberação.
- e. `onbtSelectClicked(EventButton& eb)`: Função chamada quando o botão de seleção é clicado. Executa a ação selecionada no menu.
- f. `setup()`: Função de inicialização do Arduino. Configura a balança, o display LCD, os botões e o motor de passo.
- g. `loop()`: Função principal do Arduino que é executada repetidamente. Verifica o estado dos botões e libera a ração no tempo especificado.
- h. `medirPeso()`: Mede o peso utilizando a balança e atualiza a variável `medidaMod`.
- i. `liberarRacao()`: Libera a ração enquanto a medida for menor que o peso da porção especificada.
- j. `printAjustarPeso()`: Ajusta o peso da porção.
- k. `printAjustarHoraliberacao()`: Ajusta o intervalo de liberação da ração.
- l. `waitStartTimer()`: Inicia o temporizador para liberar a ração.
- m. `updateMenu()`: Atualiza o menu no display LCD.

Um ponto que talvez merecesse mais atenção na escrita do código, seria o processo de calibração do Módulo HX711, onde a função `set_scale` é utilizada para definir o fator de escala do mesmo. Este fator de escala é essencial para converter a leitura bruta do ADC (Conversor Analógico-Digital) em uma unidade de medida significativa, como gramas ou quilogramas. Nos parágrafos que seguem, descrevemos com pouco mais detalhes esse processo de calibração.

Para obter o valor correto do fator de escala, é necessário calibrar a balança. O processo de calibração envolve preparar o sistema, zerando a balança com `scale.tare()` para descontar o peso da estrutura ou recipiente, e em seguida, adicionando um peso conhecido sobre a célula de carga, como um objeto de 1 kg. Após isso, lê-se o valor bruto do ADC sem o fator de escala ajustado, o que pode ser feito com um código simples, como `Serial.println(scale.get_units(10))`; para obter a média de 10 leituras.

Para calcular o fator de escala, divide-se a leitura bruta pelo peso conhecido. Por exemplo, se a leitura bruta for 439295.459 para uma massa de 1 kg, então o fator de escala será 439295.459. Este valor representa o fator de escala calibrado, garantindo que a leitura bruta do HX711 seja convertida corretamente para o peso real em quilogramas ou outra unidade desejada.

O código utilizado para realizar a calibração é mostrado a seguir:

```
1. #include "HX711.h"
2.
3. const int LOADCELL_DOUT_PIN = A0;
4. const int LOADCELL_SCK_PIN = A1;
5.
6. HX711 scale;
7.
8. void setup() {
9.     Serial.begin(9600);
10.    scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
11.    scale.tare(); // Zera a balança
12.    Serial.println("Coloque um peso conhecido na célula de carga...");
13.    delay(5000); // Aguarde 5 segundos para colocar o peso
14. }
15.
16. void loop() {
17.     // Lê o valor bruto sem fator de escala
18.     Serial.print("Leitura bruta: ");
19.     Serial.println(scale.get_units(10), 5); // Média de 10 leituras
20.     delay(1000);
21. }
```

Durante a calibração, um peso conhecido, por exemplo, 1 kg, foi colocado na célula de carga e o valor impresso no monitor serial foi observado. O valor obtido foi 439295.459, e portanto, o fator de escala foi definido no código como `scale.set_scale(439295.459f)`. Esse valor é o fator de escala calibrado que converte as leituras brutas do HX711 em medidas reais de peso, obtido através do processo de calibração onde um peso conhecido determina a relação entre a leitura bruta do ADC e o peso real.

O código, em sua totalidade, encontra-se no Apêndice A (pag. 38), sobre o título “Código Principal”.

5. Implementação

5.1 Montagem da estrutura final

Conforme ilustrado na figura 6, a estrutura final do projeto consiste em um reservatório de ração

acoplado a um mecanismo de engrenagens projetado para liberar e bloquear a ração, seguido por uma rampa que direciona a ração para o pote. Na base da estrutura, há uma caixa que abriga os componentes eletrônicos na parte inferior e possui um encaixe frontal para o pote de ração. O sensor de peso está posicionado sob o pote de forma que o peso fique centralizado na extremidade do sensor.

À frente do pote de ração, encontra-se a interface do usuário, equipada com botões e um display LCD para navegação entre as funcionalidades do sistema.

5.2 Verificação

A verificação é o processo de garantir que os resultados do projeto atendam às especificações concebidas na tabela 2. Cada especificação foi verificada, de forma que confirme que a especificação é atendida.

Necessidade #	Necessidade do Usuário	Especificação	Verificação
1	Deve ter porções de comida adequadas ao seu porte.	Permitir a seleção de porções de ração entre 0 g e 400 g, ajustável em incrementos de 1g.	A seleção de porções foi testada, e o sistema permite o ajuste preciso de 1 g dentro do intervalo especificado, garantindo porções de até 400 g.
2	Deve ter uma quantidade limitada de porções ao dia.	Permitir a configuração de até 24 porções de ração por dia, com intervalos ajustáveis entre 1 e 24 horas.	O sistema foi configurado e testado para permitir até 24 porções diárias, com intervalos entre as porções ajustáveis de 1 hora, atendendo à especificação proposta.
3	Não deve acumular comida no pote ao longo do dia.	Detectar quando a ração no pote excede 20 g e interromper a liberação de novas porções até que o peso da ração seja inferior a 20 g.	O sensor de peso foi calibrado e testado para interromper a liberação de ração assim que detecta 20 g ou mais no pote, prevenindo o acúmulo e garantindo que a especificação seja atendida.

4	Ter um reservatório suficientemente grande para não precisar de reposições diárias de ração.	Um reservatório com espaço para abrigar 1 kg de ração.	O reservatório foi dimensionado e testado, demonstrando capacidade para armazenar até 1 kg de ração, cumprindo assim a necessidade de não precisar de reposições diárias.
5	Que o pote e reservatório de comida seja removível e fácil de lavar.	Ter 1 pote e 1 reservatório removíveis com sistema de encaixe simples, permitindo fácil retirada para limpeza.	O pote e o reservatório foram projetados com um sistema de encaixe simples, sendo facilmente removíveis e laváveis, confirmando a facilidade de manutenção conforme especificado.
6	Ter um produto com display mostrando a quantidade de comida e de tempo configurados para controle.	Ter 1 display digital que exibe a quantidade de comida (em gramas) e o tempo configurado (em horas), com capacidade para ajustar esses valores usando uma interface de 3 botões.	O display digital foi testado, e exibe corretamente as informações de quantidade de ração e tempo configurado, com uma interface de 3 botões funcionando conforme o esperado.

Tabela 5 - Verificação das Especificações das Necessidades do Usuário.

5.3 Validação

A validação é o processo de demonstrar que o projeto final atende às necessidades do usuário, conforme concebidas na tabela 1. Cada necessidade do usuário teve validação que confirma que a necessidade do usuário é atendida.

Necessidade	Usuário	Necessidade do Usuário	Validação
1	Pet.	Deve ter porções de comida adequadas ao seu porte.	Foram realizados testes com pets de diferentes tamanhos, e as porções fornecidas pelo dispositivo foram ajustadas conforme necessário, comprovando que o sistema atende às necessidades alimentares de cada pet.
2	Pet.	Deve ter uma quantidade limitada de porções ao dia.	O sistema foi configurado para fornecer diferentes quantidades de porções ao longo do dia e os testes mostraram que a quantidade de porções não excedeu o limite programado, validando a capacidade de limitar as porções diárias.
3	Pet.	Não deve acumular comida no pote ao longo do dia.	Durante a operação contínua, o sistema foi monitorado e validado para garantir que a liberação de ração parasse quando o pote atingiu 20 g, prevenindo o acúmulo excessivo e evitando desperdício.
4	Tutor do pet.	Ter um reservatório suficientemente grande para não ter que fazer uma reposição diária de ração.	O reservatório foi testado com uma carga completa de 1 kg de ração e os testes confirmaram que o nível de ração permaneceu adequado ao longo do dia, atendendo à necessidade de evitar reposições diárias.
5	Tutor do pet.	Que o pote e reservatório de comida seja removível e fácil de lavar.	O pote e o reservatório foram removidos e limpos por 3 usuários, e a facilidade de

Necessidade	Usuário	Necessidade do Usuário	Validação
6	Tutor do pet.	Poder controlar a quantidade de comida e o tempo de funcionamento do produto.	<p>desmontagem e limpeza foi confirmada, atendendo ao requisito de praticidade na manutenção.</p> <p>A funcionalidade de um display e de botões foi avaliada com base em testes de uso, e os resultados mostraram que os controles para ajustar a quantidade de comida e o tempo de operação foram precisos e intuitivos, atendendo às expectativas de controle do usuário.</p>

Tabela 6 - Validação da lista de Necessidades do Usuário.

6. Conclusão

A realização deste projeto teve como objetivo desenvolver uma máquina dispensadora de ração que entregasse a quantidade desejada de ração nas quantidades e horários programados. Consegui atingir esse objetivo, embora tenha encontrado desafios inesperados ao longo do caminho. O projeto acabou sendo mais caro e demorado do que eu inicialmente imaginei. O custo total foi mais elevado devido ao preço dos componentes. Além disso, a implementação do código e a escolha dos componentes exigiram mais tempo do que o previsto.

Uma das partes mais gratificantes do projeto foi o desenvolvimento da estrutura da máquina. A experiência prévia com peças cortadas a laser facilitou a construção da estrutura em si, mas a parte mais interessante foi o sistema de tubos de borracha corrugados que controla a passagem de ração. Este sistema, que permite a passagem ou o bloqueio da ração, foi um desafio técnico que foi particularmente empolgante e inovador.

O diferencial técnico do projeto reside na funcionalidade de checar o peso da ração antes de sua liberação. O sistema verifica se há mais de 20 gramas de ração disponível no próximo horário programado; caso

afirmativo, ele não libera mais ração. Essa abordagem evita o desperdício, garantindo que a quantidade de ração dispensada seja sempre adequada.

Embora o projeto ainda não esteja com um preço competitivo no mercado, o desenvolvimento e a implementação foram valiosos para minha formação. A experiência adquirida ao longo do processo, incluindo a solução de problemas complexos e a aplicação de conhecimentos de eletrônica, foram fundamentais para o sucesso da máquina. Com relação ao preço, não saberia quantificar exatamente o quanto poderia ser reduzido, mas comprando diretamente de fornecedores a preço de fábrica, esse preço seria significativamente reduzido. Além disso, poderíamos usar um microprocessador mais barato ainda compatível com a necessidade desse projeto, que seria ter 11 pinos digitais, sendo dois deles com capacidade de comunicação I2C. O tamanho da memória flash também teria que ser avaliado em uma etapa futura com esse propósito comercial para dimensionar os requisitos mínimos do processador.

Apêndices

Apêndice A: Código principal

Abaixo é apresentado o código completo utilizado no projeto, cuja lógica de operação foi previamente detalhada nos fluxogramas da seção 4.2.2:

```
1. // Carrega as bibliotecas necessárias
2. #include <HX711.h>
3. #include <EventButton.h>
4. #include <LiquidCrystal_I2C.h>
5. #include <Stepper.h>
6.
7. // Define os valores usados no timer para liberação de ração
8. unsigned long previousMillis = 0; // Armazena o tempo da última atualização
9. const long interval = 3600000; // Intervalo de 1 hora (3600000 milissegundos)
10. unsigned long currentMillis = 0;
11. unsigned long milisLiberacao = 0;
12.
13. // Define os pinos que serão utilizados para ligação ao display LCD
14. LiquidCrystal_I2C lcd(0x27,16,2);
15. // Número de passos do motor de passo
16. #define STEPS 200
17. // Define o motor de passo e seus pinos
18. Stepper stepper(STEPS, 4, 6, 5, 7);
19. // Define os pinos da balança
20. #define pinDT A0
21. #define pinSCK A1
22. #define pesoMin 0.001 // Peso mínimo que a balança pode medir
23. #define pesoMax 30.0 // Peso máximo que a balança pode medir
24.
25. HX711 scale; // Instancia um objeto da classe HX711 para a balança
26. float medida=0; // Declaração da variável de medida
```

```

27. float medidaMod;          // Variável para armazenar a medida modificada
28.
29. // Define os pinos dos botões
30. EventButton btRight(8);
31. EventButton btSelect(9);
32. EventButton btLeft(10);
33.
34. int counter = 0;          // Índice para seleção de opções no menu
35. bool onMenu = true;      // Flag para indicar se estamos no menu
36.
37. // Variáveis operacionais
38. int pesoPorcao = 50;      // Peso da porção de ração em gramas
39. int intervaloLiberacao = 3; // Intervalo de liberação em horas
40. bool goodToStart = false; // Flag para indicar se está pronto para começar
41. bool working = false;    // Flag para indicar se o sistema está em operação
42.
43. // Função chamada quando o botão direito é clicado
44. void onbtRightClicked(EventButton& eb) {
45.     if(onMenu){
46.         if(counter != 2){counter++;} // Impede que o índice vá além de 2
47.         updateMenu();                // Atualiza o menu
48.     }
49. }
50. // Função chamada quando o botão esquerdo é clicado
51. void onbtLeftClicked(EventButton& eb) {
52.     if(onMenu){
53.         if(counter != 0){counter--;} // Impede que o índice vá antes de 0
54.         updateMenu();                // Atualiza o menu
55.     }
56. }
57. // Função chamada quando o botão direito é pressionado
58. void onbtRightPressed(EventButton& eb) {
59.     if(!onMenu){
60.         if(counter == 1){
61.             if(pesoPorcao != 400){ //peso máximo
62.                 pesoPorcao++;      // Aumenta o peso da porção
63.                 printAjustarPeso(); // Ajusta o peso no display
64.             }
65.         }
66.         if(counter == 2){
67.             if(!goodToStart){
68.                 if(intervaloLiberacao != 24){ // 24 horas de um dia para o outro
69.                     intervaloLiberacao++; // Aumenta o intervalo de liberação
70.                     printAjustarHoraLiberacao(); // Ajusta o intervalo no display
71.                 }
72.             }
73.         }
74.     }
75. }
76. // Função chamada quando o botão esquerdo é pressionado
77. void onbtLeftPressed(EventButton& eb) {
78.     if(!onMenu){
79.         if(counter == 1){
80.             if(pesoPorcao != 0){
81.                 pesoPorcao--;      // Diminui o peso da porção

```



```

82.     printAjustarPeso();          // Ajusta o peso no display
83.   }
84. }
85. if(counter == 2){
86.   if(!goodToStart){
87.     if(intervaloLiberacao != 0){
88.       intervaloLiberacao--;      // Diminui o intervalo de liberaçã
89.       printAjustarHoraLiberacao(); // Ajusta o intervalo no display
90.     }
91.   }
92. }
93. }
94. }
95. // Função chamada quando o botão de seleção é clicado
96. void onbtSelectClicked(EventButton& eb) {
97.   if(onMenu){
98.     switch (counter) {
99.       case 0:
100.         liberarRacao();          // Libera a ração
101.         break;
102.       case 1:
103.         printAjustarPeso();      // Ajusta o peso da porção
104.         break;
105.       case 2:
106.         printAjustarHoraLiberacao(); // Ajusta o intervalo de liberaçã
107.         break;
108.     }
109.   }else{
110.     if(counter != 2){
111.       updateMenu();
112.     } else {
113.       if(goodToStart){
114.         liberarRacao();
115.         milisLiberacao = interval * intervaloLiberacao;
116.         working = true;
117.         Serial.println("Iniciando liberacao de racao");
118.         Serial.println(milisLiberacao);
119.       } else {
120.         waitStartTimer(); // Inicia o timer para liberaçã
121.       }
122.     }
123.   }
124. }
125. void setup() {
126.   Serial.begin(9600);
127.   scale.begin(pinDT, pinSCK); // Configurando os pinos da balança
128.   scale.set_scale(439295.459f); // Enviando o valor da escala calibrado
129.   delay(2000);
130.   scale.tare(); // Zera a balança para desconsiderar a massa da estrutura
131.
132.   // Liga o LCD e sua backlight
133.   lcd.init();
134.   lcd.backlight();
135.
136.   // Setup dos botões de seleção

```

```

137.     btRight.setClickHandler(onbtRightClicked);
138.     btRight.setPressedHandler(onbtRightPressed);
139.     btLeft.setClickHandler(onbtLeftClicked);
140.     btLeft.setPressedHandler(onbtLeftPressed);
141.     btSelect.setClickHandler(onbtSelectClicked);
142.
143.     // Define a velocidade do motor de passo
144.     stepper.setSpeed(60);
145.
146.     // Atualiza o menu inicial
147.     updateMenu();
148. }
149.
150. void loop() {
151.     if(working){ // Se o timer começou
152.         currentMillis = millis();
153.         Serial.println(currentMillis - previousMillis);
154.         // Verifica se o tempo selecionado passou
155.         if (currentMillis - previousMillis >= milisLiberacao) {
156.             previousMillis = currentMillis; // Atualiza o tempo da última ação
157.             medirPeso();
158.             if(medidaMod <= 20){
159.                 liberarRacao();
160.             }
161.         }
162.     } else {
163.         previousMillis = millis();
164.         Serial.println(previousMillis);
165.     }
166.     btRight.update();
167.     btLeft.update();
168.     btSelect.update();
169. }
170.
171. // Função para medir o peso
172. void medirPeso() {
173.     scale.power_up(); // Liga o sensor da balança
174.     medida = scale.get_units(5); // Salva na variável o valor da média de 5 medidas
175.
176.     if (medida <= pesoMin) { // Confere se a massa está na faixa válida
177.         scale.tare(); // Zera a balança caso a massa seja menor que o valor mínimo
178.         medida = 0;
179.     } else if (medida >= pesoMax) {
180.         scale.tare(); // Zera a balança caso a massa seja maior que o valor máximo
181.         medida = 0;
182.     }
183.     medidaMod = medida * 1000; // Converte a medida para gramas
184.     scale.power_down(); // Desliga o sensor da balança
185. }
186.
187. // Função para liberar ração
188. void liberarRacao() {
189.     Serial.println("Liberação de ração iniciada");
190.     lcd.clear();
191.     lcd.setCursor(0, 1);

```

```

192.     lcd.print("Liberando...");
193.     medirPeso();
194.     while(medidaMod <= pesoPorcao) {
195.         stepper.step(-800); // Move o motor de passo para liberar ração
196.         medirPeso();
197.     }
198.     lcd.clear();
199.     lcd.print("Racao Liberada!");
200.     delay(2000);
201.
202.     updateMenu(); // Atualiza o menu após a liberação
203. }
204.
205. // Função para ajustar o peso da porção no display
206. void printAjustarPeso() {
207.     onMenu = false;
208.     lcd.clear();
209.     lcd.setCursor(1, 0);
210.     lcd.print("peso porcao:");
211.     lcd.setCursor(0, 1);
212.     lcd.print(pesoPorcao);
213.     lcd.setCursor(8, 1);
214.     lcd.print("gramas");
215. }
216.
217. // Função para ajustar o intervalo de liberação no display
218. void printAjustarHoraLiberacao() {
219.     onMenu = false;
220.     goodToStart = false;
221.     lcd.clear();
222.     lcd.setCursor(1, 0);
223.     lcd.print("ajustar intervalo:");
224.     lcd.setCursor(0, 1);
225.     lcd.print(intervaloLiberacao);
226.     lcd.setCursor(8, 1);
227.     lcd.print("Horas");
228. }
229.
230. // Função para iniciar o timer de liberação
231. void waitStartTimer() {
232.
233.     lcd.clear();
234.     lcd.setCursor(1, 0);
235.     lcd.print("pressione para");
236.     lcd.setCursor(0, 1);
237.     lcd.print("liberar...");
238.     goodToStart = true;
239. }
240.
241. // Função para atualizar o menu no display
242. void updateMenu() {
243.     onMenu = true;
244.     lcd.clear();
245.     lcd.setCursor(1, 0);
246.     switch (counter) {

```

```

247.         case 0:
248.             lcd.print("<Liberar Racao");
249.             break;
250.         case 1:
251.             lcd.print("Ajustar Peso");
252.             lcd.setCursor(0, 1);
253.             lcd.print(" de Porcao>");
254.             break;
255.         case 2:
256.             lcd.print("<Ajustar Hora");
257.             lcd.setCursor(0, 1);
258.             lcd.print(" de liberacao>");
259.             break;
260.     }
261. }

```

Apêndice B: Manual do (a) usuário (a)

Introdução

Bem-vindo ao seu sistema automático de alimentação para pets! Este equipamento é projetado para fornecer comida fresca e limpa de forma automática para seus cães e gatos, permitindo um controle preciso sobre a quantidade de ração e os intervalos de liberação.

Configuração Inicial

Conectar a Bateria

1. **Ligue à Tomada:** Conecte o carregador da bateria a uma tomada.

Montar o Equipamento

1. **Preencha o Reservatório:** Encha o reservatório de ração com 1 kg de ração.
2. **Prepare o Pote:** Lave e coloque o pote no compartimento da base do equipamento.

Utilização do Equipamento

Navegação pelo Menu

O sistema possui três botões para navegação e seleção:

- **Botão Direito:** Move a seleção para a direita.
- **Botão Esquerdo:** Move a seleção para a esquerda.
- **Botão do Meio:** Seleciona a opção atual.

Opções de Configuração

1. **Liberação Imediata:**
 - Navegue até a opção de liberação imediata.
 - Pressione o botão do meio para liberar uma porção de ração imediatamente.
2. **Ajuste do Peso da Ração:**

- *Navegue até a opção de ajuste do peso.*
- *Use os botões direito e esquerdo para aumentar ou diminuir a quantidade de ração desejada.*
- *Pressione o botão do meio para confirmar.*

3. Ciclo de Liberação:

- *Navegue até a opção de ciclo de liberação.*
- *Use os botões direito e esquerdo para ajustar o horário e a frequência das liberações.*
- *Pressione o botão do meio para iniciar o ciclo.*

Visualização e Controle

O display LCD exibirá as opções de menu e os resultados das seleções. A interface amigável permite ajustar facilmente as configurações e verificar o status do sistema.

Manutenção

Limpeza do Equipamento

1. ***Desconecte a Bateria:*** *Sempre desconecte a bateria antes de realizar qualquer limpeza.*
2. ***Lave os Recipientes:*** *Remova os recipientes de comida e lave-os com água e sabão neutro.*
3. ***Limpe o Exterior:*** *Use um pano úmido para limpar o exterior do equipamento.*