

INSTITUTO FEDERAL DO RIO GRANDE DO SUL - IFRS  
CAMPUS RESTINGA

Michael Rodrigues

# **Projeto e Implementação de um Sistema de Monitoramento de Refeições para Cães e Gatos**

**Porto Alegre**

**2022**

Michael Rodrigues

## **Projeto e Implementação de um Sistema de Monitoramento de Refeições para Cães e Gatos**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Eletrônica Industrial no Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS) - *Campus* Restinga, como requisito parcial para obtenção de título de Tecnólogo em Eletrônica Industrial.

Prof. Dr. Eng. Alexsandro Cristovão Bonatto  
Profa. Dra. Sheila Santisi Travessa

Porto Alegre

2022

Michael Rodrigues

## **Projeto e Implementação de um Sistema de Monitoramento de Refeições para Cães e Gatos**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Eletrônica Industrial no Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS) - *Campus Restinga*, como requisito parcial para obtenção de título de Tecnólogo em Eletrônica Industrial.

Trabalho aprovado. Porto Alegre, 16 de Agosto de 2022:

---

**Prof. Dr. Eng. Alexsandro Cristovão Bonatto**  
Orientador

---

**Profa. Dra. Sheila Santisi Travessa**  
Coorientadora

---

**Profa. Dra. Elizandra Martinazzi**  
Avaliadora - IFRS *Campus Restinga*

---

**Prof. Me. João Roberto Gabbardo**  
Avaliador - IFRS *Campus Restinga*

---

**Profa. Dra. Marline Ilha da Silva**  
Avaliadora - IFRS *Campus Restinga*

Porto Alegre

2022

# AGRADECIMENTOS

Agradeço ao Criador do Universo pela oportunidade extraordinária dos anos de muito estudo, esforço e empenho que foram essenciais para a concretização de mais este sonho.

À minha família, especialmente minha mãe-pai, Rita e minha avó Leonida por todo suporte psicológico e apoio financeiro.

Ao Prof. Eng. Robson Ortiz Rodrigues que fez história na excelência de transmitir seu conhecimento. Ensinando de maneira prática a humildade de redesenhar o circuito em estudo. Aulas de solda e prototipagem de uma PCB.

A Profa. Me. Alessandra Felizari pois foi a primeira pessoa que acreditou e investiu seu tempo e sua dedicação nesse projeto.

Minha gratidão ao Prof. Me. Luis Felipe Kiesow de Macedo por ter apresentado um novo tipo de conceito matemático para escrita de artigos, o  $\text{\LaTeX}$ .

Ao Prof. Dr. Alexsandro Cristóvão Bonatto e a Profa. Dra. Sheila Santisi Travessa que tornaram esse trabalho possível. Especialmente por sua dedicação, que por muitas vezes, deixaram de lado seus momentos de descanso e lazer para ajudar e orientar o projeto.

Aos irmãos que o IFRS me deu: Tomas Thomaz e Dhamens Almansa. Aos tempos como bolsista de monitoria de Cálculo e Eletrônica e ao incentivo para continuar no caminho do conhecimento apesar das dificuldades cotidianas.

*"A verdadeira grandeza é começar onde você está, usar o que você tem, e fazer o que você pode."*

*Arthur Robert Ashe Jr  
(10/07/1943 - 06/02/1993)*

# RESUMO

Segundo Instituto PET Brasil (IPB), a alimentação de cães e gatos domésticos, ao longo do dia, não deve ser feita em grandes quantidades servidas uma única vez, pois isso seria prejudicial a saúde do animal. É indicado fragmentar a alimentação total ao longo do dia. Um alimentador automático de ração permite melhorar o controle de qualidade de vida desses animais. Um equipamento automatizado permite fazer o monitoramento constante da qualidade nutricional das refeições do seu animal de estimação. O usuário que ligar o alimentador através de um botão tátil deve informar ao sistema que é microncontrolado por um Atmega328P, os principais parâmetros da ração que será depositada no reservatório. A partir dessa etapa é calculado a quantidade total do alimento e com este resultado o usuário escolhe fragmentar a ração. A divisão do alimento é determinada por alarmes, esses alarmes farão a liberação da comida no tempo programado. A tampa do reservatório possui um sensor ultrassônico HC-SR04 para quando houver menos de cinquenta por cento da ração, é emitido um aviso. O efeito sonoro emitido pelo alto falante será o aviso necessário para acionar o motor com redução que liberará a ração. O acionamento é feito temporariamente até que a quantidade identificada seja depositada no recipiente do animal até que chegue na quantidade correta da porção através de uma célula de carga. Para a correta sincronização de dia, hora e minuto do projeto foi utilizado um módulo de Relógio em Tempo Real, o DS3231. A comunicação do alimentador com o tutor ocorre através de um mostrador digital e um controle remoto, onde o sinal obtido é coletado através de um receptor infravermelho (KY-022). Os materiais utilizados nesse projeto estão intimamente ligados a proposta de um equipamento de baixo custo. O teste final foi realizado através de um relógio digital programável, onde é possível acionar o motor em qualquer momento do dia, verificando periodicamente a quantidade de ração no recipiente. O aviso sonoro tornou-se útil para que haja a percepção da liberação de comida ao animal. O controle do alimentador foi aplicado a um controle remoto genérico, mas a partir do programa criado é possível implementar qualquer tipo de controle no alimentador.

**Palavras-chave:** alimentador automático, cães, gatos, monitoramento, controle, baixo custo.

# ABSTRACT

According to Instituto PET Brasil (IPB), feeding domestic dogs and cats throughout the day should not be done in large quantities served only once, as this would be harmful to the health of the animal. It is indicated to fragment the total food throughout the day. An automatic feed feeder allows to improve the quality of life control of these animals. Automated equipment allows you to constantly monitor the nutritional quality of your pet's meals. The user who turns on the feeder through a tactile button must inform the system that it is microncontrolled by an Atmega328P, the main parameters of the feed that will be deposited in the reservoir. From this step, the total amount of food is calculated and with this result the user chooses to fragment the ration. Food division is determined by alarms, these alarms will release food at the programmed time. The reservoir lid has an HC-SR04 ultrasonic sensor so when less than fifty percent of the feed is left, a warning is issued. The sound effect emitted by the speaker will be the necessary warning to start the engine with reduction that will release the feed. The activation is done temporarily until the identified amount is deposited in the animal's container until the correct amount of the portion arrives through a load cell. For the correct synchronization of the day, hour and minute of the project, a Real Time Clock module, the DS3231, was used. The communication between the feeder and the tutor occurs through a digital display and a remote control, where the signal obtained is collected through an infrared receiver (KY-022). The materials used in this project are closely linked to the proposal of a low cost equipment. The final test was carried out using a programmable digital clock, where it is possible to start the engine at any time of the day, periodically checking the amount of feed in the container. The audible warning became useful for the perception of the release of food to the animal. The feeder control was applied to a generic remote control, but from the created program it is possible to implement any type of control on the feeder.

**Keywords:** automatic feeder, dogs, cats, monitoring, control, low cost.

# LISTA DE ILUSTRAÇÕES

Figura 1 – Representação das quantidades dos animais domésticos no Brasil. . . .	15
Figura 2 – Diagrama de blocos do funcionamento do sistema automático de alimentação. . . . .	18
Figura 3 – Lola, 5 anos. . . . .	26
Figura 4 – Alimentador Automático Pet Dogis DA020BR. . . . .	28
Figura 5 – Comedouro PET Automatizado USB Robotizado Newpet. . . . .	29
Figura 6 – Comedouro Automático Perfect Pet Dinner. . . . .	30
Figura 7 – Impressoras 3D FlashForge Finder. . . . .	32
Figura 8 – Destaques da Impressora 3D FlashForge Finder. . . . .	33
Figura 9 – Desenho do reservatório e a tampa no FlashPrint. . . . .	33
Figura 10 – Resultado da impressão do reservatório e da tampa. . . . .	34
Figura 11 – Desenho do eixo helicoidal no FlashPrint. . . . .	34
Figura 12 – Compartimento para liberação da ração. . . . .	34
Figura 13 – Resultado da impressão do eixo helicoidal. . . . .	35
Figura 14 – ATmega328 no encapsulamento DIP. . . . .	36
Figura 15 – Conectores da placa Arduino UNO. . . . .	37
Figura 16 – Diagrama elétrico da placa Michaelino versão 1.0. . . . .	38
Figura 17 – Mostrador digital de vinte caracteres por quatro linhas. . . . .	39
Figura 18 – Seleção inicial do sistema do alimentador. . . . .	39
Figura 19 – Informação do PC do animal no sistema do alimentador. . . . .	40
Figura 20 – Faixa etária no sistema do alimentador. . . . .	40
Figura 21 – Teste gestacional do animal no sistema do alimentador. . . . .	40
Figura 22 – Resposta positiva para o teste gestacional no sistema do alimentador. . . . .	41
Figura 23 – Teste de lactação do animal no sistema do alimentador. . . . .	41
Figura 24 – Informação de número de filhotes no sistema do alimentador. . . . .	41
Figura 25 – Primeira parte dos parâmetros para calcular o valor total de gordura do alimento. . . . .	42
Figura 26 – Segunda parte dos parâmetros para calcular o valor total de gordura do alimento. . . . .	42
Figura 27 – O sistema do alimentador informa ao usuário o valor total de ração em gramas. . . . .	42
Figura 28 – O sistema do alimentador permite ao usuário o número exato que deseja fragmentar a comida do animal. . . . .	43
Figura 29 – Esquemático elétrico da placa Michaelino com o mostrador digital. . . . .	43
Figura 30 – Controle Remoto (KEYS). . . . .	44

Figura 31 – Receptor infravermelho KY-022. . . . .	44
Figura 32 – Circuito da interface do projeto. . . . .	45
Figura 33 – Célula de carga no acessório do recipiente do animal. . . . .	46
Figura 34 – Ponte de Wheatstone. . . . .	47
Figura 35 – Conversor analógico-digital HX711. . . . .	48
Figura 36 – Módulo RTC DS3231. . . . .	48
Figura 37 – Esquemático elétrico da placa Michaelino e do RTC DS3231. . . . .	50
Figura 38 – Partes essenciais de um motor em corrente contínua . . . . .	50
Figura 39 – Modelagem simples de um motor. . . . .	51
Figura 40 – Circuito para controle do motor. . . . .	51
Figura 41 – Esquemático elétrico da placa Michaelino com o motor 12 V. . . . .	52
Figura 42 – Sensor Ultrassônico HC-SR04. . . . .	53
Figura 43 – Esquemático elétrico da placa Michaelino com sensor ultrassônico. . . . .	54
Figura 44 – Destaque das funções do Arduino IDE 1.8.19. . . . .	55
Figura 45 – Opção para adicionar as bibliotecas no programa no Arduino. . . . .	55
Figura 46 – Mapeamento das teclas do controle remoto KEYES no Arduino IDE. . . . .	56
Figura 47 – Comunicação serial das teclas do controle remoto KEYES no Arduino IDE. . . . .	56
Figura 48 – Acesso aos menus para obter dados em tempo real pelo Excel. . . . .	57
Figura 49 – Fluxograma da interface entre o usuário e o alimentador automático . . . . .	58
Figura 50 – Demonstração do valor esperado e medido do sensor ultrassônico. . . . .	59
Figura 51 – Resultado final do recipiente do alimentador . . . . .	61
Figura 52 – Mostrador digital soldado ao módulo I2C . . . . .	62
Figura 53 – Esquemático da placa Michaelino. . . . .	62
Figura 54 – PCB Michaelino. . . . .	63
Figura 55 – Visualização 3D da placa Michaelino. . . . .	63

# LISTA DE TABELAS

Tabela 1 – Fases da vida de um cachorro. . . . .	20
Tabela 2 – Fases da vida de um gato. . . . .	20
Tabela 3 – Interpretação do rótulo da ração de cães e gatos. . . . .	21
Tabela 4 – Requisitos energéticos de cães em manutenção obtidos através do Peso Corpóreo. . . . .	23
Tabela 5 – Requisitos energéticos de gatos em manutenção . . . . .	23
Tabela 6 – Necessidades diárias de energia metabolizável para gatas lactantes. . .	25
Tabela 7 – Fatores Originais de Atwater. . . . .	25
Tabela 8 – Fatores de Atwater modificado. . . . .	26
Tabela 9 – Informações nutricionais da ração de 18 kg. . . . .	27
Tabela 10 – Dados Técnicos do Alimentador Automático Pet Dogis DA020BR. . . .	29
Tabela 11 – Dados Técnicos do Comedouro PET Automatizado USB Robotizado Newpet. . . . .	30
Tabela 12 – Dados Técnicos do Alimentador Digital Eletrônico Programável Lorben. .	31
Tabela 13 – Comparação dos alimentadores automáticos comerciais. . . . .	31
Tabela 14 – Descrição dos Pinos do ATmega328P. . . . .	36
Tabela 15 – Pinos de alimentação do Arduino UNO. . . . .	37
Tabela 16 – Comparativo das características periféricas das placas Arduino e Mi-chaellino. . . . .	38
Tabela 17 – Pinos do mostrador digital. . . . .	39
Tabela 18 – Características elementares do receptor KY-022. . . . .	45
Tabela 19 – Características elementares da célula de carga de 50 kg. . . . .	47
Tabela 20 – Características elementares do conversor analógico-digital de 24 bits HX711. . . . .	47
Tabela 21 – Características elementares do módulo RTC DS3231. . . . .	49
Tabela 22 – Características do motor em corrente contínua AK555/306PL13S500C .	52
Tabela 23 – Características elementares sensor ultrassônico HC-SR04. . . . .	53
Tabela 24 – Tabela dos pontos indicados no gráfico de 20 à 200 mm. . . . .	60
Tabela 25 – Tabela dos pontos indicados no gráfico de 200 à 300 mm . . . . .	61
Tabela 26 – Lista de componentes do projeto: 23/05/2022. . . . .	89

# LISTA DE ABREVIATURAS E SIGLAS

$C_a$	Cálcio
EB	Energia Bruta
ED	Energia Digestível
EE	Extrato Etéreo
EM	Energia Metabolizável
EMRBQ	Energia Metabolizável para uma Ração de Baixa Qualidade
ENN	Extrato Não-Nitrogenado
EEPROM	Electrically Erasable Programmable Read-Only Memory
DIP	Dual In-line Package
gf.cm	Grama-força/centímetro
IDE	Ambiente de Desenvolvimento Integrado
IR	Infrared ou Infravermelho
I2C	Inter-Integrated Circuit
LCD	Liquid Crystal Display
Kgf.cm	Quilograma-força/centímetro
MF	Matéria Fibrosa
MM	Matéria Mineral
NECM	Necessidade Energética de Cães em Manutenção
NEM	Necessidade Energética para Manutenção
NRC	National Research Council
P	Fósforo
PB	Proteína Bruta
PC	Peso Corpóreo

PCB	Printed Circuit Board
PM	Peso Metabólico
PWM	Pulse Width Modulation
QdA	Quantidade de Alimento
QFP	Quad Flat Package
RISC	Reduced Instruction Set Computer
RPM	Rotação por minuto
RTC	Real Time Clock
SDA	Serial Data
SCL	Serial Clock
SI	Sistema Internacional de Unidades
SPI	Serial Peripheral Interface
SRAM	Static random-access memory
TWI	Two-Wire Interface
UART	Universal Asynchronous Receiver-Transmitter
UM	Umidade Máxima
USB	Universal Serial Bus
W	Watt

# LISTA DE SÍMBOLOS

$\Omega$	Letra grega Ômega
$\mu$	Micro, que possui um fator de $10^{-6}$
$k$	Quilo, que possui um fator de $10^3$
$F$	Unidade de capacitância Farad
$p$	Pico, que possui um fator de $10^{-12}$
$kg$	Unidade de medida para a grandeza massa
$l$	Unidade de medida de volume
$q$	$\frac{\text{Peso corporal no momento da avaliação (kg)}}{\text{Peso corporal esperado quando adulto (kg)}}$
$L$	Fator de correlação para o estágio de lactação: semana 1 = 0,75; semana 2 = 0,95; semana 3 = 1,1; semana 4 = 1,2
$n$	Número de filhotes entre 1 e 4
$m$	Número de filhotes entre 5 e 8 (<5 filhotes $m = 0$ )

# SUMÁRIO

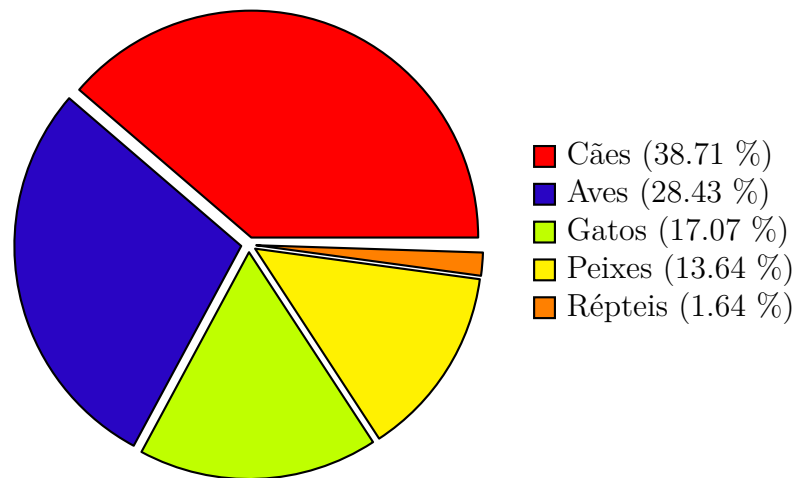
<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
<b>1.1</b>	<b>Justificativa</b>	<b>16</b>
<b>1.2</b>	<b>Objetivos</b>	<b>16</b>
1.2.1	Objetivo Geral	16
1.2.2	Objetivos Específicos	17
<b>1.3</b>	<b>Especificação do equipamento</b>	<b>17</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>19</b>
<b>2.1</b>	<b>Alteridades entre cães e gatos</b>	<b>19</b>
2.1.1	Alimentação comercial	20
<b>2.2</b>	<b>Requisitos energéticos</b>	<b>22</b>
<b>2.3</b>	<b>Energia dos alimentos</b>	<b>24</b>
<b>2.4</b>	<b>Cálculo da porção alimentar</b>	<b>26</b>
2.4.1	Estudo de caso	26
<b>3</b>	<b>METODOLOGIA</b>	<b>28</b>
<b>3.1</b>	<b>Modelos Comerciais</b>	<b>28</b>
3.1.1	Alimentador Automático Pet Dogis DA020BR	28
3.1.2	Comedouro PET Automatizado USB 35 litros Robotizado	29
3.1.3	Comedouro Automático Perfect Pet Dinner	30
3.1.4	Comparativo entre Equipamentos Comerciais	31
<b>4</b>	<b>DESENVOLVIMENTO DO PROJETO</b>	<b>32</b>
<b>4.1</b>	<b>Desenvolvimento Mecânico</b>	<b>32</b>
4.1.1	Reservatório	33
4.1.2	Eixo helicoidal	34
<b>4.2</b>	<b>Desenvolvimento do Hardware</b>	<b>35</b>
4.2.1	Processamento de Dados	35
4.2.2	Interface com o Usuário	39
4.2.2.1	Mostrador Digital	39
4.2.2.2	Receptor Infravermelho e Controle Remoto	44
4.2.3	Monitoramento	45
4.2.3.1	Controle da Porção Alimentar	46
4.2.3.2	Controle de Horários	48
4.2.3.3	Controle do Motor	50
4.2.3.4	Controle do Reservatório	53

4.3	Desenvolvimento do Software . . . . .	54
5	ANÁLISE DE RESULTADOS . . . . .	59
6	CONCLUSÕES . . . . .	64
6.1	Melhorias Futuras . . . . .	64
	REFERÊNCIAS . . . . .	65
A	CÓDIGO DO ARQUIVO RELOGIO.INO. . . . .	67
B	CÓDIGO DO ARQUIVO LIQUIDCRYSTALI2C.H. . . . .	79
C	CÓDIGO DO ARQUIVO WIRE.H. . . . .	83
D	MAPEAMENTO DAS TECLAS DO CONTROLE REMOTO KEYES. . . . .	85
E	CONFIGURAÇÃO DO ARQUIVO CONTROLE.C. . . . .	86
F	LISTA DE COMPONENTES DO PROJETO . . . . .	88

# 1 INTRODUÇÃO

O vínculo entre o homem e os animais de estimação consolidou-se nos últimos anos. Em 2019, a ABINPET (Associação Brasileira de Produtos para Animais de Estimação) contabilizou cerca de 140 milhões de animais domésticos no país. Na Figura 1 é representada a distribuição das espécies de animais domésticos que os brasileiros possuem.

Figura 1 – Representação das quantidades dos animais domésticos no Brasil.



Fonte: Adaptado de ABINPET (2019).

Quando a criação, responsabilidades e a alimentação passaram a ser essencialmente controlados por seres humanos, os cães e gatos realizaram inúmeras funcionalidades como proveniência de alimento aproximado ao homem nômade, defesa contra outros animais, rastreamento de pragas, companhia e, hoje, efetivamente são verdadeiros membros da família (OGOSHI *et al.*, 2015).

Os cães e os gatos são membros da ordem Carnívora, o que indica classes que aprimoram-se na rotina alimentar carnívora e por esse motivo exibem anatomia peculiar. Entretanto pertencem a diferentes âmbitos da ordem e, como efeito, têm ganho diferentes legados de prioridades alimentares e maneiras de escolhas de alimentos (OGOSHI *et al.*, 2015).

Ao mesmo tempo que a evolução do cão aconselha uma dieta mais onívora, a evolução do gato aponta que este gênero utilizava uma dieta rica em carne durante sua expansão evolutiva (OGOSHI *et al.*, 2015).

A estadia do gato com um regime extremamente qualificado sucedeu em adequações metabólicas que apresentam-se como particularidades nos requisitos nutricionais. Por isso, os alimentos acessíveis no mercado mostram características estabelecidas pelas

diferenças nas imposições nutricionais e na rotina alimentar de cada uma destas espécies. A assimilação dessas diferenças é de interesse prático, uma ocasião que alguns protetores podem desacertadamente crer que gatos podem ser alimentados da mesma forma que cães (OGOSHI et al., 2015).

O progresso nas pesquisas sobre nutrição nos últimos anos provocou melhor entendimento sobre as prioridades nutricionais destes animais, e assim sendo houve uma grandiosa evolução na alimentação deles. Os alimentos, além de nutrir devem promover a saúde, o bem estar e a longevidade. Mas com a abundante quantidade de alimentos comerciais prontos para o aproveitamento, com definições cada vez mais aprimorados, possibilitam que os tutores estejam inclinados a cometerem outro equívoco no manuseio alimentar é a superalimentação. Simultaneamente a maioria dos animais domésticos padecem com a humanização, vivendo em espaços pequenos resultando no sedentarismo. Então, pode-se conceituar que o problema nutricional é o consumo de energia maior do que a demanda, o que pode ser confirmado pelos altos índices de sobrepeso e obesidade nessas duas espécies: cães e gatos (COUNCIL et al., 2006).

## 1.1 JUSTIFICATIVA

No Brasil foi identificado que entre 25 a 40 % da população canina e felina que recebe atendimento veterinário encontra-se acima do peso ou é obesa (CARVALHO, 2014).

Há uma concordância geral entre os veterinários sobre a divisão igualitária das porções alimentares. Deste modo, como tutores devemos proporcionar de uma dieta equilibrada. Com o manuseio de um alimentador automático programável é possível realizar essa tarefa, respeitando cada fase do desenvolvimento do cão e do gato (CARCIOFI; JEREMIAS, 2010).

## 1.2 OBJETIVOS

Esta seção será descrito o objetivo geral e a projeção dos objetivos específicos que esse trabalho pretende alcançar.

### 1.2.1 OBJETIVO GERAL

O objetivo geral do projeto é desenvolver um alimentador automático de cães e gatos de baixo custo, utilizando um botão, um microcontrolador ATmega328P, um sensor ultrassônico do tipo HC-SR04, uma célula de carga, um RTC DS3231, um LCD de vinte caracteres por quatro linhas, um módulo receptor infravermelho do tipo KY-022 e um controle remoto KEYES.

## 1.2.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos do projeto são:

- Construir um comedouro resistente que possa comportar 1 kg de ração;
- Desenvolver uma PCB de microcontrole do alimentador automático;
- Efetuar um fluxograma que realize o cálculo correto da quantidade total de ração para cada fase do cão/gato;
- Fabricar um eixo helicoidal para fazer o transporte da ração até a vasilha do cão/gato;
- Realizar a fragmentação da ração do horário pré-determinado.
- Utilizar um controle remoto, onde realize a configuração correta para liberar a quantidade correta de ração.

## 1.3 ESPECIFICAÇÃO DO EQUIPAMENTO

O equipamento deste projeto fundamenta-se em proporcionar uma fragmentação alimentar diária para cada fase do cão ou gato através de um sistema eletrônico de controle de intervalos de tempo e de quantidade de ração dispensada.

O botão tátil serve para ligar e desligar o sistema.

Através do relógio de tempo real, o painel de visualização mostra a data e hora da próxima refeição através do display de cristal líquido.

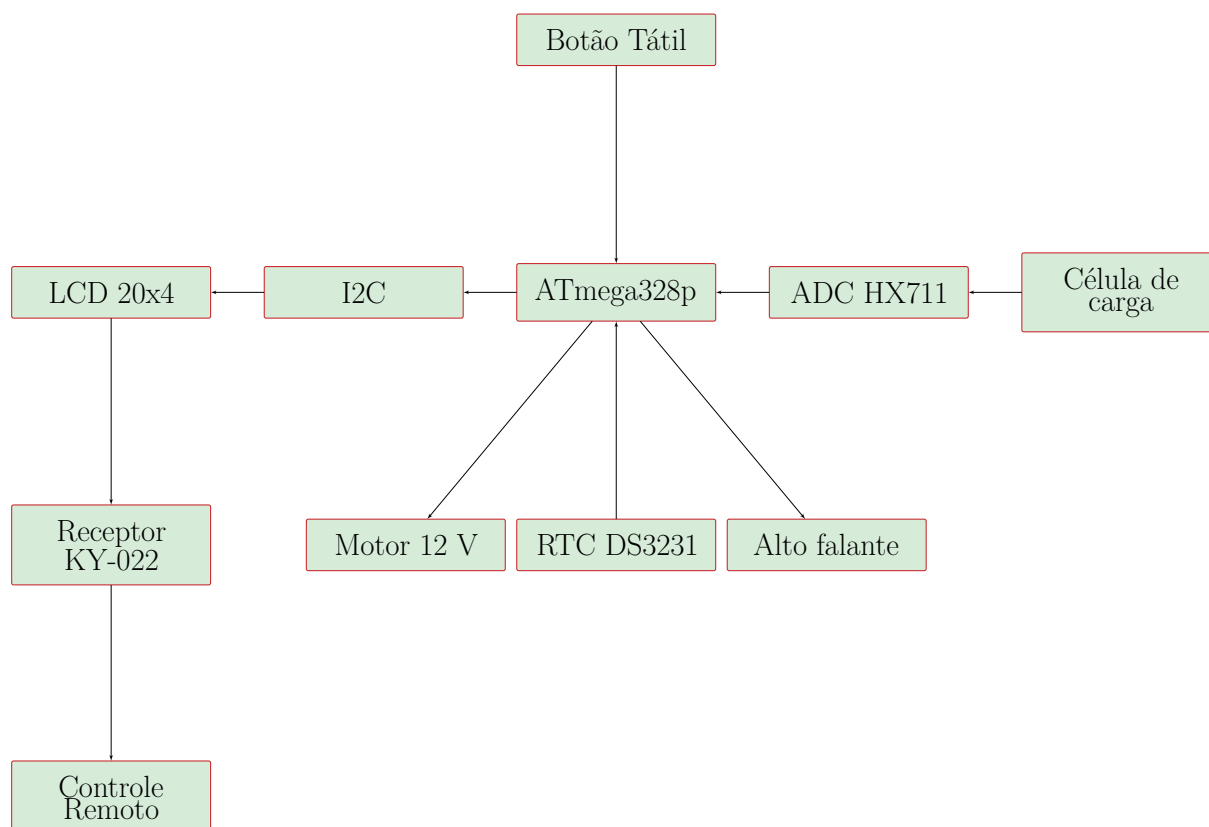
O receptor infravermelho é utilizado no momento de vistoria dos dados. Ou seja a quantidade da ração que deve ser dividida será feita através de um controle.

O recipiente armazenará a quantidade diária total do alimento para o cão ou gato. Quando chegar na hora certa, o motor de passo é acionado e a movimentação do eixo helicoidal permite a liberação da ração.

A quantidade de ração será medida através de uma célula de carga que estará debaixo do recipiente do animal. Dessa forma, a ração será liberada até que esta quantidade seja atingida.

A Figura 2 apresenta um diagrama de blocos com os principais componentes do sistema alimentador de cães e gatos. Fazem parte deste sistema: um botão tátil que ligará o sistema, um microcontrolador ATmega328P, um módulo RTC DS3231 para a determinação da data, um módulo I2C Interface Serial para ser conectado no único mostrador digital LCD 20x4, um Receptor Infravermelho KY-022 com um controle remoto de 17 teclas que será responsável por acionar o motor de 12 V e o alto falante.

Figura 2 – Diagrama de blocos do funcionamento do sistema automático de alimentação.



Fonte: O próprio autor (2022).

## 2 FUNDAMENTAÇÃO TEÓRICA

O conteúdo deste capítulo busca apresentar a revisão bibliográfica a cerca dos principais tópicos que estão incluídos no tema desse projeto.

### 2.1 ALTERIDADES ENTRE CÃES E GATOS

Cães e gatos pertencem a classe Mammalia, são mamíferos caracterizados por apresentarem: glândulas mamárias, corpo totalmente ou parcialmente encoberto por pelos, dentes diferenciados com incisivos, caninos, pré-molares e molares e diafragma que é uma membrana muscular que separa o tórax do abdome e facilita na ventilação dos pulmões. Além de serem membros da ordem Carnívora, o que os diferencia são as superfamílias que pertencem: Canoidea e Feloidea (THOMAS, 2010).

A superfamília Canoidea possui famílias com costumes alimentares variados:

- Ursidae (ursos) e Procyonidae (guaxinins): são animais onívoros que apresentam uma alimentação que cercam plantas, algas e até animais;
- Aluridae (panda-vermelho): é um tipo de animal herbívoro estrito que alimenta-se basicamente de bambu e é complementado com bagas, frutos, cogumelos, raízes e etc;
- Mustelidae (lontras, doninhas, furões) e Canidae (cães) são animais carnívoros. As lontras alimentam-se de peixes, crustáceos e moluscos enquanto os cães para manter uma boa saúde deve ser nutrido com proteínas, cálcio, lipídios, carboidratos, minerais e vitaminas.

A Tabela 1 mostra as fases do cachorro em função do seu estágio de vida e idade. É importante destacar que o processo de alimentação inicia no período neonatal, onde o cão depende totalmente da mãe para ser alimentar e fazer suas necessidades fisiológicas. O manejo alimentar só muda quando entra no período da infância, por volta da nona semana de vida. O tutor deve fazer a transição do leite materno para a ração apropriada para sua idade (CHAVES, 2011).

A superfamília Feloidea possui três famílias:

- Viverridae (urso-gato-asiático): são animais carnívoros estritos, ou seja, alimentam-se de carnes e de vegetais, come folhas, frutas e roedores;

Tabela 1 – Fases da vida de um cachorro.

Estágio da vida	Idade do cão
Neonatal	0 a 12 dias de vida
Transição	13 a 20 dias de vida
Infância	3 a 12 semanas de vida
Juvenil	de 12 semanas até a puberdade
Adulta	de 1 a 7 anos
Velhice	a partir de 7 anos

Fonte: Adaptado de Buddy toys & care (2022).

- Hyaenidae (hiena): são animais carnívoros estritos, composta por cadáveres de animais e restos de comida. São capazes de caçar filhotes de fazelas, de zebras e gnus;
- Felidae (gatos): são animais carnívoros estritos, quando estão aptos a caça alimentam-se de insetos, pequenas aves e roedores.

A Tabela 2 apresenta as fase da vida do gato em função da idade e estágio de vida. Podemos identificar também a primeira fase da vida: filhote. O desmame ocorre entre a sétima e oitava semana de vida. Após é necessário que o tutor inicie a alimentação com ração própria para filhotes (BARROS et al., 2019).

Tabela 2 – Fases da vida de um gato.

Estágio da vida	Idade do gato
Filhote	0 a 6 meses
Junior	7 meses a 2 anos
Jovem Adulto	3 a 6 anos
Adulto	7 a 10 anos
Idoso	11 a 14 anos
Geriátrico	A partir dos 15 anos

Fonte: Adaptado de Clínica Veterinária Gatos & Gatos (2021).

A história evolutiva do cão demonstra uma dieta mais onívora na natureza, já o gato mostra que está espécie consumia uma dieta a base de carne através de seu progresso evolutivo (SILVA, 2011).

### 2.1.1 ALIMENTAÇÃO COMERCIAL

O setor de bichos de estimação é o segmento do agronegócio relacionado com o desenvolvimento das atividades de criação, produção e comercialização de cães, gatos, aves canoras e ornamentais, peixes ornamentais, pequenos mamíferos e répteis (ABINPET, 2019).

No mercado há uma diversidade imensa de ração para animais de estimação. São centenas de marcas que oferecem três categorias como:

- a) Econômica: usa produtos de baixa qualidade no processo de fabricação, mas são as mais baratas. Como regra geral, contém farelos de ossos, soja, penas e bicos triturados, trigo e milho, o que torna a digestão e rendimento dos nutrientes mais trabalhoso, o que faz com que o cão e o gato tenha que comer mais para se sentir saciado. O aproveitamento é de 77 %.
- b) Premium: é balanceada, rica em proteínas e possui aproveitamento de 87 %, ou seja, o animal se satisfaz com menos quantidade.
- c) Super Premium: possui proteína animal, o que facilita a absorção de nutrientes e a digestibilidade, que pode ultrapassar a taxa de 92 %. Como o bicho se sente saciado com pouca ração, o volume de fezes também diminui.

A Tabela 3 evidencia os principais nutrientes presentes na ração de cães e gatos. É comum que esses dados estejam expressos em porcentagem ou grama por quilograma.

Tabela 3 – Interpretação do rótulo da ração de cães e gatos.

Componentes Nutricionais	Descrição
Proteína Bruta (PB)	Mostra a qualidade da proteína depende dos ingredientes utilizados na formulação, já que alguns têm maior digestibilidade, ou seja, são melhor aproveitados pelo organismo do animal
Extrato Etéreo (EE)	Indica a quantidade mínima de gordura presente na ração.
Extrativo Não Nitrogenado (ENN)	Indica os carboidratos da dieta, exceto a fibra.
Umidade (UM)	Registra a quantidade máxima de umidade (água) que pode estar presente na comida.
Matéria Fibrosa (MF)	Designa a quantidade máxima de fibra da ração.
Matéria Mineral (MM)	Mostra a quantidade máxima de minerais presentes.
Cálcio ( $C_a$ )	Sinaliza a quantidade de cálcio no alimento.
Fósforo (P)	Revela a quantidade mínima que deve estar presente no alimento.

Fonte: Adaptado de BitCão (2021).

Para suprir as necessidades nutricionais de cães e gatos, os fabricantes dividem as rações da seguinte forma: filhote, adulto, sênior e em tamanho do animal. Filhotes necessitam de mais proteínas e nutrientes que os adultos, mas na velhice as exigências energéticas diminuem. Os animais pequenos possuem a relação entre o tamanho do corpo com a massa corporal mais elevada do que a dos animais grandes, ou seja, têm metabolismo

mais rápido e consomem mais energia. Os animais grandes exigem medidas controladas de cálcio, fósforo e minerais para manter a estrutura óssea e garantir o bom desenvolvimento das articulações (CARCIOFI; JEREMIAS, 2010).

## 2.2 REQUISITOS ENERGÉTICOS

A nutrição é a análise dos alimentos, os seus nutrientes e outros elementos, acrescentando o desempenho dos nutrientes específicos, as suas interações com o outro, e seu equilíbrio dentro de uma dieta. Os seis grupos de nutrientes são: água, carboidratos, proteínas, gorduras, sais minerais e vitaminas, os quais têm finalidades específicas que colaboram para o crescimento, preservação dos tecidos do corpo e saúde. Quando oxidados pelo metabolismo o corpo produz energia (CASE et al., 2010).

A energia dietética condiz com o fundamental regulador de consumo voluntário de animais. Deste modo, a totalidade dos nutrientes se mostram balanceados de acordo com a densidade energética do alimento. Para cães e gatos, tanto os requisitos energéticos como valor energético dos alimentos, se expõe em Energia Metabolizável (EM) uma vez que em carnívoros as perdas energéticas pela urina são essenciais (OGOSHI et al., 2015).

As carências energéticas de cães e gatos, como em outras espécies animais, se estimam de acordo com o Peso Metabólico (PM). O PM associa o Peso Corpóreo (PC) ao desenvolvimento alométrico do bicho, uma vez que a perda de calor é equilibrado a superfície corporéa. Por isso, as necessidades energéticas de um animal precisam mais do seu PM do que do seu PC (CARCIOFI, 2007).

Para a determinação do PM em cães, o NRC (National Research Council) indica operar o PC com o expoente 0,75 por empregar variações nas proporções dos órgãos em cães. Por causa das inúmeras raças caninas presentes, ostentam uma variedade de PC entre 1,0 kg a 100,0 kg e padrões de crescimentos diferentes, à vista disso, requisitos nutricionais e manipulações alimentares distintas. Já para os felinos não obesos emprega-se o expoente 0,67, dado que a espécie não expõe grande variação entre as raças, todavia, para animais obesos o expoente é 0,4 (OGOSHI et al., 2015).

A Necessidade Energética para Manutenção (NEM) é a energia vital para resistir o equilíbrio energético (onde a EM é igual à geração de calor), acima de um longo intervalo de tempo. Dessa forma, a NEM pode alterar com qualquer condição que influe na fabricação de calor. Isso inclui a energia manifestada para termo regulação, atividade espontânea e exercício equilibrado. Assim, um animal adulto em manutenção combina àquele que não está na fase de gestação ou lactação e não realiza atividades acentuadas para todas as fases do cão existe uma equação específica para cálculo do NEM representada na Tabela 4.

A Tabela 5 mostra os requisitos energéticos para gatos em manutenção. É importante

Tabela 4 – Requisitos energéticos de cães em manutenção obtidos através do Peso Corpóreo.

Tipos de cães	Requisitos energéticos (kcal por dia)
Cães adultos jovens e ativos	$140 \text{ kcal} \times PC^{0,75}$
Cães ativos	$130 \text{ kcal} \times PC^{0,75}$
Cães idosos	$105 \text{ kcal} \times PC^{0,75}$
Cães sedentários	$95 \text{ kcal} \times PC^{0,75}$

Fonte: Adaptado de National Research Council (2006).

observar que o PC para gatos em boa saúde corresponde ao escore de condição corporal  $\leq 5$ , em escala de 1 a 9. Já o escore de condição corporal para um gato em obesidade significa que o escore de condição corporal  $> 5$ , em escala de 1 a 9 (OGOSHI et al., 2015).

Tabela 5 – Requisitos energéticos de gatos em manutenção

Tipos de gatos	Requisitos energéticos (kcal por dia)
Gatos em condição física adequada	$100 \text{ kcal} \times PC^{0,67}$
Gatos obesos	$130 \text{ kcal} \times PC^{0,4}$

Fonte: Adaptado de National Research Council (2006).

O desenvolvimento equivale à fase de desmama até atingir o PC adulto. O momento de maior vigor do crescimento em cães e gatos acontece até os sete primeiros meses de vida. Em raças grandes este intervalo pode se prolongar até por volta dos 12 meses iniciais. Independente da raça, os cães alcançam 50 % da altura corporal muito antes de atingir 50 % do peso adulto. Para aferir as necessidades energéticas de cães em crescimento, o NRC 2006 inclui os cálculos do peso quando adulto e adota o logaritmo natural no caso de modelos de crescimento não-linear (OGOSHI et al., 2015).

As necessidades energéticas de cães em crescimento serão apresentadas na Equação 2.1,

$$EM \text{ (kcal)} = NEM \times 3,2 \times [e^{-0,87q} - 0, 1], \quad (2.1)$$

a constante  $q$  pode ser determinada pela divisão do Peso Corporal no Momento da Avaliação dado em kg por Peso Corporal Esperado quando Adulto dado em kg)(OGOSHI et al., 2015).

Nas necessidades energéticas de gatos devemos considerar a equação 2.2

$$EM \text{ (kcal)} = NEM \times 6,7 \times [e^{-0,189q} - 0, 66]. \quad (2.2)$$

Para uma performance reprodutiva satisfatória, o ganho de PC na gestação engloba o ganho tecidual na preparação para lactação, fetal, placentário e as renovações teciduais.

A alteração de ganho PC de cadelas e gatos distingue notavelmente durante a gestação e lactação. As gatas tendem a emagrecer durante a lactação independentemente da sua dieta. Portanto, é aconselhável para gatas grávidas uma dieta para intensificar 40 % a 50 % de energia, assim para elas a orientação de ingestão de energia passa a ser  $140 \times (PC)^{0,67}$  (OGOSHI et al., 2015).

Cadelas que estão em período gestacional, não requerem muita energia nos vinte dias iniciais da gestação, sendo considerada nessa a fase a Necessidades Energéticas em NEM. Apenas no final da gravidez há um crescimento expressivo dos fetos há um acréscimo de 26 kcal/kg de PC. Nesse caso, a equação 2.3 é útil para encontrar a necessidade diária de energia metabolizável no final da gestação. É importante destacar que a necessidade média para esse caso é de  $130 \text{ kcal} \times PC^{0,75}$  (OGOSHI et al., 2015).

$$EM \text{ (kcal)} = NEM + 26 \text{ kcal} \times PC. \quad (2.3)$$

A medida de leite e a capacidade de energia incluso no leite são essenciais na estimativa das exigências energéticas de cadelas e gatas durante a lactação. O NRC 2006 adota como cálculo as necessidades nessa fase, considerando o número de filhotes e o estágio de lactação (OGOSHI et al., 2015).

A Equação 2.4 calcula as necessidades diárias de energia metabolizável para cadelas em lactação fundamentado no número de filhotes e em semanas de lactação. A necessidade de energia extrapolada para manutenção durante a lactação é  $145 \text{ kcal} \times PC^{0,75}$  (OGOSHI et al., 2015).

$$EM \text{ (kcal)} = NEM + (PC) \times (24n + 12m) \times L, \quad (2.4)$$

a constante  $n$  é o número de filhotes entre 1 e 4 e a constante  $m$  refere-se ao número de filhotes entre 5 e 8 (< 5 filhotes  $m = 0$ )

A Tabela 6 refere-se às gatas lactantes. A constante  $L$  é um fator para o estágio de lactação: semana 1 = 0,75; semana 2 = 0,95; semana 3 = 1,1; semana 4 = 1,2.

## 2.3 ENERGIA DOS ALIMENTOS

No meio dos integrantes dos alimentos, os carboidratos, lipídeos e as proteínas são provedores de energia para o corpo animal. A energia total envolvida na comida é designada de Energia Bruta (EB). Cães e gatos, assim como os demais animais, não conseguem incorporar totalmente a energia do alimento. A energia acessível para o organismo é denominada de EM (OGOSHI et al., 2015).

Tabela 6 – Necessidades diárias de energia metabolizável para gatas lactantes.

Número de filhotes	Necessidade de Energia (kcal)
<3	$EM = NEM + 18 \times (PC) \times L$ $EM \text{ kcal} = 100 \times (PC)^{0,67} + 18 \times (PC) \times L$
3-4	$EM = \text{manutenção} + 60 \times (PC) \times L$ $EM = 100 \times (PC)^{0,67} + 60 \times (PC) \times L$
>4	$EM = \text{manutenção} + 70 \times (PC) \times L$ $EM = 100 \times (PC)^{0,67} + 70 \times (PC) \times L$

Fonte: Adaptado de National Research Council (NRC, 2006).

A EM de um alimento é determinada a partir de resultados experimentais com animais, porém uma ferramenta prática que ajusta o aproveitamento dos animais com equações de predição que estimam as perdas urinárias da Energia Digestível (ED) por equações que determinam a EM. As perdas pela urina são antecipadas pelo conteúdo de proteína digestível, em que para cães utiliza-se a diminuição de 1,25 kcal/g da proteína digestível e para os gatos há o decremento de 0,9 kcal/g da proteína digestível (OGOSHI et al., 2015).

Para atingir o conteúdo de energia metabolizável via equação de predição, o NRC (2006) permite utilizar como base de cálculo a composição química analisada do alimento, o que nos remete ao ano de 1902. Wilbur Olin Atwater foi um químico estadunidense que elaborou estudos sobre a nutrição humana e metabolismo. A ele é creditado os Fatores de Atwater, onde aventou estimar a energia de alimentos delegando-se valores calóricos, evidenciados na Tabela 7. Todavia, estas indicações são admissíveis quando se consideram dietas de alterosa digestibilidade. Para a maioria dos alimentos comerciais para cães e gatos, os quais possui quantidade relevante de carboidratos fibrosos, o químico conduziu sua teoria superestimando os valores de EM. Por isso, são usados esses valores para ração de alta qualidade (OGOSHI et al., 2015; ATWATER, 1910).

Tabela 7 – Fatores Originais de Atwater.

Proteína Bruta (PB)	4 kcal/g
Extrato Etéreo (EE)	9 kcal/g
Extrativo Não Nitrogenados (ENN)	4 kcal/g

Fonte: Adaptado de Carciofi (2007).

Para a aplicabilidade do cálculo da porção alimentar precisamos, executar uma modificação nos Fatores de Atwater, apresentada na Tabela 8. Estas modificações foram realizadas com base na digestibilidade de dietas acessíveis no mercado nas décadas de 70 e 80. São utilizados para ração de baixa qualidade.

A EM de uma ração de alta qualidade é dada pela equação 2.5:

Tabela 8 – Fatores de Atwater modificado.

Proteína Bruta (PB)	3,5 kcal/g
Extrato Etéreo (EE)	8,5 kcal/g
Extrativo Não Nitrogenados (ENN)	3,5 kcal/g

Fonte: Carciofi (2022).

$$EM \text{ (kcal)} = (4 \times PB) + (9 \times EE) + (4 \times ENN), \quad (2.5)$$

onde a PB, EE e Extrativo Não Nitrogenados (ENN) são dadas pela unidade g (grama).

A EM de uma ração de alta qualidade é dada pela equação 2.6, onde a PB, EE e ENN são dadas pela unidade g (grama).

$$EM \text{ (kcal)} = (3,5 \times PB) + (8,5 \times EE) + (3,55 \times ENN) \quad (2.6)$$

## 2.4 CÁLCULO DA PORÇÃO ALIMENTAR

A quantidade de alimento (Q.d.A) a ser suprida é determinada pela Equação 2.7, considerando a Energia Metabolizável (EM) do alimento e a necessidade energética estabelecida para o animal.

$$Q.d.A = \frac{\text{necessidade energética do animal (kcal por dia)}}{\text{energia metabolizável do alimento (kcal por kg)}} \quad (2.7)$$

### 2.4.1 ESTUDO DE CASO

A Figura 3 retrata uma cachorra jovem, nascida em 2017. No dia 05/06/2022 foi aferido o peso de 19,6 kg.

Figura 3 – Lola, 5 anos.



Fonte: O próprio autor (2022).

A primeira etapa envolve calcular as Necessidades Energéticas de Cães em Manutenção (NECM), vista no primeiro item da Tabela 4.

$$NECM = 140 \text{ kcal} \cdot (19,6)^{0.75},$$

$$NECM = 1,304127609 \times 10^3 = 1304 \text{ kcal/kg}.$$

Em posse dos dados de peso, e considerando a idade do cão, observaremos que a ração Coby trás no seu verso todas as informações que necessitamos para realizar o cálculo da quantidade de ração. A Tabela 9 mostra todas as informações nutricionais que estão localizadas no verso da ração.

Tabela 9 – Informações nutricionais da ração de 18 kg.

Umidade Máxima (UM)	120 g/kg
Proteína Bruta (PB)	160 g/kg
Extrato Etéreo (EE)	70 g/kg
Matéria Fibrosa (MF)	60 g/kg
Matéria Mineral (MM)	100 g/kg

Fonte: Verso da ração Coby.

Através da Equação 2.8, calculamos o Extrativo Não Nitrogenado (g/kg) da Ração Coby (OGOSHI et al., 2015):

$$ENN(g/kg) = 1000 - UM - PB - EE - MF - MM, \quad (2.8)$$

$$ENN(g/kg) = 1000 - 120 - 160 - 60 - 70 - 100,$$

$$ENN(g/kg) = 490 \text{ g}.$$

Neste momento do processo é preciso efetuar a estimativa de Energia Metabolizável (EM) para uma ração de baixa qualidade. Dessa forma, utilizando a equação 2.6, tem-se:

$$EMRBQ = (3,5 \cdot 160) + (8,5 \cdot 70) + (3,5 \cdot 490),$$

$$EMRBQ = (560 + 595 + 1715) = 2870 \text{ kcal/kg de alimento}.$$

Assim, a quantidade de ração para o cachorro, utilizando a equação 2.7.

$$QdA = \frac{NECM}{EMRBQ},$$

$$QdA = \frac{1304}{2870},$$

$$QdA = 0,4543554 = 454,355 \text{ g por dia}.$$

Como foi apresentado neste estudo de caso, a quantidade de ração para a Lola de 5 anos, em boa condição física de saúde, é de aproximadamente 454 g de ração.

## 3 METODOLOGIA

No capítulo anterior, onde foi alicerçado os principais conceitos sobre a nutrição e alimentação de cães e gatos. Essas informações são extremamente necessárias para determinar a quantidade correta de ração para os animais. Será essa quantidade que será fragmentada ao longo do dia. Subsequentemente, é necessário fazer uma busca dos principais alimentadores automáticos comercializados e determinar o público alvo para o produto que será implementado. A partir disso, é realizado o desenvolvimento mecânico, de hardware e software do alimentador.

### 3.1 MODELOS COMERCIAIS

Será analisado o modelo de desenvolvimento eletrônico dos três principais alimentadores comercializados.

#### 3.1.1 ALIMENTADOR AUTOMÁTICO PET DOGIS DA020BR

O alimentador indicado na Figura 4 possui programador de refeições, onde é possível definir a quantidade de porções em até quatro vezes ao dia, além disso é possível fazer uma gravação para ser reproduzida no momento primordial. O sistema é alimentado por USB ou com 3 pilhas do tipo D, que só deverão ser acionados em caso de oscilações na rede. Se o sistema é desligado, não perde as informações salvas.

Figura 4 – Alimentador Automático Pet Dogis DA020BR.



Fonte: DOGIS (2022).

A Tabela 10 mostra dados específicos a respeito da estrutura do alimentador.

Tabela 10 – Dados Técnicos do Alimentador Automático Pet Dogis DA020BR.

Modelo	DA020BR
Marca	Dogis
Fonte de Alimentação	5 V e 1 A
Display	LCD
Capacidade	6 Litros
Medidas	21 cm x 39 cm x 25 cm
Peso	2,800 kg
Garantia	12 meses
Embalagem	1 Alimentador 1 Tigela 1 Cabo USB Manual de Instruções

Fonte: DOGIS (2022).

### 3.1.2 COMEDOURO PET AUTOMATIZADO USB 35 LITROS ROBOTIZADO

O comedouro visualizado na Figura 5 é inspirado num robô. Em sua interface com o usuário é possível observar que os botões são *touch*, sendo possível indicar os horários de cada refeição, que pode variar de 1 a 20 porções por dia.

Figura 5 – Comedouro PET Automatizado USB Robotizado Newpet.



Fonte: Concept Shop (2022).

A Tabela 11 mostra os dados do equipamento, observa-se que possui garantia menor que o produto anterior.

Tabela 11 – Dados Técnicos do Comedouro PET Automatizado USB Robotizado Newpet.

Modelo	RobPAF-35L
Marca	NewPet
Fonte de Alimentação	5 V e 1 A
Capacidade	1 kg
Medidas	37 cm x 20 cm x 19 cm
Peso	1,7 kg
Garantia	3 meses
Embalagem	1 Alimentador 1 Tigela 1 Cabo USB Manual de Instruções

Fonte: Concept Shop (2022).

### 3.1.3 COMEDOURO AUTOMÁTICO PERFECT PET DINNER

O alimentador apontado na Figura 6 é indicado para animais de pequeno porte, onde é possível programar no máximo 3 refeições ao dia com uma porção que pode variar de um quarto a 3 xícaras por porção. É possível gravar a voz do tutor para ser acionada no momento da alimentação.

Figura 6 – Comedouro Automático Perfect Pet Dinner.



Fonte: Lorben (2022).

A Tabela 12 indica que na embalagem além dos componentes esperados há a garrafa de 5 litros que comporta a ração.

Tabela 12 – Dados Técnicos do Alimentador Digital Eletrônico Programável Lorben.

Modelo	GT652
Marca	Lorben
Fonte de Alimentação	3 Pilhas do tipo D
Capacidade	1,23 kg
Medidas	40 cm x 37 cm x 20 cm
Peso	1,6 kg
Garantia	3 meses
Embalagem	1 Alimentador 1 Tigela 1 Comedouro Manual de Instruções

Fonte: Lorben (2022).

### 3.1.4 COMPARATIVO ENTRE EQUIPAMENTOS COMERCIAIS

A Tabela 13 mostra principais características dos equipamentos comerciais que fizeram parte do mapeamento do público-alvo do projeto que será implementado. A verificação do valor de cada produto foi feito no dia 11/07/2022.

Tabela 13 – Comparação dos alimentadores automáticos comerciais.

Equipamento	Display LCD	Touch	Quantidade Diária	Quantidade Total	Valor
Alimentador Automático Pet Dogis DA020BR	Sim	Não	4	2,8 kg	R\$ 599,90
Comedouro PET Automatizado USB 35 litros Robotizado Lorben	Sim	Sim	20	1,00 kg	R\$ 415,00
Comedouro Automático Perfect Pet Dinner	Sim	Não	3	1,23 kg	R\$ 369,86

Fonte: O próprio autor (2022).

Observa-se que para todos os alimentadores a estética foi muito valorizada. Porém a prioridade de um alimentador é proporcionar uma alimentação saudável através de porções divididas ao longo do dia. Dessa forma, proponho fazer a liberação da ração através de um motor DC, onde no recipiente existe uma célula de carga para fazer a correta especificação da porção alimentar.

## 4 DESENVOLVIMENTO DO PROJETO

Este capítulo apresenta os materiais e métodos utilizados, onde as três etapas de desenvolvimento do projeto são: mecânico, hardware e software.

### 4.1 DESENVOLVIMENTO MECÂNICO

Para a implementação do protótipo do alimentador deste projeto foram fabricadas pelas peças plásticas utilizando impressoras em 3D. O filamento adequado para esse tipo de prototipagem é o PLA pois é um termoplástico biodegradável de origem natural e de fontes renováveis, como amido de milho ou cana-de-açúcar. A fabricação do objeto ocorre por extrusão ou difusão de plástico (FDM), ou seja, o PLA de 1,75 mm, é aquecido até atingir uma temperatura de fusão, transformando-o em fluido, que se corporifica depois de entrar em contato com uma superfície com uma temperatura mais fria. O processo se repete camada após camada até que a peça seja finalizada (ZUCCA et al., 2019; REIS, 2016).

Na Figura 7 são mostradas as duas impressoras implementadas no quinto bloco do IFRS *Campus Restinga*.

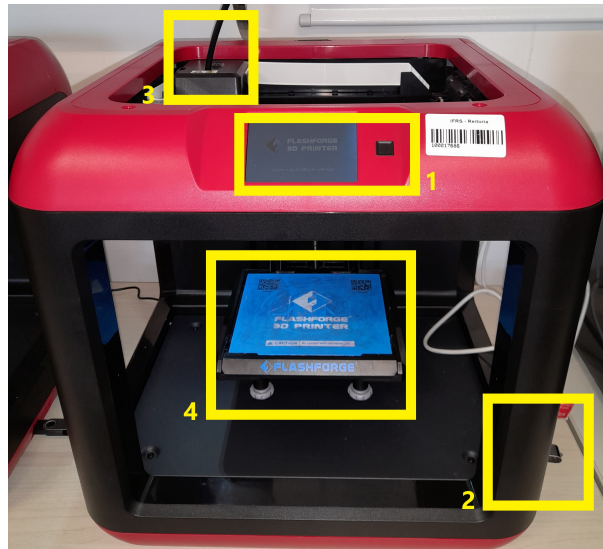
Figura 7 – Impressoras 3D FlashForge Finder.



Fonte: O próprio autor (2022)

A Figura 8 possui 3 principais setores: 1) Interface de comunicação com usuário: é feita a seleção para imprimir o objeto armazenado no computador ou no *pen drive*; 2) Entrada: possui duas portas, através dos conectores USB (*Universal Serial Bus*) tipo A e B; 3) Carregamento: onde é realizado a alimentação ou remoção do filamento e 4) Mesa: o objeto é impresso nesse compartimento.

Figura 8 – Destaques da Impressora 3D FlashForge Finder.

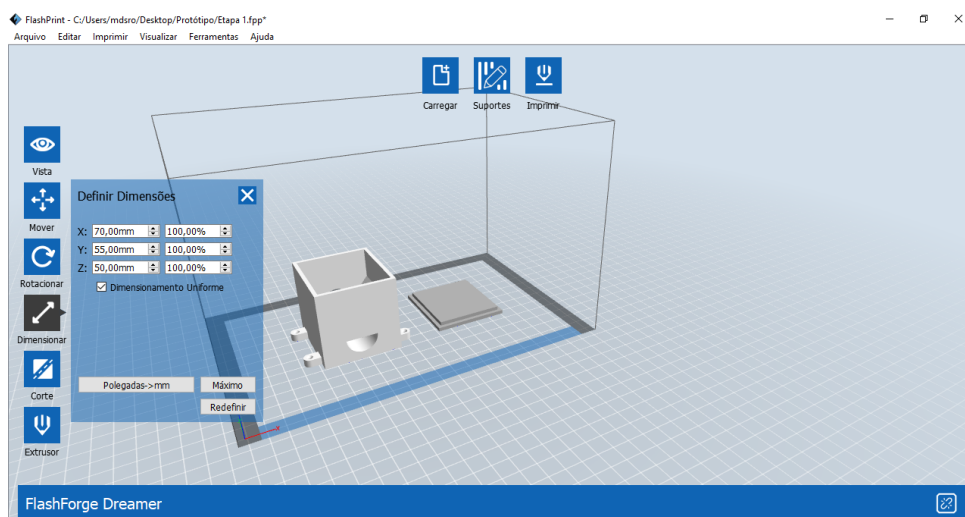


Fonte: O próprio autor (2022).

#### 4.1.1 RESERVATÓRIO

O reservatório do alimentador é onde será armazenado a quantidade diária total do alimento. Com o auxílio do software *FlashPrint*, versão 4.6.4 fazemos a impressão desse objeto. Foi escolhido um modelo de reservatório do *MakerBot - Thingiverse*, na Figura 9 apresenta o componente (70 mm, 55 mm, 50 mm) e a tampa (50 mm, 55 mm, 5 mm) tiveram suas dimensões alteradas para teste com o motor de 5 V que seria utilizado.

Figura 9 – Desenho do reservatório e a tampa no FlashPrint.

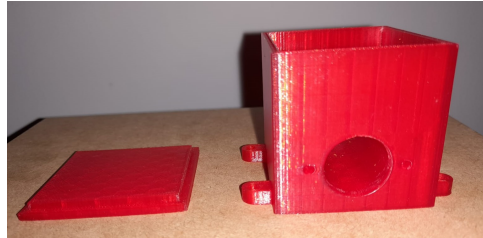


Fonte: O próprio autor (2022).

O software gera um arquivo em camadas que tem seu formato *.gx*. É este tipo de arquivo que é compatível com a impressora 3D.

Observe que a Figura 10 apresenta as peças resultantes da fabricação usando a impressora 3D.

Figura 10 – Resultado da impressão do reservatório e da tampa.



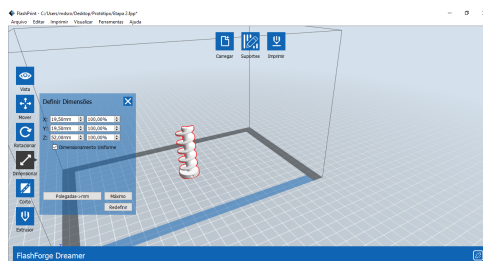
Fonte: O próprio autor (2022).

#### 4.1.2 EIXO HELICOIDAL

O transportador helicoidal é um dos métodos para movimento relativo entre o material e a sua estrutura. Aproveitado para pequenas vazões em pequenas distâncias (SACRAMENTO, 2015).

O eixo helicoidal projetado na Figura 11, de dimensões 19,50 mm, 19,50 mm e 52 mm, é um método para realizar o transporte da ração para dentro do pote do cão ou gato.

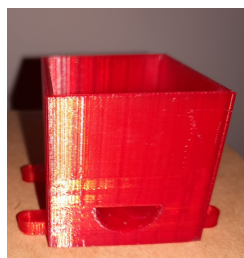
Figura 11 – Desenho do eixo helicoidal no FlashPrint.



Fonte: O próprio autor (2022).

Na Figura 12 percebe-se que existe um semicírculo na lateral do reservatório, é por esse compartimento que será transportado a ração.

Figura 12 – Compartimento para liberação da ração.



Fonte: O próprio autor (2022).

A Figura 13 mostra o eixo helicoidal com seis camadas, suficientes para transportar a ração sem trancar nenhum grão na estrutura.

Figura 13 – Resultado da impressão do eixo helicoidal.



Fonte: O próprio autor (2022).

## 4.2 DESENVOLVIMENTO DO HARDWARE

Para ligar o alimentador será utilizado um botão táctil. Enquanto ele estiver acionado o alimentador funcionará normalmente. Quando o botão é desligado as informações continuam salvas no microcontrolador.

Acionado o sistema, a interface solicita três entradas: a tipologia do animal, faixa etária e quantas vezes será fragmentada a ração. O comedouro terá na sua tampa acoplada um sensor ultrassônico que informará a quantidade de ração que está no depósito. O motor 12 V acionado em horários pré-programados para liberar o alimento. Após a liberação da ração é depositada no recipiente do animal até que chegue na quantidade indicada pelo sistema de pesagem.

### 4.2.1 PROCESSAMENTO DE DADOS

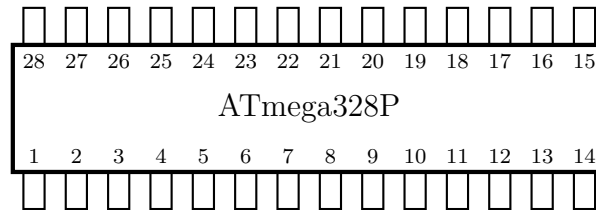
A plataforma de desenvolvimento do projeto é a placa Michaelino, cuja inspiração é o Arduino UNO.

Criado em 2005, o Arduino é uma empresa de hardware e software de computador de código aberto. O Arduino UNO é uma ferramenta de prototipagem eletrônica, projetada com um microcontrolador Atmel AVR com suporte de entrada/saída incluso, uma linguagem de programação padrão, essencialmente C/C++. A principal ideia do projeto é criar mecanismos que são acessíveis, com baixo custo, que podem ser usados por profissionais da área ou iniciantes (FEZARI; DAHOUD, 2018).

A Figura 14 mostra o desenho eletrônico do microcontrolador ATMEGA328P de encapsulamento DIP (*Dual In-line Package*). É uma proteção que envolve o chip de silício, sua principal função é proteger o chip contra a umidade, corrosão e isolar os pinos. Esse

tipo de encapsulamento será utilizado na placa Michaelino porque o encapsulamento QFP teria que ser soldado na placa, deste modo não facilitaria a troca do componente.

Figura 14 – ATmega328 no encapsulamento DIP.



Fonte: O próprio autor (2022).

A Tabela 14 contém a descrição completa dos 28 pinos do microcontrolador, sendo que 23 podem ser utilizados como entrada/saída.

Tabela 14 – Descrição dos Pinos do ATmega328P.

1				RESET	PCINT5	PC6
2			0	RX0	PCINT16	PD0
3			1	TX0	PCINT17	PD1
4			2	INT0	PCINT18	PD2
5	OC2B	PWM	3	INT1	PCINT19	PD3
6		T0	4	XCK	PCINT20	PD4
7	VCC					
8	GND					
9				OSC1	XTAL1	PCINT6
10				OSC2	XTAL2	PCINT7
11	OC0B	PWM	5	T1	PCINT21	PD5
12	OC0A	PWM	6	AIN0	PCINT22	PD6
13			7	AIN1	PCINT23	PD7
14		ICP1	8	CLK0	PCINT0	PB0
15		PWM	9	OC1A	PCINT1	PB1
16	SS	PWM	10	OC1B	PCINT2	PB2
17	MOSI	PWM	11	OC2A	PCINT3	PB3
18			12	MISO	PCINT4	PB4
19			13	SCK	PCINT5	PB5
20	VCC					
21	AREF					
22	GND					
23			A0	ADC0	PCINT8	PC0
24			A1	ADC1	PCINT9	PC1
25			A2	ADC2	PCINT10	PC2
26			A3	ADC3	PCINT11	PC3
27		SDA	A4	ADC4	PCINT12	PC4
28		SCL	A5	ADC5	PCINT13	PC5

Fonte: O próprio autor (2022).

Na Figura 15 em destaque estão todos os conectores que é possível alimentar o circuito.

Figura 15 – Conectores da placa Arduino UNO.



Fonte: O próprio autor (2022).

A Tabela 15 mostra a função de cada pino de alimentação do Arduino.

Tabela 15 – Pinos de alimentação do Arduino UNO.

IOREF	Fornecer uma tensão de referência para o circuito selecionar o tipo de interface apropriada. Existem circuitos que são alimentados com 3,3 V e podem ser adaptados para ser utilizado 5 V.
RESET	Reset externo.
3,3 V	Tensão positiva de 3,3 V e corrente máxima de 50 mA.
5 V	Tensão positiva de 5 V.
GND	Terra.
VIN	Quando a placa é alimentada através do conector <i>Jack</i> , a tensão da fonte estará nesse pino

Fonte: Adaptado de ViaHobby (2022).

O microcontrolador ATMEGA328 é um dispositivo de 8 bits da família AVR com arquitetura RISC avançada. Possui 32 KB de *Flash* (mas 512 Bytes são utilizados para o *bootloader*, 2 KB de RAM e 1 KB de EEPROM (*Electrically Erasable Programmable Read-Only Memory*)), possibilitando operar até 20 MHz, porém a placa do Arduino UNO opera em 16 MHz, valor do oscilador cristal externo.

A Tabela 16 apresenta as principais características das placas do Arduino UNO e Michaelino. Observe que a alimentação da plataforma original apresenta mais versatilidade pois suporta uma alimentação de 7 a 12 V enquanto a placa Michaelino possui apenas uma alimentação de 9 V.

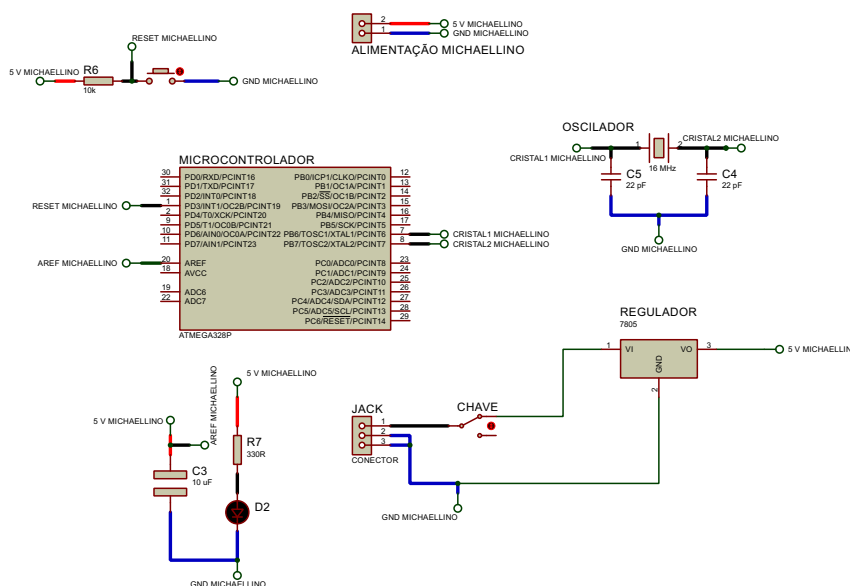
Tabela 16 – Comparativo das características periféricas das placas Arduino e Michaelino.

Periférico da Placa	Arduino UNO (R3)	Michaelino
Microcontrolador	ATmega328P	ATmega328P
CPU Velocidade	16 MHz	16 MHz
Conexões	USB	JACK (3 pinos)
Memória Flash	32 KB	32 KB
Memória Bootloader	0,5 KB	0,5 KB
Memória SRAM	2 KB	2 KB
Memória EEPROM	1 KB	1 KB
Portas PWM	6	6
Portas Digitais	14	14
Portas Analógicas	6	6
Tensão das Portas	5 V	5 V
Corrente das Portas	40 mA	40 mA
I2C (TWI)	1	1
SPI	1	1
UART	1	1
Alimentação	7-12 V	9 V

Fonte: Adaptado de FVM Learning (2022).

A Figura 16 mostra o circuito simulado no Proteus Release, versão 8.13, para a verificação do funcionamento dos periféricos da placa que será desenvolvida. Observe que existe uma chave seletora no conector de três pinos que está representando o conector Jack. Entra no conector 9 V mas essa tensão é reduzido para 5 V feito pelo regulador 7805.

Figura 16 – Diagrama elétrico da placa Michaelino versão 1.0.



Fonte: O próprio autor (2022).

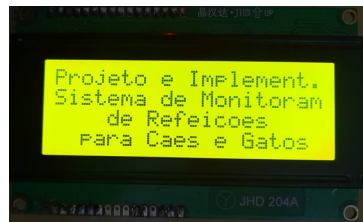
## 4.2.2 INTERFACE COM O USUÁRIO

A interface com o usuário inclui o mostrador digital e as entradas dos botões do controle remoto.

### 4.2.2.1 Mostrador Digital

A Figura 17 mostra um LCD (*Liquid Crystal Display*), versão JDH-204A com 16 pinos.

Figura 17 – Mostrador digital de vinte caracteres por quatro linhas.



Fonte: O próprio autor (2022).

A Tabela 17 é composta com a função de cada pino.

Tabela 17 – Pinos do mostrador digital.

Pinos	Função	Pinos	Função
1	VSS (Ground)	9	D2 (Data Pino 2)
2	VDD (5 V)	10	D3 (Data Pino 3)
3	VE (Contraste)	11	D4 (Data Pino 4)
4	RS (Register Select)	12	D5 (Data Pino 5)
5	RW (Read/Write)	13	D6 (Data Pino 6)
6	E (Enable)	14	D7 (Data Pino 7)
7	D0 (Data Pino 0)	15	LED Positivo
8	D1 (Data Pino 1)	16	LED Negativo

Fonte: Adaptado de Sunrom Electronics (2022).

O primeiro contato do usuário com a interface do alimentador está na Figura 18, onde é possibilitado a seleção do animal que deverá ser alimentado. Se cão, a opção um deve ser selecionado ou se gato, a opção dois deve ser marcada no controle remoto.

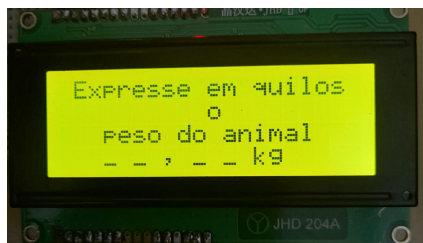
Figura 18 – Seleção inicial do sistema do alimentador.



Fonte: O próprio autor (2022).

A Figura 19 mostra o momento em que é necessário declarar o PC do animal pois é um dos valores iniciais para o cálculo de NE do cão ou gato.

Figura 19 – Informação do PC do animal no sistema do alimentador.



Fonte: O próprio autor (2022).

A Figura 20 mostra ao usuário onde deve ser preenchido a idade do bicho em anos. Isso é importante para que haja a identificação da fase etária que o animal pertence.

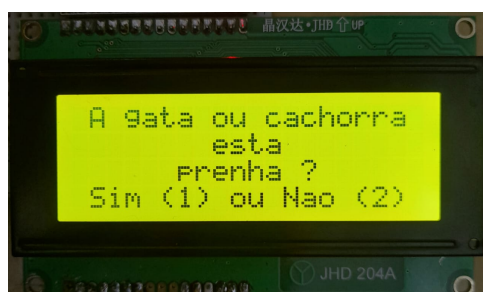
Figura 20 – Faixa etária no sistema do alimentador.



Fonte: O próprio autor (2022).

A Figura 21 o usuário é questionado sobre a gravidez do animal em questão. Para que haja a correta identificação da equação que proporciona a NE para cadelas e gatas em período gestacional.

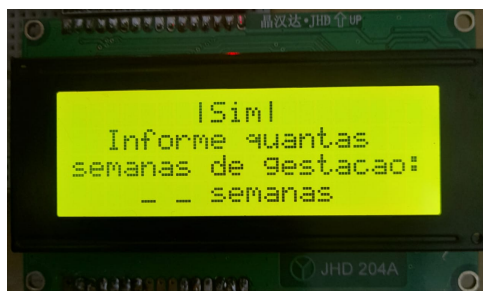
Figura 21 – Teste gestacional do animal no sistema do alimentador.



Fonte: O próprio autor (2022).

Na Figura 22 é mostrado a resposta em caso afirmativo, deste modo, é questionado sobre as semanas de gestação, assim é utilizado a equação das necessidades diárias de energia metabolizável para cadelas ou gatas gestantes.

Figura 22 – Resposta positiva para o teste gestacional no sistema do alimentador.



Fonte: O próprio autor (2022).

Se a resposta é negativa, a Figura 23 mostra outro teste, relacionado a lactação do animal, pois é fundamentada no número de filhotes e nas semanas de lactação.

Figura 23 – Teste de lactação do animal no sistema do alimentador.



Fonte: O próprio autor (2022).

Em caso afirmativo, a Figura 24 mostra que é necessário informar o número de filhotes do animal.

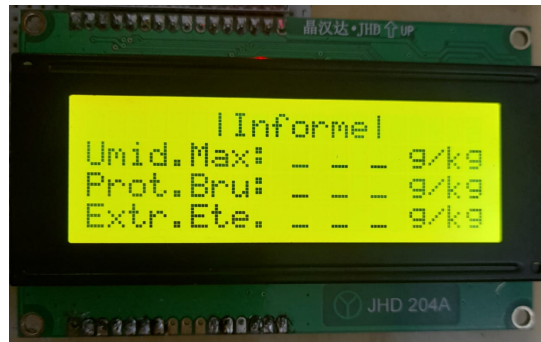
Figura 24 – Informação de número de filhotes no sistema do alimentador.



Fonte: O próprio autor (2022).

A Figura 25 mostra a primeira parte dos parâmetros da ração que são necessárias para encontrar o valor de gordura do alimento do animal.

Figura 25 – Primeira parte dos parâmetros para calcular o valor total de gordura do alimento.



Fonte: O próprio autor (2022).

A Figura 26 mostra a segunda parte dos parâmetros da ração para que haja o valor total de gordura do alimento do animal.

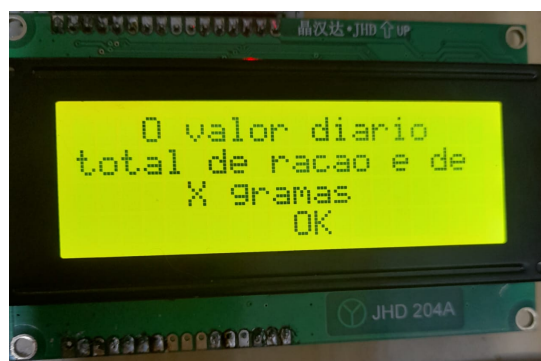
Figura 26 – Segunda parte dos parâmetros para calcular o valor total de gordura do alimento.



Fonte: O próprio autor (2022).

Na Figura 27 o sistema já mostra para o usuário o valor total da porção diária.

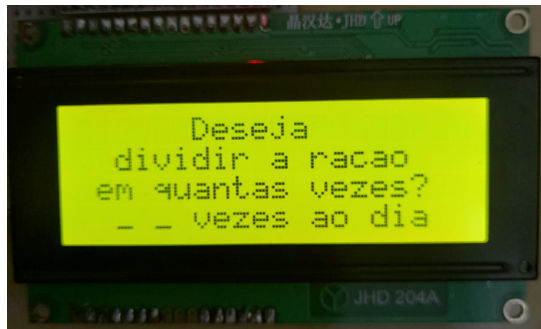
Figura 27 – O sistema do alimentador informa ao usuário o valor total de ração em gramas.



Fonte: O próprio autor (2022).

A Figura 28 há a chance de escolha de quantas vezes é possível efetuar a fragmentação da comida.

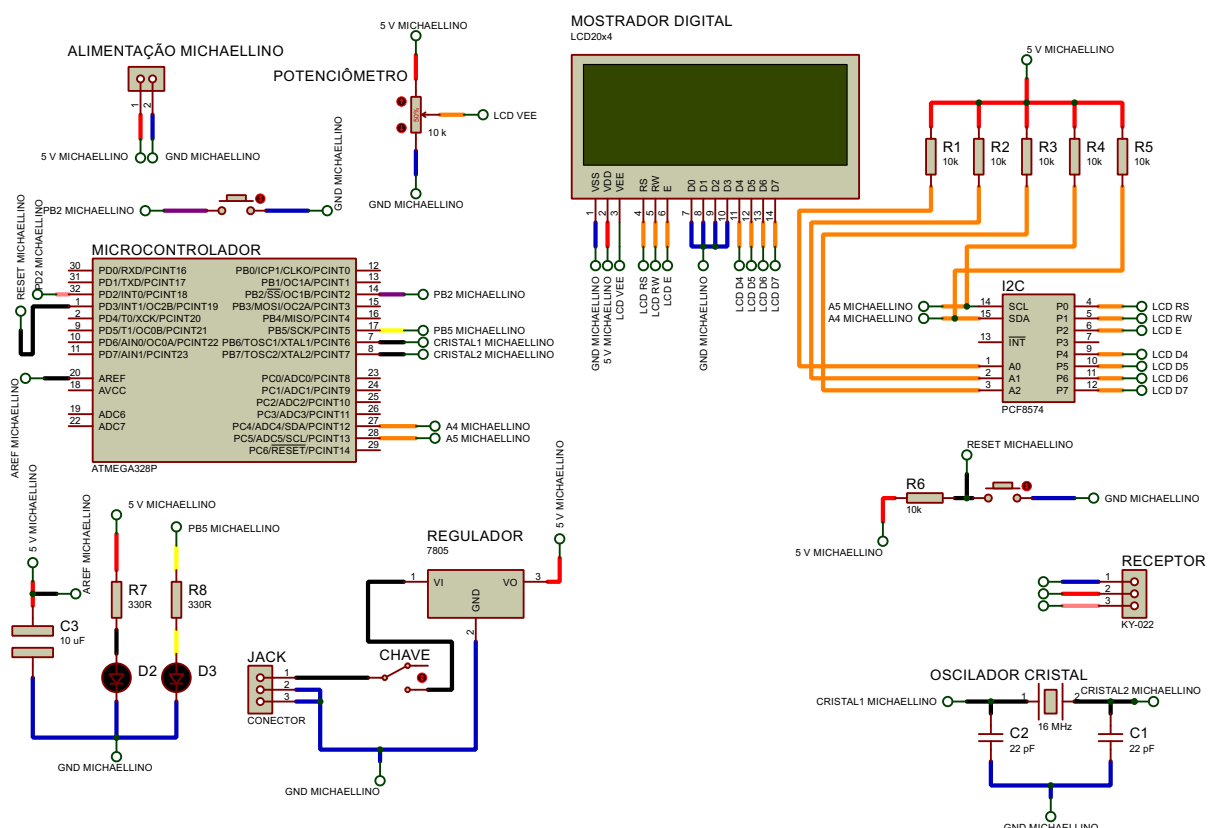
Figura 28 – O sistema do alimentador permite ao usuário o número exato que deseja fragmentar a comida do animal.



Fonte: O próprio autor (2022).

A Figura 29 mostra a primeira parte do esquemático elétrico detalhado da placa desenvolvida, Michaellino e do mostrador digital, representado por um LCD de vinte caracteres por quatro linhas.

Figura 29 – Esquemático elétrico da placa Michaellino com o mostrador digital.



Fonte: O próprio autor (2022).

Observe que a Figura 29 possui características diferenciadas nas suas cores e na metodologia de conexão dos fios. A alimentação positiva possui 5 V e 0 V representado pelo GND, vermelho e azul, respectivamente. As ligações em preto fazem parte da placa

Michaellino. As conexões em laranja representam a junção do LCD e do PCF8574. O potenciômetro serve para ajustar o contraste do LCD. Em roxo está representado o botão que liga o sistema do alimentador. Em amarelo está o LED azul que faz a indicação de quando há pouca ração no recipiente.

#### 4.2.2.2 Receptor Infravermelho e Controle Remoto

A manipulação das seleções do sistema do alimentador é feito através do controle remoto (KEYS). Ele é alimentado por uma bateria do tipo CR25, possui 17 teclas: numéricas (0-9), setas direcionais, seleção de OK, botão cerquilha e asterisco.

A Figura 30 mostra a estrutura física do controle remoto

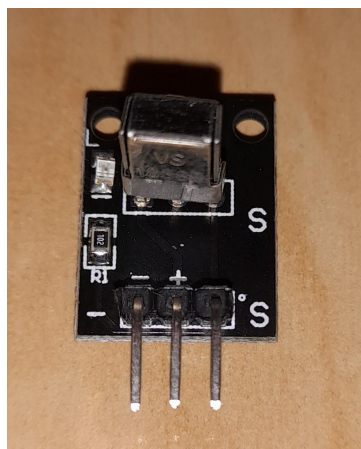
Figura 30 – Controle Remoto (KEYS).



Fonte: O próprio autor (2022).

A Figura 31 mostra o receptor Receptor Infravermelho KY-022 38 KHz.

Figura 31 – Receptor infravermelho KY-022.



Fonte: O próprio autor (2022).

A Tabela 18 mostra as principais características do módulo KY-022.

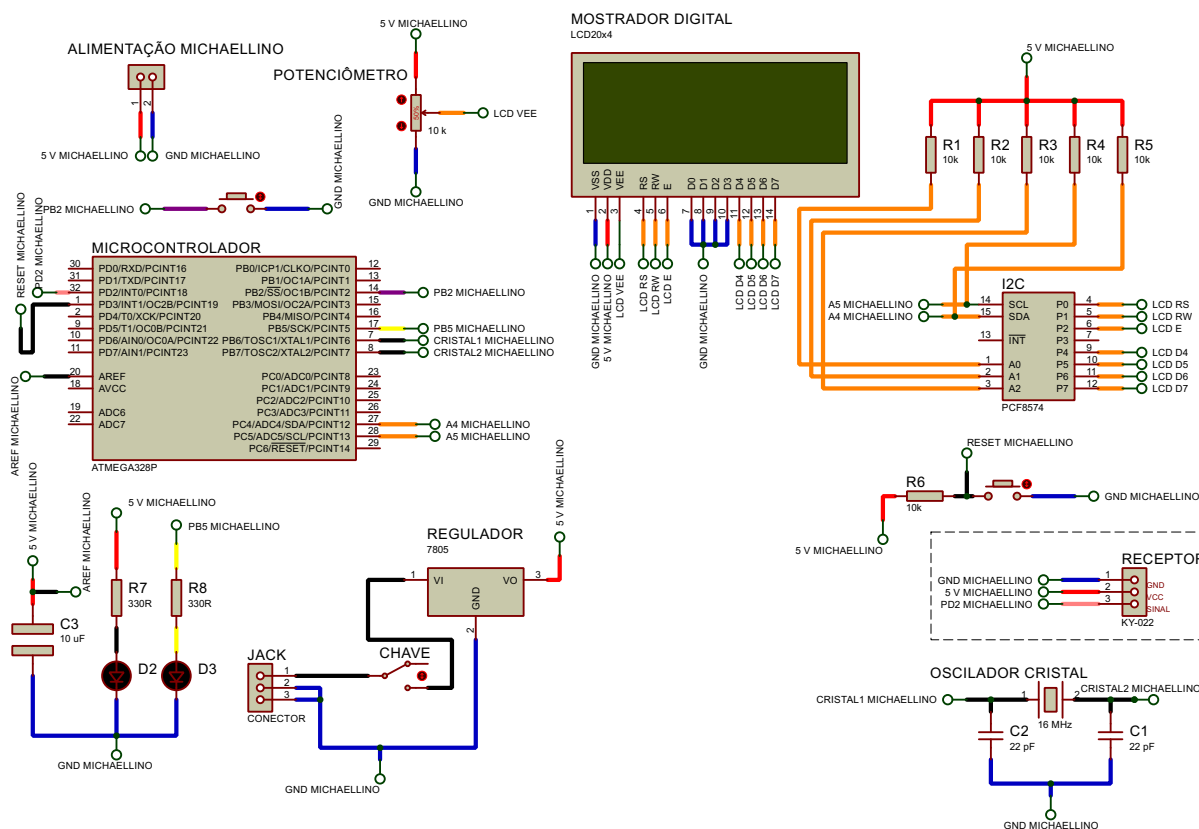
Tabela 18 – Características elementares do receptor KY-022.

Tensão de operação	2,7 - 5,5 V
Corrente de operação	0,4 - 1,5 mA
Distância de recepção	18 m
Ângulo de recepção	$\pm 45^\circ$
Frequência	38 KHz
Tensão de nível baixo	0,4 V
Tensão de nível alto	4,5 V
Filtro de luz do ambiente	acima de 500 LUX

Fonte: Adaptado de Joy-it (2022).

A Figura 32 mostra o circuito da interface que faz a comunicação com o usuário. Observe que o receptor KY-022 está representado por um conector de três pinos, destacando cada pino ao lado direito do componente.

Figura 32 – Circuito da interface do projeto.



Fonte: O próprio autor (2022).

### 4.2.3 MONITORAMENTO

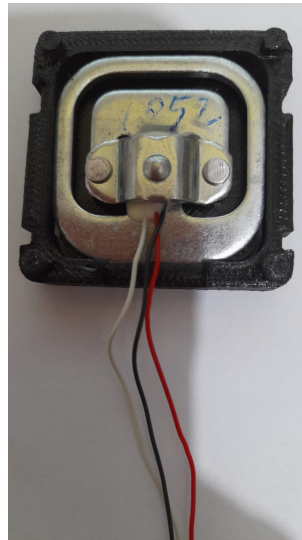
O monitoramento do sistema do alimentador é realizado através de da exibição das informações no mostrador digital e o monitoramento do recipiente do reservatório para

verificação da quantidade de ração.

#### 4.2.3.1 Controle da Porção Alimentar

A célula de carga será posicionado abaixo do recipiente, na Figura 33 observamos o componente comprado para o projeto.

Figura 33 – Célula de carga no acessório do recipiente do animal.



Fonte: O próprio autor (2022).

A célula de carga é um sensor que funciona a partir princípios físicos para determinar o peso. Há muitas informações que podem ser utilizadas, tais como a tensão, pressão e a força, e são convertidos em sinais elétricos através de tensões elétricas, frequência ou corrente. Os extensômetros resistivos aproveitam uma pressão mecânica em uma estrutura deformável, formando células de carga utilizadas para mensurar grandezas como força e pressão, ligados a partir de uma ponte de Wheatstone (VIEIRA, 2016).

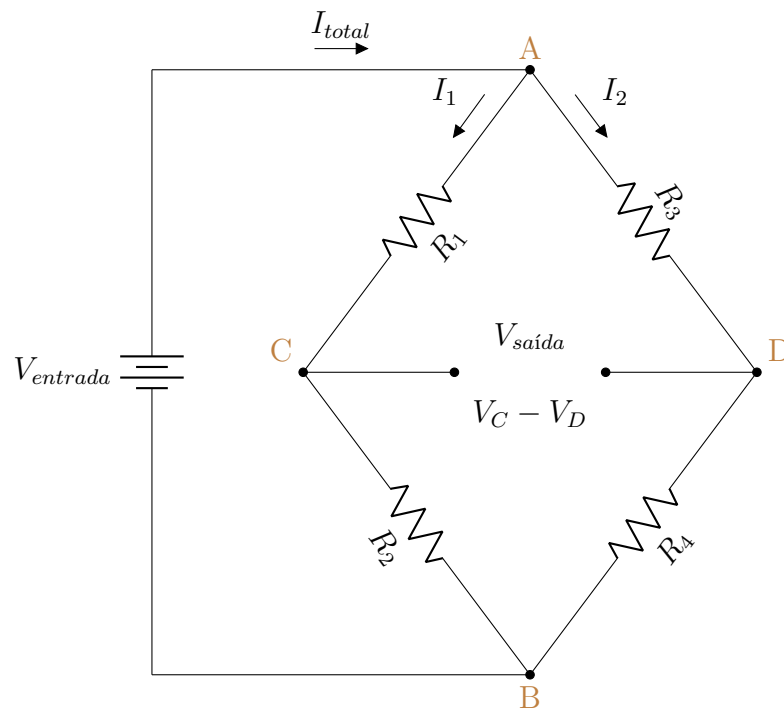
A Figura 34 mostra o circuito que gera sinais elétricos a partir de extensômetros resistivos. Uma rede de quatro pernas resistivas interligadas, representando dois divisores de tensão em paralelo (NETO et al., 2001).

Os resistores  $R_1$  e  $R_2$  em composição produzem o primeiro divisor de tensão,  $R_3$  e  $R_4$  formam o segundo,  $V_{saída}$  é aferida entre os nós C e D. Quando há uma mudança de tensão, é gerada pelo peso, a resistência dos elementos da ponte é alterada. O número de elementos resistivos do componente utilizará a configuração da ponte (NETO et al., 2001).

A Tabela 19 mostra as características de precisão da célula de carga.

Na Figura 35 é observado o componente que faz a conversão conversão analógico-digital da célula de carga, o HX711.

Figura 34 – Ponte de Wheatstone.



Fonte: O próprio autor (2022).

Tabela 19 – Características elementares da célula de carga de 50 kg.

Precisão	$\pm 0,05\%$ ou $\pm 25$ g
Capacidade máxima	50 kg
Conexão	3 fios
Dimensões	34 x 34 x 7 mm

Fonte: Circuit Journal, 2022.

Na Tabela 20 é observado as principais características elétricas do conversor analógico-digital.

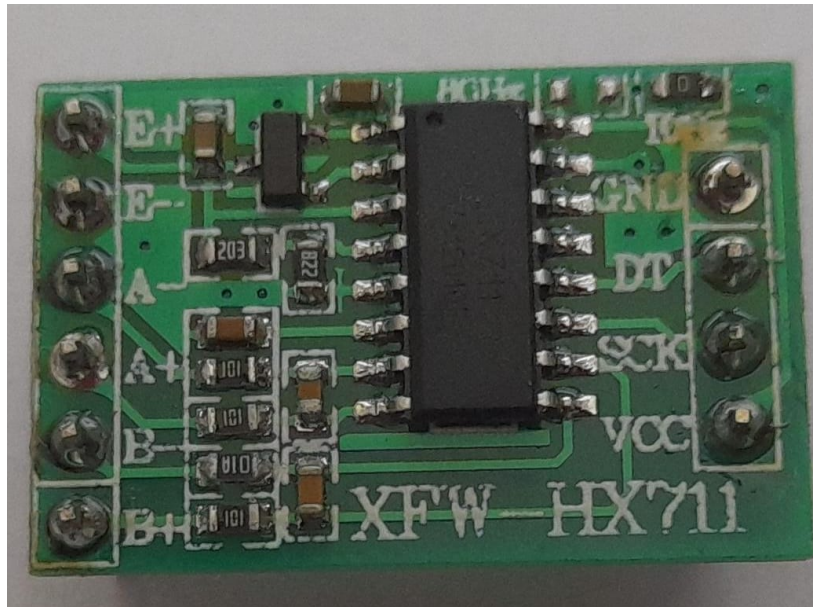
Tabela 20 – Características elementares do conversor analógico-digital de 24 bits HX711.

Tensão de operação	4,8 à 5,5 V
Corrente de operação	1,6 mA
Temperatura de operação	-20 à 85°C
Interface	SPI
Dimensões	29 x 17 x 4 mm

Fonte: AVIA Semiconductor, 2022.

O processo eletrônico responsável pela transformação de um sinal analógico em seu equivalente digital. É necessário fazer esse procedimento porque o sinal analógico não é estável, e a cada intervalo existem infinitos valores de corrente analógica. O modo que ocorre a alteração em sinal digital é formado resumidamente pela amostragem de alguns

Figura 35 – Conversor analógico-digital HX711.



Fonte: O próprio autor (2022).

destes valores analógicos recebidos em valores fixos, sendo que quanto maior o número de bits recebidos maior é o número de sinais analógicos amostrados para representar um sinal digital, e menor é o erro (FALLEIRO, 2015).

#### 4.2.3.2 Controle de Horários

O controle do horário que será programado está associado ao módulo RTC (Real Time Clock). O módulo escolhido foi o DS3231 porque computa segundos, minutos, horas, dias da semana, dias do mês e anos, sendo que suporta até o ano de 2099 no seu sistema (MAXIM, 2006).

Na Figura 36, que possui uma bateria do tipo CR2032, para o caso de falta de energia no módulo.

Figura 36 – Módulo RTC DS3231.



Fonte: Eletrônica Castro (2022).

A Tabela 21 apresenta as principais características elétricas do módulo RTC.

Tabela 21 – Características elementares do módulo RTC DS3231.

Tensão de operação	3,3 - 5 V
Precisão do sensor de temperatura	$\pm 3^{\circ}\text{C}$
Memória	AT24C32 (Capacidade de 32 K bytes que podem ser usadas como RAM estendida do microcontrolador)
Interface	I2C
Faixa de temperatura	0 a $40^{\circ}\text{C}$
Dimensões	38 x 22 x 14 mm

Fonte: Maxim Integrated, 2022.

O RTC mantém segundos, minutos, horas, dia, data, informações de mês e ano. A data no final do mês é ajustado automaticamente para meses com menos de 31 dias, incluindo correções em casos de ano bissexto. o relógio opera no formato de 24 horas ou 12 horas com um indicador AM/PM. Dois horários programáveis alarmes e uma saída de onda quadrada programável são fornecidos. Endereço e dados são transferidos em série através de um barramento bidirecional I2C (MAXIM, 2006).

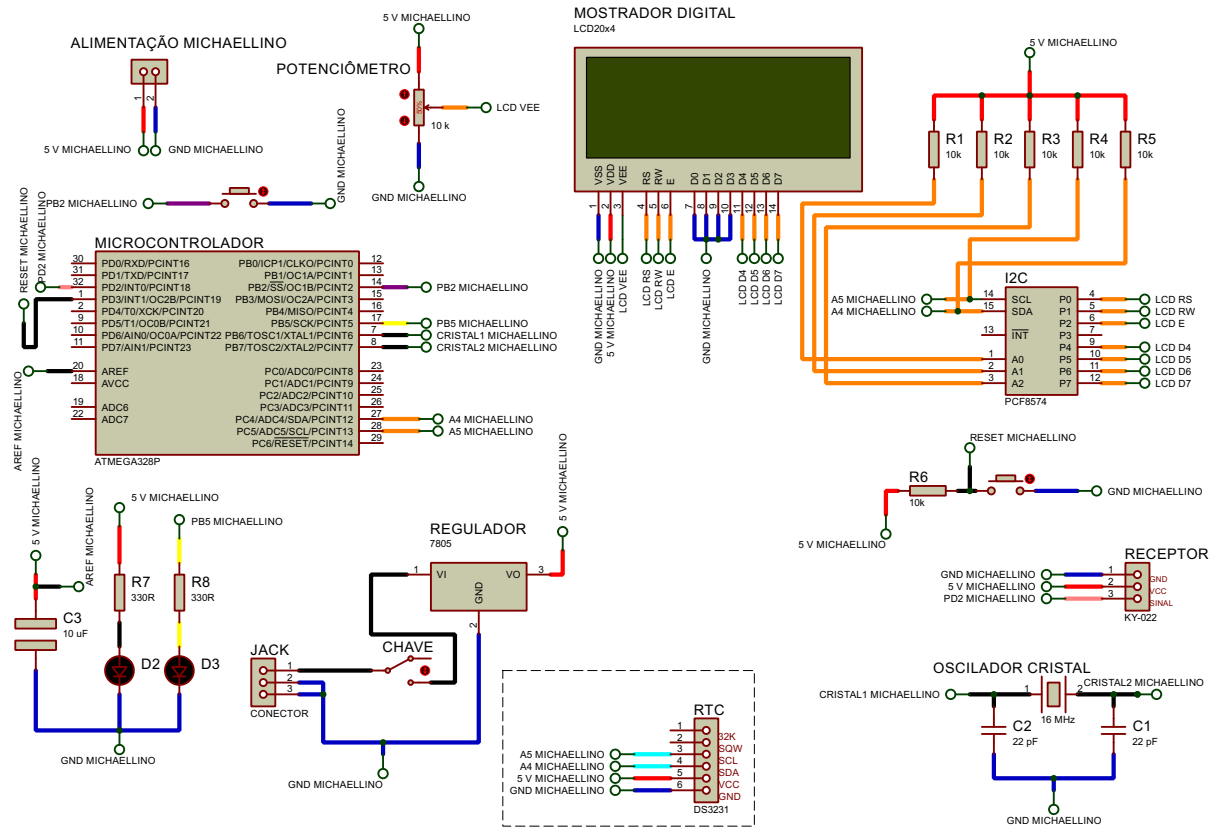
O módulo está conectado à fonte de alimentação de +5 V. Seu funcionamento ocorre com alimentação regulada de +5V e uma tensão mais alta pode danificar o componente. Será conectado os pinos SDA do módulo ao SDA da placa Michaelino e o SCL é conectado ao SCL da placa Michaelino. O objetivo é simplificar e padronizar a forma de comunicação entre o microcontrolador. O I2C é um protocolo para curtas distâncias, dentro da mesma placa de circuito impresso, média velocidade, e necessita de apenas dois fios para comunicação (SDA – dados, e SCL (*clock*)) (MAXIM, 2006).

Normalmente a informação é enviada ou recebida byte a byte. Portanto, usar bibliotecas próprias para o módulo DS3231 é recomendado pelo fabricante, pois facilita a comunicação. Uma vez que o arquivo é incluído no programa, o controlador realiza a comunicação sozinho e fornece a data e hora (MAXIM, 2006).

Quando a energia é interrompida, o chip do módulo RTC extrai a energia da fonte de bateria conectada a ele automaticamente. Assim, o tempo estará atualizado. E quando o sistema é reiniciado, o controlador pode obter o tempo real do módulo sem erro (MAXIM, 2006).

A Figura 37 possui o módulo RTC com as conexões em azul claro, observando que fazem parte das mesmas ligações do mostrador digital. Percebe que no lado direito do conector que está representado o sensor existe a descrição de cada um dos seis pinos.

Figura 37 – Esquemático elétrico da placa Michaelino e do RTC DS3231.



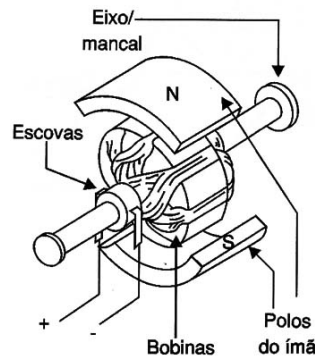
Fonte: O próprio autor (2022).

#### 4.2.3.3 Controle do Motor

O motor elétrico é um dispositivo que converte energia elétrica em energia mecânica. É definido pelos seguintes segmentos: armadura (rotor), bobinas de campo (polos), comutador e escovas (TORGA, 2016).

A Figura 38 mostra os principais mecanismos de um motor.

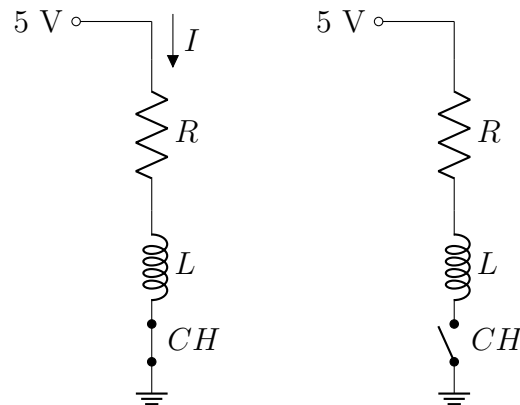
Figura 38 – Partes essenciais de um motor em corrente contínua



Fonte: Instituto Newton C. Braga (2022).

A Figura 39 apresenta um modelo simples de um motor em corrente contínua, onde o motor é representado pela resistência e a indutância. O indutor armazena a energia elétrica, com isso é necessário fornecer uma maneira de dissipar energia elétrica de maneira segura. Abrir o interruptor não interrompe imediatamente a corrente nos enrolamentos do motor, isso é caracterizado pelo comportamento indutivo, que faz com que a corrente continue a fluir quando o interruptor é aberto repentinamente.

Figura 39 – Modelagem simples de um motor.



Fonte: O próprio autor (2022).

A Figura 40 mostra o motor de 12 V conectado no eixo helicoidal para fazer o transporte da ração, grão a grão.

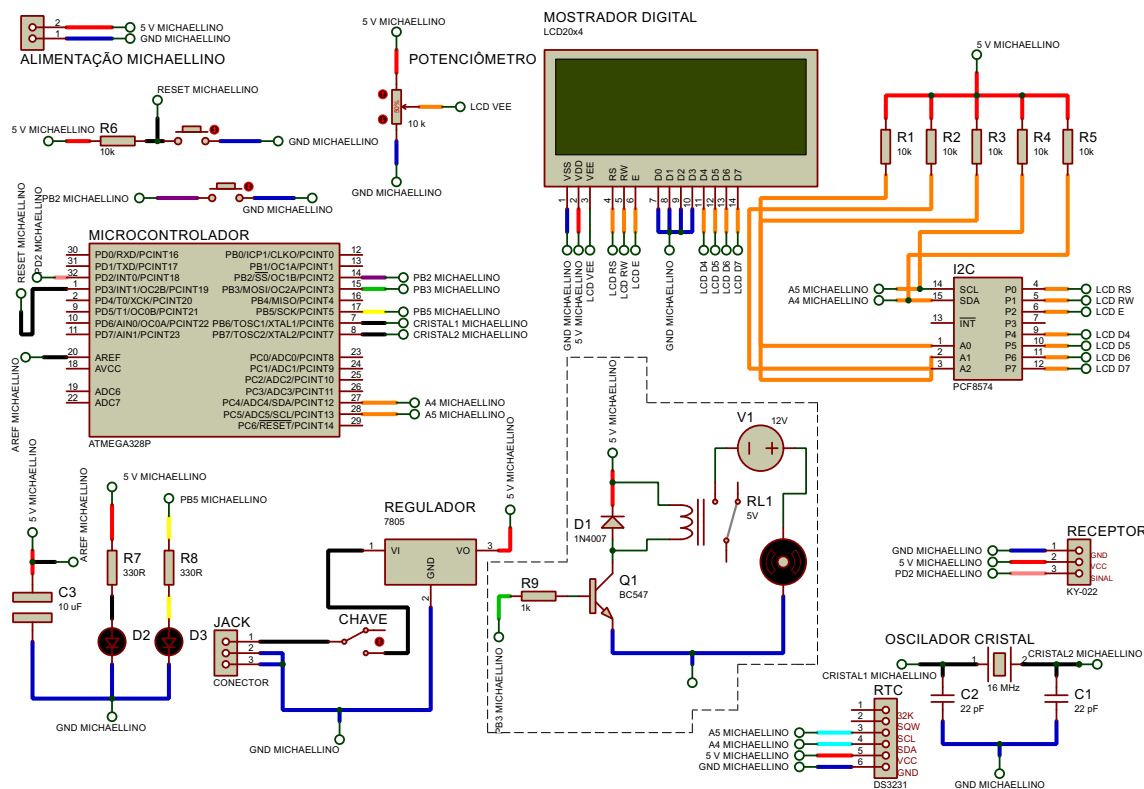
Figura 40 – Circuito para controle do motor.



Fonte: O próprio autor (2022).

A Figura 41, apresenta as conexões feitas para o devido acionamento do motor, onde é necessário um transistor BC547 para controle de um sinal elétrico maior, o motor de 12 V. O relé de 5 V atua como um interruptor do motor.

Figura 41 – Esquemático elétrico da placa Michaellino com o motor 12 V.



Fonte: O próprio autor (2022).

A Tabela 22 mostra as principais características do motor implementado no circuito elétrico. Normalmente os fabricantes costumam divulgar a tensão que deve ser aplicada ao motor para que funcione em uma situação ideal, porém não é uma tensão fixa.

Tabela 22 – Características do motor em corrente contínua AK555/306PL13S500C

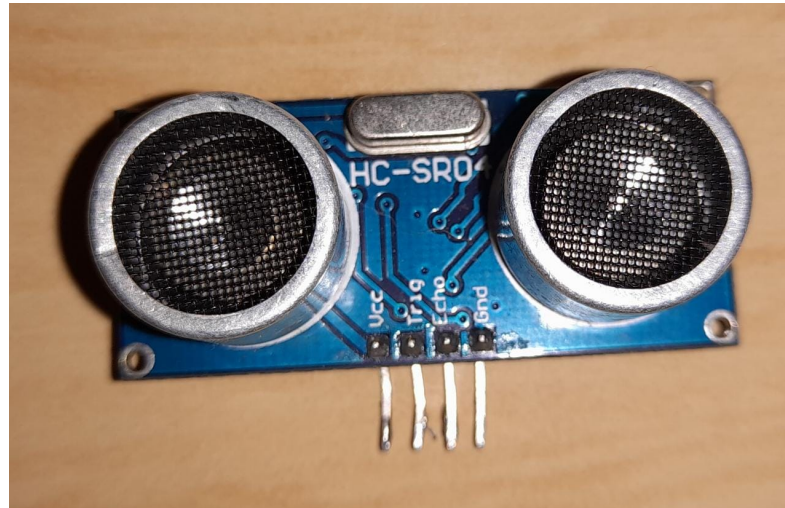
Operação	9 - 16 V
Nominal	12 V
Rotação (sem carga)	6500 rpm
Corrente (sem carga)	360 mA
<b>Máximo rendimento</b>	
Rotação	5300 rpm
Corrente	1,76 A
Torque	306 gf.cm
Potência	16,5 W
Rendimento	78,1 %
Torque de partida	1,7 Kgf.cm

Fonte: Adaptado de Akiyama Motors (2022).

#### 4.2.3.4 Controle do Reservatório

O reservatório possui uma tampa onde está conectado o sensor ultrassônico, HC-SR04 observado na Figura 42.

Figura 42 – Sensor Ultrassônico HC-SR04.



Fonte: O próprio autor (2022).

A Tabela 23 possui as características elétricas do sensor.

Tabela 23 – Características elementares sensor ultrassônico HC-SR04.

Alimentação	5 V
Corrente de operação	2 mA
Frequência de operação	40 Hz
Ângulo de efeito	15°
Alcance	20 à 4000 mm
Precisão	3 mm

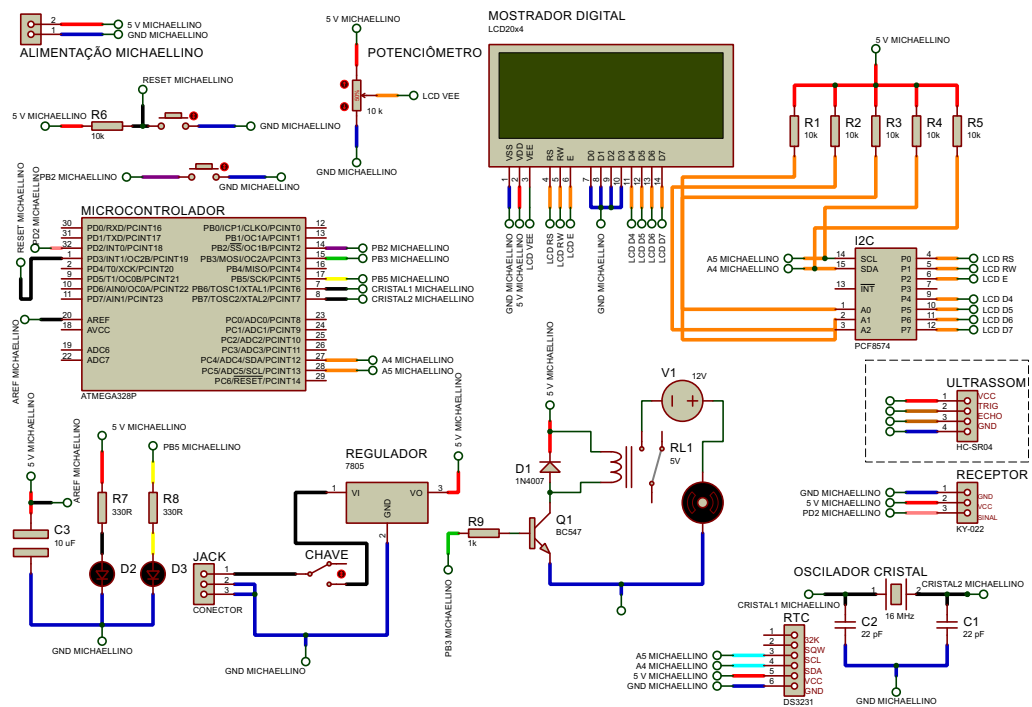
Fonte: Electro Schematics (2022).

Posicionado na tampa do comedouros o sensor será capaz de identificar quando a ração estiver faltando.

O funcionamento do sensor é fundamentado no envio (*trigger*) e recebimento (*echo*) de ondas ultrassônicas. Não há envio de medida alguma para o microcontrolador, apenas indica quando o pulso enviado é recebido. O ATMEGA328 será responsável por medir o tempo e calcular as distâncias, considerando como referência o tempo que a onda ultrassônica atingir um objeto e voltar.

Na Figura 43 o sensor ultrassônico será conectado nos pinos 10 do Arduino (*trigger*) e o pino 9 do Arduino (*echo*).

Figura 43 – Esquemático elétrico da placa Michaellino com sensor ultrassônico.



Fonte: O próprio autor (2022).

### 4.3 DESENVOLVIMENTO DO SOFTWARE

O *software* utilizado para fazer a programação do projeto é o Arduino IDE, versão 1.8.19, onde todas as ações são executadas através de um *sketch* conforme a programação em C++. Uma das grandes vantagens da plataforma está no seu ambiente de desenvolvimento, pois esta linguagem utiliza uma estrutura simples, facilmente implementado.

A programação dentro do *software* é realizada por meio de um programa capaz de comportar todas as ações do projeto, desde ligar o sistema do alimentador até as definições de horários para acionamento do motor para liberação da ração.

A conexão com o computador é feita via cabo USB, permitindo que os comandos definidos no programa sejam devidamente transferidos até a placa. Se o *software* nunca foi utilizado antes, abrirá um *sketch* em branco onde você pode começar programação imediatamente. Primeiro, devemos definir as configurações da placa e da porta para permitir o envio do código (ARDUINO, 2015; FEZARI; DAHOUD, 2018).

A Figura 44 mostra o *software* depois de aberto. Observa-se as seguintes funções: (A) nome do sketch e a versão do Arduino; (B) menu de opções; (C) verificação da programação antes do envio ao Arduino; (D) envio; (E) novo *sketch*; (F) abrir um *sketch*; (G) salvar o *sketch*; (H) guia atual; (I) abrir monitor serial; (J) menu de guias; (K) código; (L) linha selecionada pelo cursor; (M) comunicação porta selecionada pelo USB; (N) execução do código, se houver erro estará discriminado nessa aba e (o) estado do código criado.

Figura 44 – Destaque das funções do Arduino IDE 1.8.19.

```

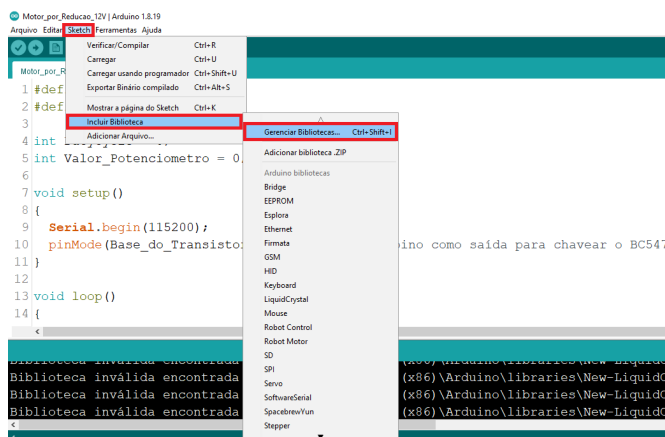
1 #define Pino_Analogico A0
2 #define Base_do_Transistor 3
3
4 int DutyCycle = 0;
5 int Valor_Potenciometro = 0;
6
7 void setup()
8 {
9   Serial.begin(115200);
10  pinMode(Base_do_Transistor,OUTPUT); //Define pino como saída para chavear o BC547
11 }
12
13 void loop()
14 {

```

Fonte: O próprio autor (2022).

Para o correto funcionamento do sensor ultrassônico, RTC DS3231 e mostrador digital é necessário incluir as bibliotecas que corresponde a esses componentes: HCSR04 ultrasonic sensor, versão 2.0.3; LCD\_I2C, versão 2.3.0; DS3231\_RTC, versão 1.1.0 e IRremote, versão 3.9.0. Todas as versões evidenciadas estão disponíveis no repositório do Arduino ou através do próprio software destacados na Figura 45.

Figura 45 – Opção para adicionar as bibliotecas no programa no Arduino.



Fonte: O próprio autor (2022).

Uma particularidade das bibliotecas instaladas é a biblioteca do sensor infravermelho, onde é necessário fazer o mapeamento das teclas que serão acionadas para o funcionamento do alimentador. Seja para definir o alarme que será responsável por acionar o motor que liberará a ração.

A Figura 46 mostra o programa necessário para descobrir o código em decimal ou hexadecimal do controle remoto. Observe que indicado primeiramente a biblioteca IRremote, depois é feita o acionamento do pino responsável pela leitura do sensor.

Conforme as teclas são apertadas apontam em decimal e hexadecimal seu código.

Figura 46 – Mapeamento das teclas do controle remoto KEYES no Arduino IDE.

```

Mapeamento_das_teclas
1 #include <IRremote.h>
2
3 int RECV_PIN = 2;
4 IRrecv irrecv(RECV_PIN);
5 decode_results results;
6
7 void setup() {
8   Serial.begin(9600);
9   irrecv.enableIRIn();
10 }
11
12 void loop() {
13   if (irrecv.decode(&results)) {
14     Serial.print("Codigo HEX: ");
15     Serial.println(results.value, HEX);
16     Serial.print("Codigo DEC: ");
17     Serial.println(results.value);
18     Serial.println("");
19     irrecv.resume();
20   }
21   delay(100);
22 }

```

Fonte: O próprio autor (2022).

Observe que na Figura 47 mostra a comunicação serial entre o sensor. Observe que quando a tecla é apertada rapidamente ocorre um erro, simbolizado pelo código hexadecimal FFFFFFFF e código decimal 4294967295.

Figura 47 – Comunicação serial das teclas do controle remoto KEYES no Arduino IDE.

```

Mapeamento_das_teclas | Arduino 1.8.19
Arquivo Editar Sketch Ferramentas Ajuda

Mapeamento_das_teclas
1 #include <IRremote.h>
2
3 int RECV_PIN = 2;
4 IRrecv irrecv(RECV_PIN);
5 decode_results results;
6
7 void setup() {
8   Serial.begin(9600);
9   irrecv.enableIRIn();
10 }
11
12 void loop() {
13   if (irrecv.decode(&results)) {
14     Serial.print("Codigo HEX: ");
15     Serial.println(results.value, HEX);
16     Serial.print("Codigo DEC: ");
17     Serial.println(results.value);
18     Serial.println("");
19     irrecv.resume();
20   }
21   delay(100);
22 }

COM5
The function decode(&results) is deprecated and may not work as expected! Just use decode() without a parameter and IrReceiver.decodedIR
Codigo HEX: FF9067
Codigo DEC: 16750695

Codigo HEX: FFA25D
Codigo DEC: 16753245

Codigo HEX: FF639D
Codigo DEC: 16736925

Codigo HEX: FFFFFFFF
Codigo DEC: 4294967295

Codigo HEX: FFF21D
Codigo DEC: 16769565

Codigo HEX: FFFFFFFF
Codigo DEC: 4294967295

Codigo HEX: FF21D0
Codigo DEC: 16720605

Codigo HEX: FF63FD
Codigo DEC: 16712445

O sketch usa 5523 bytes (17%) de espaço de armazenamento para programas. O máximo são 32256 bytes.
Bibliotecas globais usam 700 bytes (84%) de memória dinâmica, deixando 1948 bytes para variáveis locais. O máximo são 2048 bytes.
Biblioteca inválida encontrada em C:\Program Files (x86)\Arduino\libraries\New-LiquidCrystal-master; no headers files (.h) found in C:\Program Files (x86)\Arduino\libraries\New-LiquidCrystal-master

Auto-rolagem Show timestamp Nova-linha 9600 velocidade Deleta a saída
Arduino Uno em COM5

```

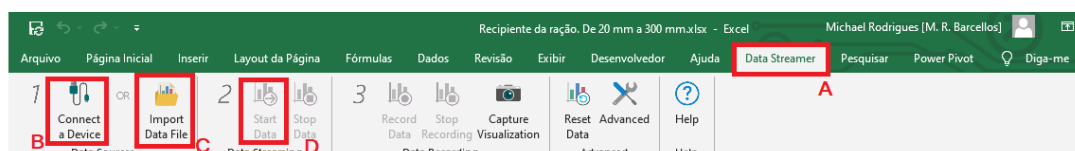
Fonte: O próprio autor (2022).

Para a verificação da quantidade de ração no recipiente foi utilizado o Microsoft Excel pela facilidade de manipulação dos dados. É uma ferramenta útil pois o complemento *Data Streamer* permite importar, visualizar e analisar dados em tempo real de dispositivos externos, como o microcontrolador Arduino. Através desse mecanismo é possível realizar a leitura dos valores que observamos na porta serial do computador, que é a mesma metodologia que os dados são apresentados no monitor serial do Arduino IDE. Isso torna a coleta dos dados dos sensores acessíveis, econômica e fácil de usar para uma ampla

variedade de aplicações (NICHOLS, 2017).

A Figura 48 demonstra como ter acesso a seleção que determina que serão lidos os valores emitidos pelo microcontrolador na porta USB. O menu (A) Data Streamer quando selecionado, podemos conectar o cabo USB no computador (B) ou selecionar um arquivo com dados (C), logo através de uma dessas duas opções é possível visualizar os dados (D).

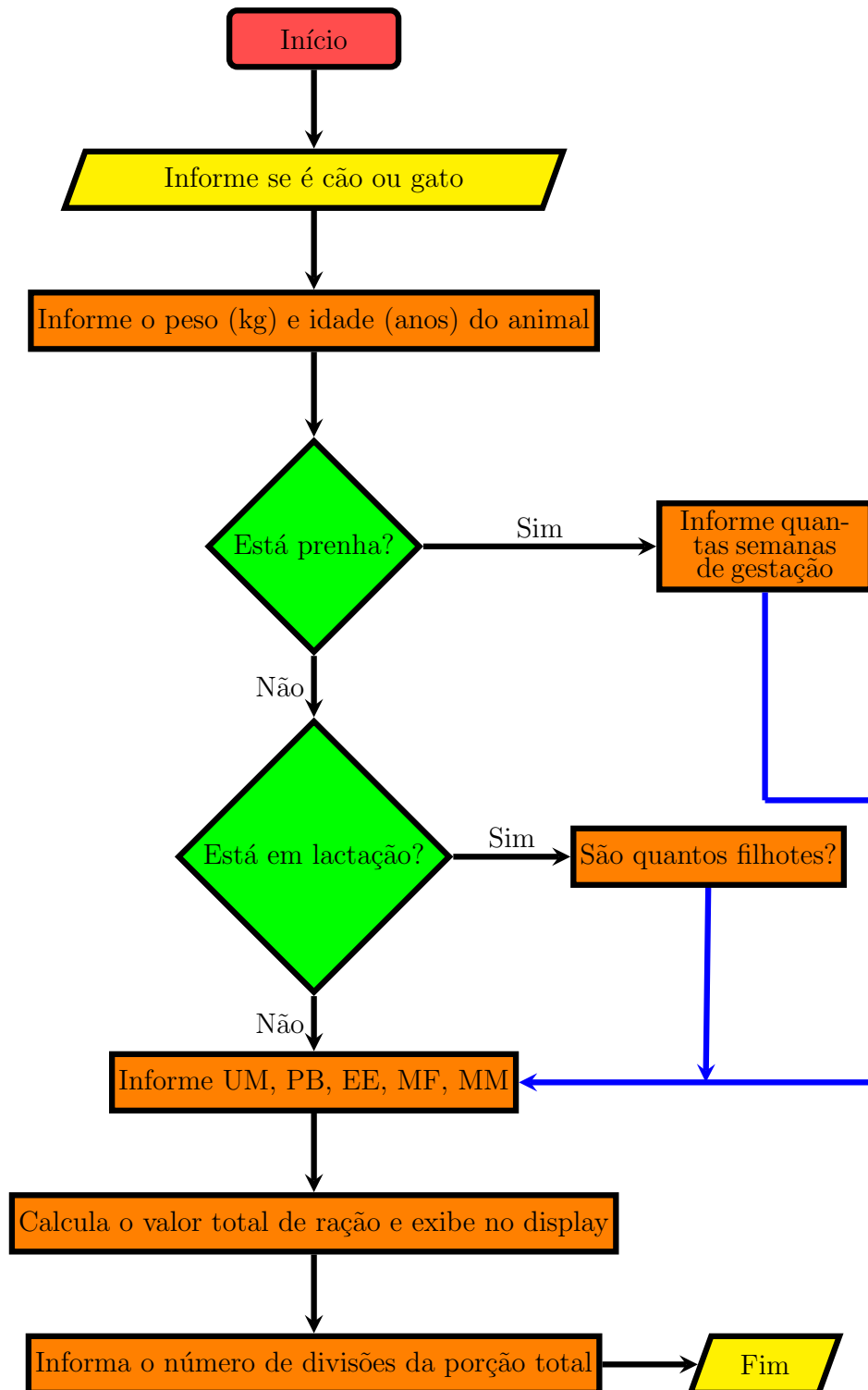
Figura 48 – Acesso aos menus para obter dados em tempo real pelo Excel.



Fonte: O próprio autor (2022).

Para melhorar o entendimento das equações que serão utilizadas no sistema do alimentador, foi construído um fluxograma, observado na Figura 49 que estará presente no manual do equipamento.

Figura 49 – Fluxograma da interface entre o usuário e o alimentador automático



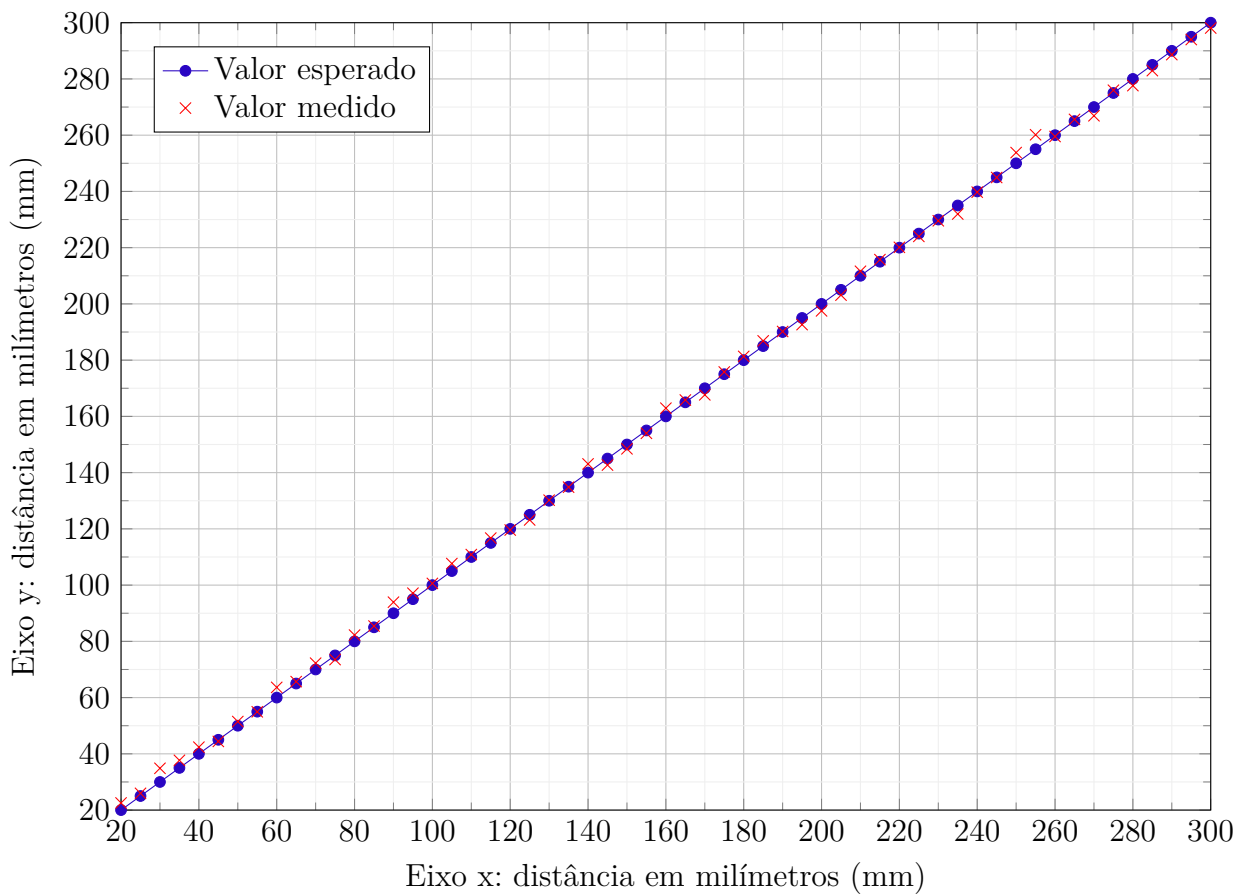
Fonte: O próprio autor (2022).

## 5 ANÁLISE DE RESULTADOS

O projeto do alimentador automático para cães e gatos iniciou com a verificação da calibração do sensor ultrassônico HC-SR04. O ensaio foi realizado de uma distância entre 20 mm até 300 mm, onde o obstáculo a frente do sensor foi o próprio solo.

A Figura 50 mostra a comparação da distância da ração no reservatório do alimentador. Observe que o monitoramento abaixo não inicia em 0 mm pois o sensor não identifica distâncias inferiores a 20 mm.

Figura 50 – Demonstração do valor esperado e medido do sensor ultrassônico.



Fonte: O próprio autor, 2022.

A Tabela 24 mostra-se com mais detalhes aos resultados apresentados na Figura 50, destacando a médias dos valores obtidos.

A Tabela 25 é a continuação da tabela acima. É importante destacar que a Figura 50 apresenta os valores esperados e valores medidos.

A intenção de utilizar um módulo DS3231 foi para disponibilizar o horário em tempo real, onde seja possível gravar até três horários para liberação da ração.

Tabela 24 – Tabela dos pontos indicados no gráfico de 20 à 200 mm.

Distância (mm)	1º Teste	2º Teste	3º Teste	4º Teste	5º Teste	Média
20	21,6	21,61	23,32	24,53	21,61	22,534
25	28,65	25	24,83	22,65	28,65	25,956
30	38,1	38,71	32,5	32,5	32,5	34,862
35	35,52	35	39,4	38,54	40,01	37,694
40	42,25	42,25	41,58	41,79	43,99	42,372
45	44	44,65	44,74	44,56	44,12	44,414
50	51,25	50,69	51,36	51,79	52,33	51,484
55	56,98	55,01	55,05	55,01	52,69	54,948
60	62,35	60,58	63,4	69,5	62,35	63,636
65	70,65	65,12	65,12	63,58	63,94	65,682
70	70,1	70,15	76,5	72,23	72,23	72,242
75	73,56	74,89	74,89	72	72,25	73,518
80	80,25	81,27	83,52	83,95	82,15	82,228
85	84,95	84,12	83,5	85,69	88,95	85,442
90	96,1	96,1	88,9	94	94,56	93,932
95	96,78	96,82	98,5	97,06	96,45	97,122
100	103,51	101,98	97,45	103,51	96,58	100,606
105	104,92	105,74	108,1	109,57	110	107,666
110	116,1	111,53	109,85	109,85	107	110,866
115	113,95	117,2	117,2	117,65	117,6	116,72
120	118,25	123,74	118,7	119,32	117,95	119,592
125	121,5	127,89	121,52	123,45	121,5	123,172
130	127,56	129,9	132,5	133,5	127,56	130,204
135	136,87	134,5	132,68	134,5	135,64	134,838
140	148,68	141,23	141,89	145,1	138,6	143,1
145	141,15	144,25	141,9	141,9	144,25	142,69
150	150,01	149,89	148,8	147,25	146,38	148,466
155	153,69	152,13	151,98	155,36	156,8	153,992
160	160,05	162,1	162,1	162,56	167,8	162,922
165	166,12	163,68	167,8	166,41	165,09	165,82
170	169,52	168,36	165,29	165,47	169,7	167,668
175	174,1	174,1	174,1	178,05	178,6	175,79
180	180,25	180,35	183,5	180,01	182,56	181,334
185	188,5	186,23	186,24	186,65	186,65	186,854
190	190,3	190,3	191,45	189,5	189,41	190,192
195	195,05	194,03	193,25	190,98	189,85	192,632
200	199,5	197,36	197,36	198,9	194,5	197,524

Fonte: O próprio autor, 2022.

A implementação da entrada do dados, a interface do usuário está em desenvolvimento para que haja a utilização do sensor sensor infravermelho para manipular hora e minuto que serão disponibilizados as comidas do cão ou gato. Uma alternativa que também é implementada no projeto é a utilização do terminal de dados do próprio software Arduino

Tabela 25 – Tabela dos pontos indicados no gráfico de 200 à 300 mm

Distância (mm)	1º Teste	2º Teste	3º Teste	4º Teste	5º Teste	Média
200	199,5	197,36	197,36	198,9	194,5	197,524
205	205,14	203,15	201,15	202,65	203,48	203,114
210	216,05	211,5	213,05	209,52	207,9	211,604
215	218,09	218,09	215,64	214,53	212,39	215,748
220	219,5	218,98	223,65	219,5	218,98	220,122
225	223,68	228,79	221,5	223,49	222,56	224,004
230	224,51	236,5	224,51	224,51	237,8	229,566
235	231,48	233,65	232,22	232,22	230,25	231,964
240	239,65	239,21	237,8	240,56	241,48	239,74
245	241,4	241,4	248,5	247,1	246,35	244,95
250	250,5	251,26	253,5	256,91	256,91	253,816
255	259,64	260,01	259,64	259,64	261,78	260,142
260	260,1	260,89	260,51	258,45	257,69	259,528
265	263,51	263,78	263,98	268,41	268,41	265,618
270	267,8	267,8	265,01	268,9	265,01	266,904
275	271,5	275,08	275,95	279,25	278,14	275,984
280	278,63	277,1	278,63	278,63	275,12	277,622
285	283,15	280,02	288,12	283,18	280,3	282,954
290	288,25	289,5	285,01	289,9	290,56	288,644
295	290,1	295,15	295,15	295,15	294,48	294,006
300	297,15	297,25	298,54	299,56	298,01	298,102

Fonte: O próprio autor, 2022.

IDE.

A Figura 51 mostra a liberação da ração pelo eixo helicoidal, isso ocorreu enquanto existia um motor 12 V, em tensões menores não é possível fazer o transporte de ração pelo compartimento criado. Foi testado um motor com redução de 6 V mas não houve eficácia neste processo.

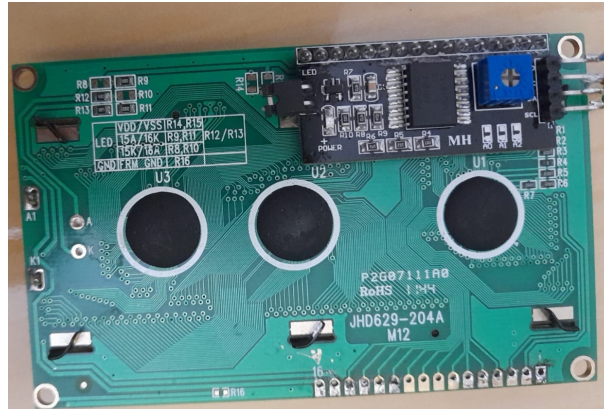
Figura 51 – Resultado final do recipiente do alimentador



Fonte: O próprio autor, 2022.

A Figura 52 mostra que o módulo I2C foi soldado no próprio LCD afim de reduzir o número de ligações.

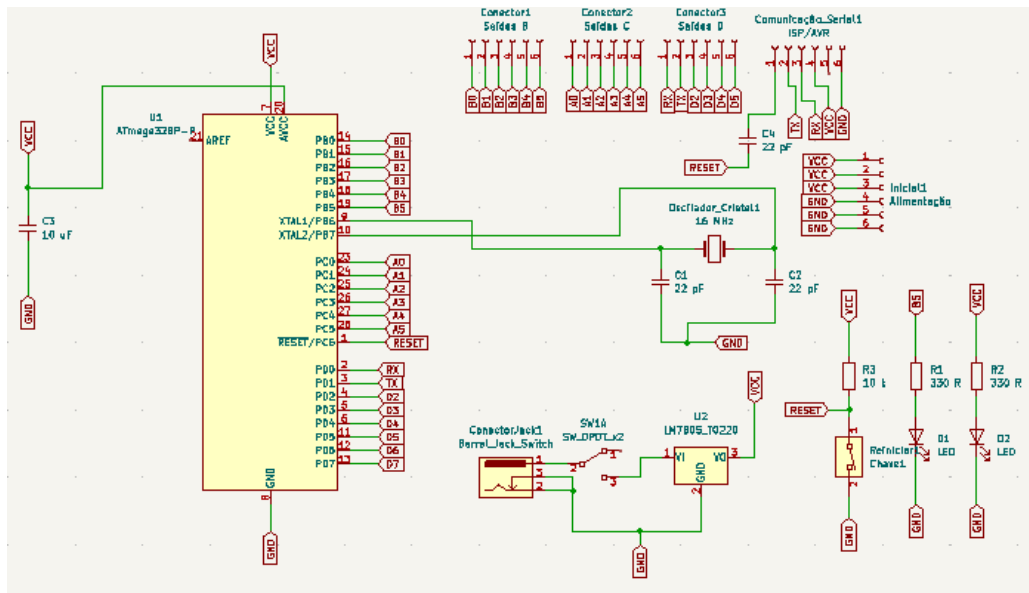
Figura 52 – Mostrador digital soldado ao módulo I2C



Fonte: O próprio autor, 2022.

Foi efetuado uma placa de desenvolvimento para o processamento de dados: Michaellino versão 1.0, assim será substituído pelo Arduino UNO. Esta placa foi desenvolvida através do KiCad versão 6.0. Na Figura 53 podemos observar o esquemático completo da placa, que será alimentado por um conector *Jack*, por uma fonte externa de 9 V, onde há um regulador tensão, oscilador de 16 MHz, um botão para reiniciar o sistema, LEDs indicadores de pinos RX, TX e o momento de gravação do código a ser lido pelo microcontrolador ATmega328P.

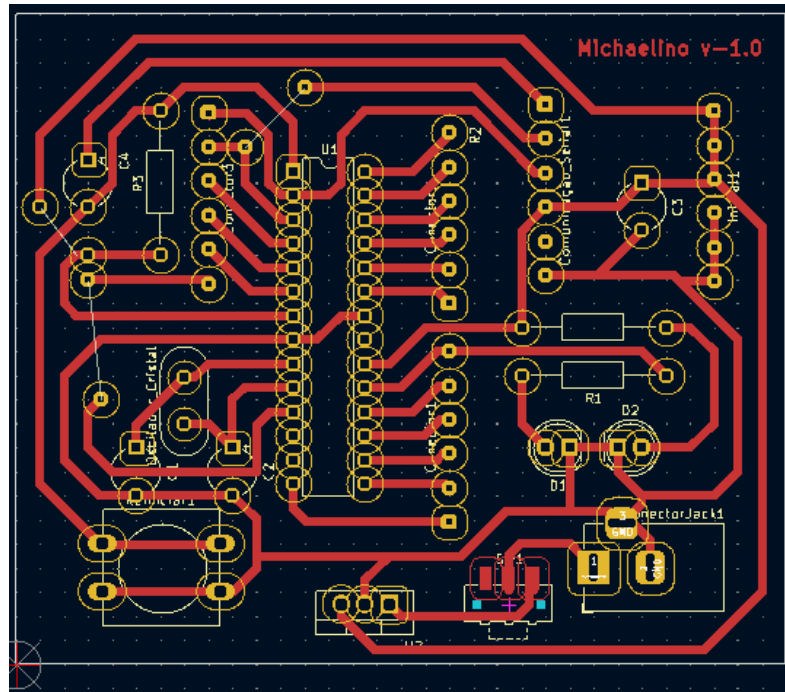
Figura 53 – Esquemático da placa Michaellino.



Fonte: O próprio autor, 2022.

A Figura 54 contém a disposição dos componentes na placa. Como o circuito será impresso na Fresadora numa face única, para não haver ligações cruzadas nas trilhas houve a necessidade de inclusão de três *jumpers*.

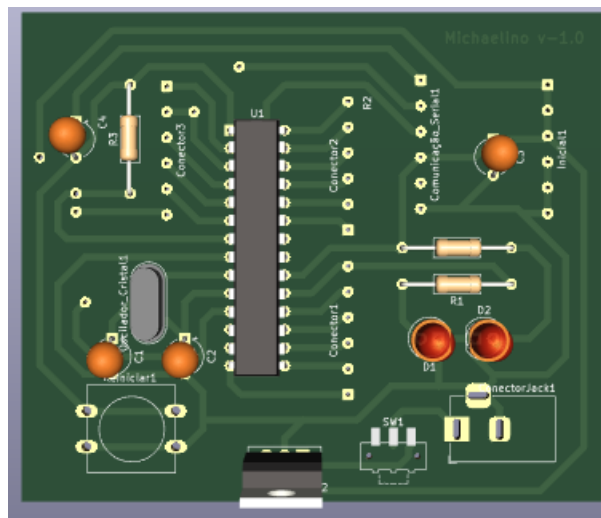
Figura 54 – PCB Michaelino.



Fonte: O próprio autor, 2022.

Por fim, a Figura 55 mostra como ficará a placa na sua finalização. Observa-se que existem 3 objetos que não foram adicionados a visualização 3D da placa porque KiCAD não possui estes modelos que foram utilizados no projeto. Então, fez-se as medidas do conector Jack, o botão de reinício e de uma chave de três posições para adicioná-los manualmente ao projeto da placa.

Figura 55 – Visualização 3D da placa Michaelino.



Fonte: O próprio autor, 2022.

## 6 CONCLUSÕES

O objetivo central era criar um alimentador temporizável que foi atingido.

O primeiro recipiente do alimentador observado teria 140 mm de comprimento, 110 mm de largura e 100 mm de altura, impresso pelo filamento PLA, teria 210,53 g, o que resultaria em um processo de 12,5 horas. Não havendo a possibilidade de monitorar a impressora durante todo o processo, foi realizado a redução de tamanho do comedouro. Diminuindo em  $\pm 50\%$  temos 70 mm de comprimento, 55 mm de largura e 50 mm de altura, que levou 7,5 horas para ser impresso na impressora 3D e resultou numa vasilha de 40,93 g.

A placa Michaelino versão 1.0 foi idealizada para controle de pinos do ATmega328P. É possível notar que a placa não possui conexão para o USB, ou seja, a gravação dos programas utilizados no projeto ocorre através de um Arduino UNO.

O fluxograma será uma parte importante no manual do usuário para que haja a compreensão de como é executado cada entrada no sistema.

O transporte da ração foi realizado pelo eixo helicoidal, onde ele é movimentado através de um motor de corrente contínua.

A célula de carga de 50 kg não funcionou da maneira esperada, apresentou mais 80% de incerteza quando foi realizado a verificação da calibração, o que tornou a fragmentação impossível de ser realizada.

O módulo RTC que executa o relógio digital no projeto não funcionou da maneira esperada porque cada vez que era desligado o sistema, a hora não era salva. Para solucionar isso foi utilizado a temporização do próprio microcontrolador, o que gerou uma precisão de horário razoável.

### 6.1 MELHORIAS FUTURAS

Como era primordial o baixo custo do projeto não foi adicionado nenhum outro componente muito custoso, como por exemplo, o microcontrolador ESP32, que com o seu recurso de WI-FI embutido poderia ser utilizado como monitoramento de quantidade de ração e assim poderia haver a exclusão do sensor ultrassônico. O recipiente do alimentador poderia ser aumentado para comportar até 5 kg de ração, pois como o alimentador é vedado com uma tampa é possível manter a umidade necessária para a comida.

# REFERÊNCIAS

- ABINPET. 2019 - *Mercado Pet no Brasil*. 2019. Disponível em: <[http://abinpet.org.br/wp-content/uploads/2019/10/abinpet\\_folder\\_2019\\_draft5.pdf](http://abinpet.org.br/wp-content/uploads/2019/10/abinpet_folder_2019_draft5.pdf)>. Citado na página 20.
- ARDUINO, S. A. Arduino. *Arduino LLC*, v. 372, 2015. Citado na página 54.
- ATWATER, W. O. *Principles of nutrition and nutritive value of food*. [S.l.]: US Department of Agriculture, 1910. Citado na página 25.
- BARROS, M. J. d. S. et al. Estudo retrospectivo da casuística de cães e gatos em fase pediátrica atendidos no hospital veterinário universitário prof. ivon macêdo tabosa-universidade federal de campina grande. Universidade Federal de Campina Grande, 2019. Citado na página 20.
- CARCIOFI, A. C. Métodos para estudo das respostas metabólicas de cães e gatos a diferentes alimentos. *Revista Brasileira de Zootecnia*, SciELO Brasil, v. 36, p. 235–249, 2007. Citado na página 22.
- CARCIOFI, A. C.; JEREMIAS, J. T. Progresso científico sobre nutrição de animais de companhia na primeira década do século xxi. *Revista Brasileira de Zootecnia*, SciELO Brasil, v. 39, p. 35–41, 2010. Citado 2 vezes nas páginas 16 e 22.
- CARVALHO, G. L. C. de. *Obesidade de Cães e Gatos*. 2014. Disponível em: <<https://www.ufrgs.br/lacvet/site/wp-content/uploads/2014/11/obesidade.pdf>>. Citado na página 16.
- CASE, L. P. et al. *Canine and feline nutrition: a resource for companion animal professionals*. [S.l.]: Elsevier Health Sciences, 2010. Citado na página 22.
- CHAVES, M. S. Neonatologia em cães e gatos: Aspectos relevantes da fisiologia e patologia-revisão de literatura e relato de caso de diprosopo tetraoftalmo. Universidade Federal de Minas Gerais, 2011. Citado na página 19.
- COUNCIL, N. R. et al. *Nutrient requirements of dogs and cats*. [S.l.]: National Academies Press, 2006. Citado 3 vezes nas páginas 16, 23 e 24.
- FALLEIRO, F. D. *Conversor Analógico-digital com capacitores mínimos integrado na tecnologia CMOS*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2015. Citado na página 48.
- FEZARI, M.; DAHOUD, A. A. Integrated development environment “ide” for arduino. *WSN applications*, ResearchGate, p. 1–12, 2018. Citado 2 vezes nas páginas 35 e 54.
- MAXIM, D. S. *Extremely Accurate I2C-Integrated RTC/TCX0/Crystal*. 2006. Disponível em: <<https://web.wpi.edu/Pubs/E-project/Available/E-project-010908-124414/unrestricted/DS3231-DS3231S.pdf>>. Citado 2 vezes nas páginas 48 e 49.
- NETO, C. D. et al. Desenvolvimento de uma plataforma de força em pedal de ciclismo. *Revista Brasileira de Biomecânica*, v. 2, n. 3, p. 39–44, 2001. Citado na página 46.

- NICHOLS, D. Arduino-based data acquisition into excel, labview, and matlab. *The Physics Teacher*, American Association of Physics Teachers, v. 55, n. 4, p. 226–227, 2017. Citado na página 57.
- OGOSHI, R. C. S. et al. Conceitos básicos sobre nutrição e alimentação de cães e gatos. *Ciência Animal*, v. 25, n. 1, p. 64–75, 2015. Citado 7 vezes nas páginas 15, 16, 22, 23, 24, 25 e 27.
- REIS, J. *Desenvolvimento e projeto de impressora DMF para impressão multicolor*. Tese (Doutorado) — tese de mestrado, Aveiro, 2016. Citado na página 32.
- SACRAMENTO, R. Transportadores contínuos para granéis sólidos. *Apostila UFBA Universidade Federal da Bahia*, p. 341–349, 2015. Citado na página 34.
- SILVA, D. P. d. *Canis familiaris: aspectos da domesticação (origem, conceitos, hipóteses)*. 2011. Citado na página 20.
- THOMAS, K. *O homem e o mundo natural: mudanças de atitude em relação as plantas e aos animais (1500-1800)*. Companhia de Bolso, 2010. ISBN 9788535915976. Disponível em: <<https://books.google.com.br/books?id=1DIItQwAACAAJ>>. Citado na página 19.
- TORGA, D. S. *Desenvolvimento de uma plataforma didática para práticas de controle de velocidade de motor de corrente continua*. 2016. Citado na página 50.
- VIEIRA, M. A. R. *Sensor inteligente para medição de cargas mecânicas*. Tese (Doutorado) — Instituto Politecnico de Braganca (Portugal), 2016. Citado na página 46.
- ZUCCA, R. et al. *Desenvolvimento de impressora 3d de baixo custo para prototipagem de peças para o meio rural*. Universidade Federal da Grande Dourados, 2019. Citado na página 32.

# A CÓDIGO DO ARQUIVO RELOGIO.INO.

```

#define short_time      1400
#define med_time       2400
#define long_time      4000

//Incluindo a biblioteca LCD + I2C
#include <LiquidCrystal_I2C.h>
// Incluindo biblioteca Wire
#include <Wire.h>

LiquidCrystal_I2C lcd(0x27,  2,1,0,4,5,6,7,3, POSITIVE);

const int pino_do_alarme = 12;
const int buzzer = 13;
void setup() {
    pinMode(pino_do_alarme, OUTPUT);
    pinMode(buzzer, OUTPUT);
    digitalWrite(pino_do_alarme, LOW);
    lcd.begin(20,4);
    lcd.backlight();
    // Configuracao do Modulo Timer1
    TCCR1A = 0;
    TCCR1B = 0;
    TCNT1 = 0;
    TIMSK1 = 1;
    Wire.begin();
    attachInterrupt(1, RC5_read, CHANGE);
    attachInterrupt(0, Alarme, FALLING);
}
// Variaveis do Controle Remoto
boolean rc5_ok = 0, toggle, last_toggle;
byte rc5_state = 0, j;
unsigned int rc5_code;
// Variaveis do RTC
bool alarme1_status, alarme2_status;
char Time[] = " : : ",

```

```

    calendario [] = "      / /20  ",
    alarme1 [] = "A1:      : :00", alarme2 [] = "A2:      : :00",
    temperatura [] = "T:      .      C";
byte  i, segundo, minuto, hora, dia, data, mes, ano,
      minuto_do_alarme1, hora_do_alarme1, minuto_do_alarme2, hora_do_alar
      status_reg;

void RC5_read() {
unsigned int timer_value;
  if(rc5_state != 0){
    timer_value = TCNT1;
    TCNT1 = 0;
  }
  switch(rc5_state){
  case 0 :
    TCNT1 = 0;
    TCCR1B = 2;
    rc5_state = 1;
    j = 0;
    return;
  case 1 :
    if((timer_value > long_time) || (timer_value < short_time)){
      rc5_state = 0;
      TCCR1B = 0;
      return;
    }
    bitSet(rc5_code, 13 - j);
    j++;
    if(j > 13){
      rc5_ok = 1;
      toggle = bitRead(rc5_code, 11);
      rc5_code &= 0x07FF;
      detachInterrupt(1);
      return;
    }
    if(timer_value > med_time){
      rc5_state = 2;
      if(j == 13){
        rc5_ok = 1;

```

```
        bitClear(rc5_code, 0);
        toggle = bitRead(rc5_code, 11);
        rc5_code &= 0x07FF;
        detachInterrupt(1);
        return;
    }
}
else
    rc5_state = 3;
return;
case 2 :
    if((timer_value > long_time) || (timer_value < short_time)){
        rc5_state = 0;
        TCCR1B = 0;
        return;
    }
    bitClear(rc5_code, 13 - j);
    j++;
    if(timer_value > med_time)
        rc5_state = 1;
    else
        rc5_state = 4;
    return;
case 3 :
    if((timer_value > med_time) || (timer_value < short_time)){
        TCCR1B = 0;
        rc5_state = 0;
        return;
    }
    else
        rc5_state = 1;
    return;
case 4 :
    if((timer_value > med_time) || (timer_value < short_time)){
        TCCR1B = 0;
        rc5_state = 0;
        return;
    }
    else
```

```
        rc5_state = 2;
    if(j == 13){
        rc5_ok = 1;
        bitClear(rc5_code, 0);
        toggle = bitRead(rc5_code, 11);
        rc5_code &= 0x07FF;
        detachInterrupt(1);
    }
}
ISR(TIMER1_OVF_vect) {
    rc5_state = 0;
    TCCR1B = 0;
}

void Alarme(){
    digitalWrite(pino_do_alarme, HIGH);
    tone(buzzer, 1000);
}

void DS3231_read(){
    Wire.beginTransmission(0x68);
    Wire.write(0);
    Wire.endTransmission(false);
    Wire.requestFrom(0x68, 7);
    segundo = Wire.read();
    minuto = Wire.read();
    hora    = Wire.read();
    dia     = Wire.read();
    data    = Wire.read();
    mes     = Wire.read();
    ano     = Wire.read();
}

void alarmes_lidos_no_display(){
    byte control_reg, temperatura_lsb;
    char temperatura_msb;
    Wire.beginTransmission(0x68);
    Wire.write(0x08);
    Wire.endTransmission(false);
    Wire.requestFrom(0x68, 11);
```

```

minuto_do_alarme1 = Wire.read();
hora_do_alarme1 = Wire.read();
Wire.read();
minuto_do_alarme2 = Wire.read();
hora_do_alarme2 = Wire.read();
Wire.read();
control_reg = Wire.read();
status_reg = Wire.read();
Wire.read();
temperatura_msb = Wire.read();
temperatura_lsb = Wire.read();
// Converte BCD para decimal
minuto_do_alarme1 = (minuto_do_alarme1 >> 4) * 10 + (minuto_do_alarme1 & 0xf);
hora_do_alarme1 = (hora_do_alarme1 >> 4) * 10 + (hora_do_alarme1 & 0xf);
minuto_do_alarme2 = (minuto_do_alarme2 >> 4) * 10 + (minuto_do_alarme2 & 0xf);
hora_do_alarme2 = (hora_do_alarme2 >> 4) * 10 + (hora_do_alarme2 & 0xf);
// Fim da conversão
alarme1[8] = minuto_do_alarme1 % 10 + 48;
alarme1[7] = minuto_do_alarme1 / 10 + 48;
alarme1[5] = hora_do_alarme1 % 10 + 48;
alarme1[4] = hora_do_alarme1 / 10 + 48;
alarme2[8] = minuto_do_alarme2 % 10 + 48;
alarme2[7] = minuto_do_alarme2 / 10 + 48;
alarme2[5] = hora_do_alarme2 % 10 + 48;
alarme2[4] = hora_do_alarme2 / 10 + 48;
alarme1_status = bitRead(control_reg, 0);
alarme2_status = bitRead(control_reg, 1);
if(temperatura_msb < 0){
    temperatura_msb = abs(temperatura_msb);
    temperatura[2] = '-';
}
else
    temperatura[2] = ' ';
temperatura_lsb >>= 6;
temperatura[4] = temperatura_msb % 10 + 48;
temperatura[3] = temperatura_msb / 10 + 48;
if(temperatura_lsb == 0 || temperatura_lsb == 2){
    temperatura[7] = '0';
    if(temperatura_lsb == 0) temperatura[6] = '0';
}

```

```
    else                temperatura[6] = '5';
}
if(temperatura_lsb == 1 || temperatura_lsb == 3){
    temperatura[7] = '5';
    if(temperatura_lsb == 1) temperatura[6] = '2';
    else                temperatura[6] = '7';
}
temperatura[8] = 223;
lcd.setCursor(10, 0);
lcd.print(temperatura);
lcd.setCursor(0, 2);
lcd.print(alarme1);
lcd.setCursor(17, 2);
if(alarme1_status) lcd.print("LIG ");
else                lcd.print("DES");
lcd.setCursor(0, 3);
lcd.print(alarme2);
lcd.setCursor(17, 3);
if(alarme2_status) lcd.print("LIG ");
else                lcd.print("DES");
}
void calendario_display(){
    switch(dia){
        case 1: strcpy(calendario, "Dom   /   /20  "); break;
        case 2: strcpy(calendario, "Seg   /   /20  "); break;
        case 3: strcpy(calendario, "Ter   /   /20  "); break;
        case 4: strcpy(calendario, "Qua   /   /20  "); break;
        case 5: strcpy(calendario, "Qui   /   /20  "); break;
        case 6: strcpy(calendario, "Sex   /   /20  "); break;
        case 7: strcpy(calendario, "Sab   /   /20  "); break;
        default: strcpy(calendario, "Sab   /   /20  ");
    }
    calendario[13] = ano % 10 + 48;
    calendario[12] = ano / 10 + 48;
    calendario[8] = mes % 10 + 48;
    calendario[7] = mes / 10 + 48;
    calendario[5] = data % 10 + 48;
    calendario[4] = data / 10 + 48;
    lcd.setCursor(0, 1);
```

```

    lcd.print(calendario);
}
void DS3231_display(){
    // Convert BCD to decimal
    segundo = (segundo >> 4) * 10 + (segundo & 0x0F);
    minuto = (minuto >> 4) * 10 + (minuto & 0x0F);
    hora = (hora >> 4) * 10 + (hora & 0x0F);
    data = (data >> 4) * 10 + (data & 0x0F);
    mes = (mes >> 4) * 10 + (mes & 0x0F);
    ano = (ano >> 4) * 10 + (ano & 0x0F);
    // End conversion
    Time[7]      = segundo % 10  + 48;
    Time[6]      = segundo / 10  + 48;
    Time[4]      = minuto % 10  + 48;
    Time[3]      = minuto / 10  + 48;
    Time[1]      = hora   % 10  + 48;
    Time[0]      = hora   / 10  + 48;
    calendario_display();
    lcd.setCursor(0, 0);
    lcd.print(Time);
}
void Blink(){
    byte k = 0;
    while ((!rc5_ok || (rc5_code != 0x10) && (rc5_code != 0x11) && (rc5_code
        k++;
        delay(25);
    }
}
byte edit(byte x, byte y, byte parameter){
    char text[3];
    rc5_reset();
    while(true){
        if(rc5_ok && (rc5_code == 0x20 || rc5_code == 0x21)){
            if(rc5_code == 0x20){
                parameter++;
                if(((i == 0) || (i == 5)) && parameter > 23)
                    parameter = 0;
                if(((i == 1) || (i == 6)) && parameter > 59)
                    parameter = 0;
            }
        }
    }
}

```

```
    if(i == 2 && parameter > 31)
        parameter = 1;
    if(i == 3 && parameter > 12) // Se mes >
        parameter = 1;
    if(i == 4 && parameter > 99)
        parameter = 0;
    if(i == 7 && parameter > 1)
        parameter = 0;
}
if(rc5_code == 0x21){
    if(((i == 0) || (i == 5)) && parameter < 1)
        parameter = 24;
    if(((i == 1) || (i == 6)) && parameter < 1)
        parameter = 60;
    if(i == 2 && parameter < 2)
        parameter = 32;
    if(i == 3 && parameter < 2)
        parameter = 13;
    if(i == 4 && parameter < 1)
        parameter = 100;
    if(i == 7 && parameter < 1)
        parameter = 2;
    parameter--;
}
lcd.setCursor(x, y);
if(i == 7){
    if(parameter == 1) lcd.print("LIG ");
    else                lcd.print("DES");
}
else{
    sprintf(text,"%02u", parameter);
    lcd.print(text);
}
}
if(rc5_ok){
    delay(200);
    rc5_reset();
}
lcd.setCursor(x, y);
```

```

    lcd.print(" ");
    if(i == 7) lcd.print(" ");
    Blink();
    lcd.setCursor(x, y);
    if(i == 7){
        if(parameter == 1) lcd.print("LIG ");
        else                lcd.print("DES");
    }
    else{
        sprintf(text,"%02u", parameter);
        lcd.print(text);
    }
    Blink();
    if(i >= 5){
        DS3231_read();
        DS3231_display();
    }
    if(rc5_ok && last_toggle != toggle && ((rc5_code == 0x10 && i < 5) ||
        i++;
        return parameter;
    }
}
}
void rc5_reset(){
    rc5_ok = 0;
    rc5_state = 0;
    last_toggle = toggle;
    attachInterrupt(1, RC5_read, CHANGE);
}

void loop() {
    if(rc5_ok){
        if(rc5_code == 0x10){
            i = 0;
            hora = edit(0, 0, hora);
            minuto = edit(3, 0, minuto);
            rc5_reset();
            while(true){
                if(rc5_ok && (rc5_code == 0x20)){

```

```

        dia++;
        if(dia > 7) dia = 1;
        calendario_display();
        lcd.setCursor(0, 1);
        lcd.print(calendario);
    }
    if(rc5_ok){
        delay(200);
        rc5_reset();
    }
    lcd.setCursor(0, 1);
    lcd.print(" ");
    Blink();
    lcd.setCursor(0, 1);
    lcd.print(calendario);
    Blink();
    if(rc5_ok && last_toggle != toggle && (rc5_code == 0x10))
        break;
}
data = edit(4, 1, data);
mes = edit(7, 1, mes);
ano = edit(12, 1, ano);
// Converte decimal para BCD
minuto = ((minuto / 10) << 4) + (minuto % 10);
hora = ((hora / 10) << 4) + (hora % 10);
data = ((data / 10) << 4) + (data % 10);
mes = ((mes / 10) << 4) + (mes % 10);
ano = ((ano / 10) << 4) + (ano % 10);
// Fim da conversão
// Gravar dados de tempo e calendário no RTC
Wire.beginTransmission(0x68);
Wire.write(0);
Wire.write(0);
Wire.write(minuto);
Wire.write(hora);
Wire.write(dia);
Wire.write(data);
Wire.write(mes);
Wire.write(ano);

```

```

    Wire.endTransmission();
}
if(rc5_code == 0x11){
    i = 5;
    hora_do_alarme1 = edit(4, 2, hora_do_alarme1);
    minuto_do_alarme1 = edit(7, 2, minuto_do_alarme1);
    alarme1_status = edit(17, 2, alarme1_status);
    i = 5;
    hora_do_alarme2 = edit(4, 3, hora_do_alarme2);
    minuto_do_alarme2 = edit(7, 3, minuto_do_alarme2);
    alarme2_status = edit(17, 3, alarme2_status);
    minuto_do_alarme1 = ((minuto_do_alarme1 / 10) << 4) + (minuto_do_alarme1 % 10);
    hora_do_alarme1 = ((hora_do_alarme1 / 10) << 4) + (hora_do_alarme1 % 10);
    minuto_do_alarme2 = ((minuto_do_alarme2 / 10) << 4) + (minuto_do_alarme2 % 10);
    hora_do_alarme2 = ((hora_do_alarme2 / 10) << 4) + (hora_do_alarme2 % 10);
    // Gravar dados dos alarmes
    Wire.beginTransmission(0x68);
    Wire.write(7);
    Wire.write(0);
    Wire.write(minuto_do_alarme1);
    Wire.write(hora_do_alarme1);
    Wire.write(0x80);
    Wire.write(minuto_do_alarme2);
    Wire.write(hora_do_alarme2);
    Wire.write(0x80);
    Wire.write(4 | alarme1_status | (alarme2_status << 1));
    Wire.write(0);
    Wire.endTransmission();
}
if(rc5_code == 0x0C && digitalRead(pino_do_alarme)){
    digitalWrite(pino_do_alarme, LOW);
    noTone(buzzer);

    Wire.beginTransmission(0x68);
    Wire.write(0x0E);
    // Gravar dados no registro de controle
    Wire.write(4 | (!bitRead(status_reg, 0) & alarme1_status) | (!!bitRead(status_reg, 1) & alarme2_status)));
    Wire.write(0);
    Wire.endTransmission();
}
}

```

```
    }  
    rc5_reset ();  
  }  
  DS3231_read ();  
  alarmes_lidos_no_display ();  
  DS3231_display ();  
  delay (50);  
}
```

# B CÓDIGO DO ARQUIVO LIQUIDCRYSTALI2C.H.

```
#ifndef FDB_LIQUID_CRYSTAL_I2C_H
#define FDB_LIQUID_CRYSTAL_I2C_H

#include <inttypes.h>
#include <Print.h>

#define LCD_CLEARDISPLAY 0x01
#define LCD_RETURNHOME 0x02
#define LCD_ENTRYMODESET 0x04
#define LCD_DISPLAYCONTROL 0x08
#define LCD_CURSORSHIFT 0x10
#define LCD_FUNCTIONSET 0x20
#define LCD_SETCGRAMADDR 0x40
#define LCD_SETDDRAMADDR 0x80

#define LCD_ENTRYRIGHT 0x00
#define LCD_ENTRYLEFT 0x02
#define LCD_ENTRYSHIFTINCREMENT 0x01
#define LCD_ENTRYSHIFTDECREMENT 0x00

#define LCD_DISPLAYON 0x04
#define LCD_DISPLAYOFF 0x00
#define LCD_CURSORON 0x02
#define LCD_CURSOROFF 0x00
#define LCD_BLINKON 0x01
#define LCD_BLINKOFF 0x00

#define LCD_DISPLAYMOVE 0x08
#define LCD_CURSORMOVE 0x00
#define LCD_MOVERIGHT 0x04
#define LCD_MOVELEFT 0x00

#define LCD_8BITMODE 0x10
```

```
#define LCD_4BITMODE 0x00
#define LCD_2LINE 0x08
#define LCD_1LINE 0x00
#define LCD_5x10DOTS 0x04
#define LCD_5x8DOTS 0x00

#define LCD_BACKLIGHT 0x08
#define LCD_NOBACKLIGHT 0x00

#define En B00000100
#define Rw B00000010
#define Rs B00000001

class LiquidCrystal_I2C : public Print {
public:

    LiquidCrystal_I2C(uint8_t lcd_addr, uint8_t lcd_cols, uint8_t lcd_lines);

    void begin();

    void clear();

    void home();

    void noDisplay();

    void display();

    void noBlink();

    void blink();

    void noCursor();

    void cursor();

    void scrollDisplayLeft();
```

```
void scrollDisplayRight ();
void printLeft ();
void printRight ();
void leftToRight ();
void rightToLeft ();
void shiftIncrement ();
void shiftDecrement ();
void noBacklight ();
void backlight ();
bool getBacklight ();
void autoscroll ();
void noAutoscroll ();
void createChar(uint8_t, uint8_t []);
void setCursor(uint8_t, uint8_t);
virtual size_t write(uint8_t);
void command(uint8_t);

inline void blink_on() { blink(); }
inline void blink_off() { noBlink(); }
inline void cursor_on() { cursor(); }
inline void cursor_off() { noCursor(); }

void setBacklight(uint8_t new_val);
void load_custom_character(uint8_t char_num, uint8_t *rows);
void printstr(const char []);

private:
void send(uint8_t, uint8_t);
void write4bits(uint8_t);
void expanderWrite(uint8_t);
void pulseEnable(uint8_t);
uint8_t _addr;
uint8_t _displayfunction;
uint8_t _displaycontrol;
uint8_t _displaymode;
uint8_t _cols;
uint8_t _rows;
uint8_t _charsize;
```

```
        uint8_t _backlightval;  
};  
  
#endif // FDB_LIQUID_CRYSTAL_I2C_H
```

## C CÓDIGO DO ARQUIVO WIRE.H.

```

#ifndef TwoWire_h
#define TwoWire_h

#include <inttypes.h>
#include "Stream.h"

#ifndef I2C_BUFFER_LENGTH

#define BUFFER_LENGTH 128
#define I2C_BUFFER_LENGTH BUFFER_LENGTH
#endif

class TwoWire: public Stream
{
private:
    static uint8_t rxBuffer [];
    static size_t rxBufferIndex;
    static size_t rxBufferLength;

    static uint8_t txAddress;
    static uint8_t txBuffer [];
    static size_t txBufferIndex;
    static size_t txBufferLength;

    static uint8_t transmitting;
    static void (*user_onRequest)(void);
    static void (*user_onReceive)(size_t);
    static void onRequestService(void);
    static void onReceiveService(uint8_t*, size_t);

public:
    TwoWire();
    void begin(int sda, int scl);
    void begin(int sda, int scl, uint8_t address);
    void pins(int sda, int scl) __attribute__((deprecated));

```

```

void    begin ();
void    begin (uint8_t);
void    begin (int);
void    setClock (uint32_t);
void    setClockStretchLimit (uint32_t);
void    beginTransmission (uint8_t);
void    beginTransmission (int);
uint8_t endTransmission (void);
uint8_t endTransmission (uint8_t);
size_t  requestFrom (uint8_t address, size_t size, bool sendStop);
uint8_t status ();

uint8_t requestFrom (uint8_t, uint8_t);
uint8_t requestFrom (uint8_t, uint8_t, uint8_t);
uint8_t requestFrom (int, int);
uint8_t requestFrom (int, int, int);

virtual size_t write (uint8_t);
virtual size_t write (const uint8_t*, size_t);
virtual int    available (void);
virtual int    read (void);
virtual int    peek (void);
virtual void   flush (void);
void          onReceive (void (*)(int));
void          onReceive (void (*)(size_t));
void          onRequest (void (*)(void));

using Print::write;
};

#if !defined(NO_GLOBAL_INSTANCES) && !defined(NO_GLOBAL_TWOWIRE)
extern TwoWire Wire;
#endif

#endif
Footer

```

## D MAPEAMENTO DAS TECLAS DO CONTROLE REMOTO KEYES.

Tecla	Código Decimal	Hexadecimal
00	FF9867	16750695
01	FFA25D	16753245
02	FF629D	16736925
03	FFE21D	16769565
04	FF22DD	16720605
05	FF02FD	16712445
06	FFC23D	16761405
07	FFE01F	16769055
08	FFA857	16754775
09	FF906F	16748655
Asterisco	FF6897	16738455
Cerquilha	FFB04F	16756815
Cima	FF18E7	16718055
Baixo	FF4AB5	16730805
Direita	FF5AA5	16734885
Esquerda	FF10EF	16716015
OK	FF38C7	16726215

# E CONFIGURAÇÃO DO ARQUIVO CONTROL.C.

```
#include <IRremote.h>

const int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup() {
  Serial.begin(9600);
  irrecv.enableIRIn();
}

void loop(){
  if (irrecv.decode(&results)){
    long int decCode = results.value;
    Serial.println(results.value);
    switch (results.value){
      case 16753245: Serial.println("1"); break ;
      case 16736925: Serial.println("2"); break ;
      case 16769565: Serial.println("3"); break ;
      case 16720605: Serial.println("4"); break ;
      case 16712445: Serial.println("5"); break ;
      case 16761405: Serial.println("6"); break ;
      case 16769055: Serial.println("7"); break ;
      case 16754775: Serial.println("8"); break ;
      case 16748655: Serial.println("9"); break ;
      case 16750695: Serial.println("0"); break ;
      case 16738455: Serial.println("Asterisco"); break ;
      case 16756815: Serial.println("Cerquilha"); break ;
      case 16718055: Serial.println("Cima"); break ;
      case 16730805: Serial.println("Baixo"); break ;
      case 16734885: Serial.println("Direita"); break ;
      case 16716015: Serial.println("Esquerda"); break ;
```

```
    case 16726215: Serial.println("OK"); break ;
default:
    case UNKNOWN: Serial.println("Tecla Desconhecida"); break ;
}
```

# F LISTA DE COMPONENTES DO PROJETO

A Tabela 26 demonstra a quantidade (QtDs) de cada componente, demonstrando os valores em R\$ em duas lojas físicas e uma loja virtual realizada no dia 28/04/2022.

Tabela 26 – Lista de componentes do projeto: 23/05/2022.

Qt ds	Componentes	<b>Eletrônica System</b> Av Alberto Bins, 850 - Centro de POA, RS.	<b>Severo Roth</b> Av. Alberto Bins, 468 - Centro de POA, RS.	<b>FILIFELOP</b> < <a href="https://www.filipeflop.com/">https://www.filipeflop.com/</a> >
20	Resistor 1/4 W de 220 $\Omega$	R\$ 3,50	R\$ 5,50	R\$ 2,90
20	Resistor 1/4 W de 1 k $\Omega$	R\$ 3,50	R\$ 5,50	R\$ 2,90
20	Resistor 1/4 W de 10 k $\Omega$	R\$ 3,50	R\$ 5,50	R\$ 2,90
20	Resistor 1/4 W de 100 k $\Omega$	R\$ 3,50	R\$ 5,50	R\$ 2,90
02	Potenciômetro de 10 k $\Omega$	R\$ 4,00	R\$ 8,50	R\$ 5,80
02	Potenciômetro de 100 k $\Omega$	R\$ 4,00	R\$ 8,50	R\$ 4,80
02	Relé 5 V	R\$ 12,50	R\$ 17,00	R\$ 9,80
01	Motor 12 V	R\$ 9,50	R\$ 21,00	R\$ 9,80
02	Transistor NPN BC548	R\$ 1,75	R\$ 4,00	R\$ 0,40
02	Diodo Retificador 1N4148	R\$ 2,50	R\$ 4,50	R\$ 1,00
03	LED Vermelho LED Verde LED Amarelo	R\$ 3,75	R\$ 5,00	R\$ 0,90
01	Arduino UNO	R\$ 199,90	R\$ 299,90	R\$ 126,90
01	Sensor Ultrassônico HC-SR04	R\$ 59,90	R\$ 89,90	R\$ 38,90
01	LCD 20x4	79,90	R\$ 69,90	R\$ 59,90
01	Sensor + Receptor Infravermelho	35,90	R\$ 68,90	R\$ 12,90
05	Conectores Macho	R\$ 12,50	R\$ 29,90	R\$ 9,90
05	Conectores Fêmea	R\$ 12,50	R\$ 29,90	R\$ 9,90
05	Placas de Fenolite (5cmx5cm)	R\$ 12,50	59,90	R\$ 29,90
-	Total	R\$ 440,10	R\$ 738,80	R\$ 332,40

Fonte: O próprio autor (2022).