

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ALGACIR CANESTRARO JUNIOR

ALIMENTADOR PET VIA WIFI

CURITIBA

2022

ALGACIR CANESTRARO JUNIOR

ALIMENTADOR PET VIA WIFI

Pet feeder via wifi

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Sistema de Telecomunicações, do Departamento Acadêmico de Eletrônica/DAELN, da Universidade Tecnológica Federal do Paraná. Orientador: Prof. Nelson Garcia de Paula

CURITIBA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

ALGACIR CANESTRARO JUNIOR

ALIMENTADOR PET VIA WIFI

Trabalho de Conclusão do Cursor Superior de Tecnologia em Sistemas de Telecomunicações apresentado como requisito para obtenção do título de Tecnólogo em Sistemas de Telecomunicações da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 03 Maio de 2022

Tânia Lucia Monteiro
Doutorado
Universidade Tecnológica Federal do Paraná

Valfredo Pilla Júnior
Mestrado
Universidade Tecnológica Federal do Paraná

Nelson Garcia de Paula
Mestrado
Universidade Tecnológica Federal do Paraná

CURITIBA

2022

Aos meus pais Algacir e Irene (in memoriam), que o destino não me deixou ver realizar este sonho..

AGRADECIMENTOS

Agradeço primeiramente a Deus em que me apóia nos momentos difíceis, a minha esposa Madalena e minhas filhas Hellen e Kelly pelo apoio e ajuda quando eu mais precisava. Ao meu colega Bruno e sua esposa Marynea pela ajuda neste trabalho, aos professores e a instituição de ensino a qual tive o orgulho de estudar.

1

Toda a forma de vida é uma manifestação
de Deus e está sob os nossos cuidados.
(DE ASSIS; SÃO FRANCISCO, 1181 - 1226)

RESUMO

Este trabalho teve como objetivo criar um protótipo de alimentador pet controlado via WIFI. Os parâmetros a serem aplicados no protótipo são baixo custo e facilidade de utilização, valorizando o uso de materiais de fácil adaptação e substituição, visando um controle da alimentação do pet pelos donos ou tutores quando estão distantes dos animais.

Palavras-chave: Alimentação; Animais; Internet das Coisas; Animal de companhia.

ABSTRACT

This work had the objective of create a prototype for an automatic pet feeding machine with wifi interface control. There was it used low-cost versatile materials aiming at easy adaptation and mantainance. The WiFi interface controls the pet-feeding prototype machine while allowing communication with the feeding machine when the pet-guardians are distant from home or the animal.

Keywords: Food; Feeding Animals; Internet of Things; Pet animal.

LISTA DE ILUSTRAÇÕES

Figura 1 - Ração seca para gatos adultos.....	17
Figura 2 - Ração para gatos base salmão e frango.....	17
Figura 3 - Ração para gatos castrados.....	18
Figura 4 - Diferença das rações.....	18
Figura 5 - Fluxograma.....	19
Figura 6 - Corrente alternada.....	22
Figura 7 - Desenho com medidas do disco dosador.....	23
Figura 8 - Desenho esquemático alimentador.....	25
Fotografia 1 - Arduino.....	20
Fotografia 2- Shield Ethernet.....	21
Fotografia 3- Motor de Micro-ondas.....	21
Fotografia 4- Sensor ultrassônico fixado na tampa do dispenser.....	22
Fotografia 5- Peso da ração.....	23
Fotografia 6- Alimentador PET.....	24
Fotografia 7- Imagem do gauge no Celular.....	26
Fotografia 8- Bório Testando alimentador.....	26

LISTA DE ABREVIATURAS E SIGLAS

AJAX	Asynchronous JavaScript and XML
BBC	British Broadcasting Corporation
DNS	Domain Name System
IDE	Integrated Development Environment
IoT	Internet of Things
IP	Internet Protocol
PEGN	Pequenas Empresas Grandes Negócios
PVC	Policloreto de Vinila
RPM	Rotações por Minuto
XML	Extensible Markup Language

SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	Motivação.....	14
1.2	Justificativa.....	14
1.3	Objetivos.....	14
1.3.1	Objetivo geral.....	14
1.3.2	Objetivos específicos.....	14
2	FUNDAMENTAÇÃO TEORICA.....	15
2.1	Trabalhos similares.....	15
2.2	Produto similares no mercado.....	16
2.3	Alimentação.....	16
2.4	Reciclagem.....	18
3	METODOLOGIA.....	19
3.1	Fluxograma.....	19
3.2	Arduino.....	19
3.3	AJAX.....	20
3.4	Shield ethernet.....	20
3.5	Motor.....	21
3.6	Sensor de ultrassom.....	22
3.7	Disco dosador.....	23
4	RESULTADOS.....	24
4.1	Protótipo.....	24
4.2	Testes.....	26
5	CONSIDERAÇÕES FINAIS.....	27
	REFERÊNCIAS.....	29
	APÊNDICE A - Código fonte.....	32

1 INTRODUÇÃO

A Internet já possibilita a comunicação entre dispositivos que facilitam a vida do homem moderno, uma realidade que há alguns anos era roteiro de ficção científica. A internet das coisas (*Internet of Things* - IoT) começa a se tornar uma realidade, muitas situações já se tornaram corriqueiras e o leque de possibilidades é enorme: câmera de circuito fechado controlada por um celular; na saúde, monitorando pacientes e suas reações (LÓPEZ, 2020), na agricultura, com sensores de umidade e temperatura ou acionando sistema de irrigação (SANTOS, 2020).

Neste trabalho foi construído um alimentador que pode ser controlado pela internet, liberando alimento de forma automatizada e devolvendo ao dono do PET a informação em tempo real da quantidade de alimento presente no dispositivo. A plataforma escolhida para o desenvolvimento foi o Arduino Uno R3, por ser um controlador de baixo custo muito usado em IoT (BANZI 2012).

Este Alimentador PET via WEB foi projetado para uso em IoT pensando no cuidado com animais domésticos, pois os proprietários e cuidadores nem sempre podem estar presentes com seus Pets, isso devido a fatores como trabalho, estudo, viagens e outros.

O projeto em questão foi pensado para ser concebido com baixo custo, sendo ainda portátil, seguro e de fácil operação. Na elaboração do protótipo foi utilizado o microcontrolador Arduino Uno R3 com código aberto e materiais como PVC e peças reaproveitadas de sucata eletrônica.

O protótipo foi desenvolvido em duas partes. Sendo a primeira parte focada na montagem da estrutura do dispositivo de alimentação com tubos e conexões de PVC, facilitando o desenvolvimento do trabalho, no sistema eletrônico e mecânico, no sistema de apoio e na pintura

A segunda parte foi o desenvolvimento de um mini servidor WEB com auxílio do Shields ethernet e programação do microcontrolador

1.1 Motivação

Ter um animal de companhia em casa é uma grande responsabilidade, muitas vezes tratado como um amigo ou até como membro da família, gerando preocupação com o bem-estar e saúde do Pet, uma boa alimentação se faz necessária, conforme a raça, idade, porte, lactação. (SAAD, 2009).

Os tutores afastados de suas casas devido a trabalho, estudo, viagens, e outras atividades, tendem a ficar apreensivos por não poderem acompanhar presencialmente seus pets. A tecnologia moderna proporciona uma facilidade para minimizar os efeitos deste afastamento. Deixar de aproveitar estes recursos para dar uma atenção melhor ao seu Pet seria um desperdício

1.2 Justificativa

Esta proposta de projeto trata da necessidade de um alimentador controlado via internet para animais de pequeno porte, tais como cães e gatos. A solução busca suprir as necessidades de pessoas que precisam se afastar do seu animal de estimação por pequenos períodos.

1.3 Objetivo

1.3.1 Geral

Construir um alimentador PET controlado via internet, com matérias de fácil aquisição e com baixo custo financeiro,

1.3.2 Objetivos Específicos

- Desenvolvimento de uma plataforma automatizada para controle da alimentação de animais domésticos;
- Analisar protocolos de comunicação adequados ao dispositivo;
- Desenvolver sistema eletromecânico do dispositivo

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Trabalhos similares

O Alimentador construído por ALMEIDA (2016) visa somente alimentação de cães, tem uma comunicação wi-fi que informa a quantidade de alimento ingerida pelo animal pesando a tigela de alimento. O trabalho não informa se há controle de alimento armazenado no aparelho. Comenta de programação, mas não mostra nenhum código-fonte.

O projeto desenvolvido por SANTOS (2015) é semelhante ao BUOGO (2017), respeitando diferenças eletrônicas e mecânicas. Os dois alimentadores são programados com horários até três a quatro vezes ao dia, com o fornecimento de água e um sistema de reposição com controle de nível.

Em OLIVEIRA (2017) é feita uma programação a distância de horários, utilizando um Raspberry Pi. Não é apresentada a parte mecânica do alimentador.

O trabalho de PETER, KLOCNER, PEREIRA (2015), tem um projeto totalmente teórico, focando somente na parte comercial de um alimentador sem um projeto pratico.

O trabalho de DESSBESELL, VALDIERO, KLEVERSON, RASIA (2014) apresenta um alimentador programado com PIC, infelizmente não forneceram nenhuma informação sobre a programação do microcontrolador. Não há código-fonte.

Em FARHAT, KLINGENFUS (2019), é apresentado o alimentador usando Raspberry Pi, tem adicionado em sua programação um controle de horário. O diferencial deste alimentador é o de ter uma câmera que mostra a tigela de alimentação, informando visualmente se o recipiente esta cheio de ração.

No trabalho de ULRICH, NOVAK (2012) é comentado sobre alimentação, tipos de alimentadores mecânicos, potes, materiais utilizados. O projeto apresenta o design e a forma de ser construído, superficialmente falando sobre parte eletrônica e mecânica do seu projeto.

2.2 Produtos similares no mercado

Foi publicada uma reportagem sobre um alimentador automático para pets muito semelhante à idéia proposta fabricado pela Zenpet, (2018) enviando informação a um aplicativo de celular (MOREIRA, 2019).

A empresa Petnet fornece uma solução de IoT integrada com smartphone, o produto também interage com outros aparelhos inteligentes na residência. Nas redes sociais alguns clientes relataram problemas de comunicação, onde pets foram negligenciados por falhas no serviço de comunicação entre a empresa e o dispensador de comida da Petnet. Animais confiados ao alimentador da Petnet ficaram abandonados e sem alimentação devido a esta falha de serviço (GONÇALVES, 2020).

O artigo de WILLIANS (2020) comenta da tendência do desaparecimento prematuro deste tipo de serviço em IoT. Deixando usuários abandonados quando o projeto se baseia fortemente em comunicação com a central.

Um bom sistema alimentador deveria ser autônomo, mas algumas soluções através do paradigma IoT tem produzido sistemas que podem supostamente ser desligados por omissão, negligência ou malícia do fabricante. Além de estarem vulneráveis a hackeamento à distância por entidades maliciosas, HERNANDES (2019). Um sistema redundante poderia ter incorporado aos aparelhos evitando que o sistema parasse de funcionar ao perder comunicação com o servidor. Em inglês existe o verbo “to brick” indicando que um produto foi transformado em peso de papel ou tijolo pelo fabricante à distância. O governo do Reino Unido planeja novas leis para aparelhos inteligentes vendidos em seu país. (SHEAD, 2020).

2.3 Alimentação

Uma nutrição ideal se faz necessária para o desenvolvimento saudável de animais em diversas fases da vida: varia desde o nascimento até a parte final de sua vida; há mudanças na necessidade de sua nutrição, alterando as porções e quantidade de refeições diárias, dependendo da idade e porte; o animal não tem a capacidade de saber a hora e quantidade de sua alimentação.

Atualmente, há também uma grande variedade de alimentação para pets, um médico veterinário é o responsável pela indicação do melhor alimento, a quantidade

e porções diárias para a saúde do animal, como apresenta (CARCIOFI: JEREMIAS, 2010, pg. 39).

O modo como cães e gatos estão inseridos na sociedade, a preocupação dos proprietários com alimentação e o desenvolvimento industrial motivado pelo aquecimento econômico do setor, sem dúvida, contribuíram em grande parte para o direcionamento do atual perfil de pesquisas, como foco principal em estudos sobre formulação de alimentos capazes de maximizar a expectativa e a qualidade de vida, pela utilização de ingredientes e nutrientes que desenvolvam a capacidade de resistir a doenças e melhorem a saúde.

Conforme as Figuras 1, 2 e 3, pode-se observar que diferentes marcas de ração tem sua própria composição nutricional, podendo variar a dosagem conforme o peso e o tamanho. A Figura 4 mostra resumidamente a comparação entre três marcas nomeamos de “A, B e C”, para gatos adultos,

Figura 1 – Ração seca gatos adultos.

Guias de nutrição		
Peso do gato adulto	Quantidade* por dia	Quantidade* por dia (Seco+sachê)
3 - 4 kg	45 - 55 g	25 - 35 g + 1 sachê
4 - 5 kg	55 - 65 g	35 - 45 g + 1 sachê
5 - 6 kg	65 - 75 g	45 - 55 g + 1 sachê

*Quantidades individuais podem mudar. 1 xícara = 80 g
aproximadamente 1 sachê Gato Adulto WHISKAS® = 85 g

Fonte: Whiskas (2020)

Figura 2 – Ração para gatos base salmão e frango.

Tamanho do seu gatinho	Quantidade diária recomendada
2kg a 3,5kg	(30g a 50g)
3,5kg a 6,5kg	(50g a 90g)
6,5kg a 11kg	(90g a 150g)

Fonte: Petlove (2020)

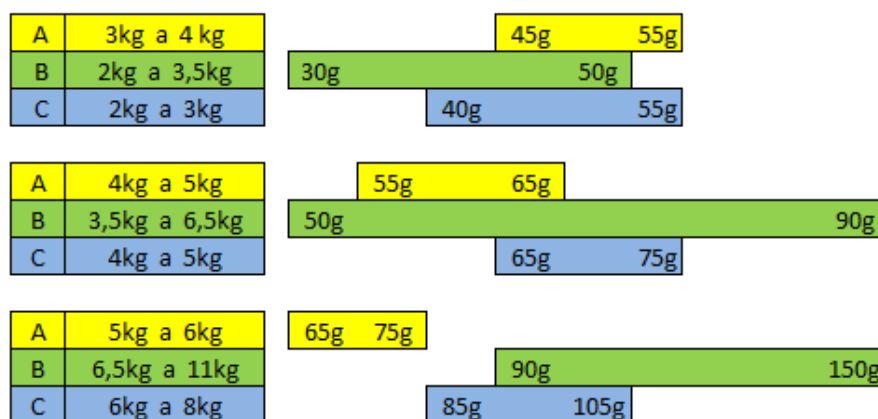
Figura 3 – Ração para gatos castrados.

Quantidade Recomendada	Peso do Gato adulto	
	2kg - 3kg	40g - 55g
	4kg - 5kg	65g - 75g
	6kg - 8kg	85g - 105g

Fonte: Purina (2020)

Pode-se observar a diferença pode ser bem discrepante, dependendo do peso, as quantidades necessárias são diferentes dependendo da marca.

Figura 4 – Diferença das rações



Fonte: Autoria própria (2020)

2.4 Reciclagem

Segundo Zanin, Mancini (2025), “Dentro da concepção de que tudo começa e termina no meio ambiente, um ciclo de vida para o desenvolvimento de produto deve priorizar a durabilidade, a facilidade de reparação e manutenção além do uso de materiais que possivelmente possam ser reaproveitados (reutilizados) e reciclados”. (ZANIN, MANCINI, 2015).

Neste trabalho, o protótipo do alimentador pet teve partes reaproveitadas de outras aplicações domésticas. Foi aproveitado o motor de uma sucata de

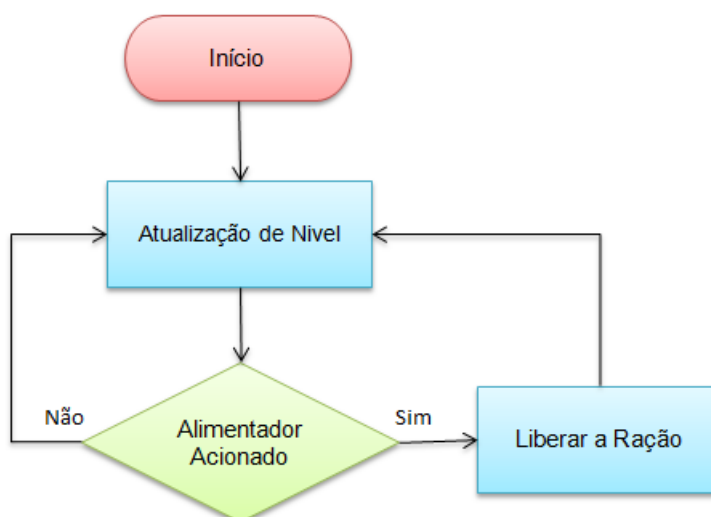
microondas, e o disco dosador foi feito do aproveitamento de uma luminária *spot* doméstico. Foi também utilizado um tubo de PVC comum para servir de repositório de ração e posterior distribuição da mesma ração. Tubos de PVC obedecem às diretivas de durabilidade, reparação, e manutenção, caso seja necessário substituição.

3 METODOLOGIA

3.1 Fluxograma

A figura 5 mostra o fluxograma do alimentador. Como se pode notar uma vez acionado, entra em um loop infinito, somente sendo interrompido por um fator externo.

Figura 5- Fluxograma



Fonte: Autoria própria (2020)

3.2 Arduino

Devido ao seu baixo custo e fácil operação, foi escolhido para o projeto o Arduino Uno R3, Figura 6, placa de prototipagem, com micro controlador Atmel AVR. Tendo a sua própria IDE, possibilita uso de diversas extensões conhecidas como Shields e módulos com uma ampla biblioteca de software livre. (ARDUINO, 2019).

Fotografia 1- Arduino

Fonte: Autoria própria (2020)

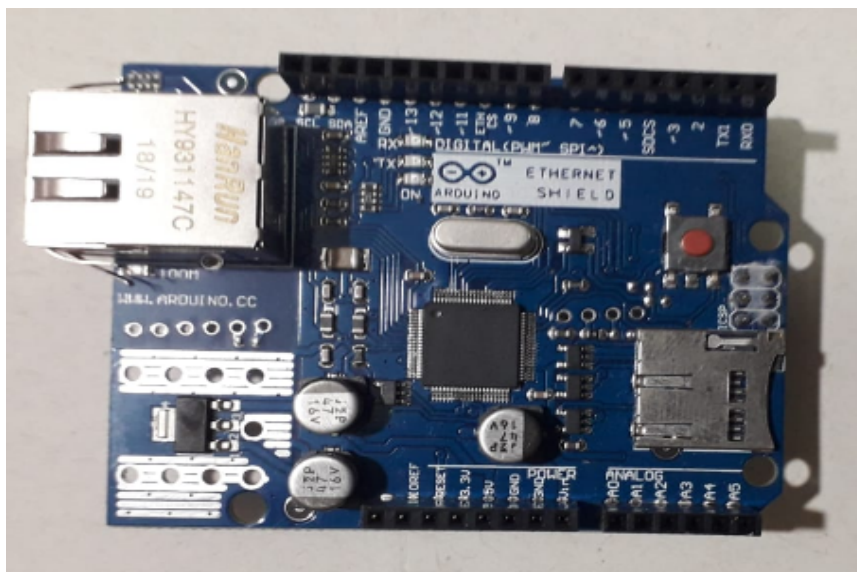
3.3 AJAX

Linguagem de programação conhecida por suas funcionalidades, com a capacidade de coordenar de um site e permitindo uma comunicação prática com o usuário. O HTML exibe os dados, o XML guarda e transmite, os dois trabalham de forma assíncrona no AJAX. Dessa forma uma aplicação que use AJAX, não necessita recarregar a página inteira, enviando e recebendo os dados do servidor.

O Ajax dial faz o interfaceamento entre o arduino e a página do alimentador, para do alimentador, para acesso local tem que entrar com o endereço ip do aparelho. para acesso externo criar uma página pelo com dns (domain name system) válido.

3.4 Shield Ethernet

Placa conectada ao Arduino UNO, Figura 7, que permite uma conexão remota, obtendo acesso a web, com uma biblioteca própria (ARDUINO, 2019). Serve tanto para transmissão quanto para recepção.

Fotografia 2 – Shield Ethernet

Fonte: Autoria própria (2020)

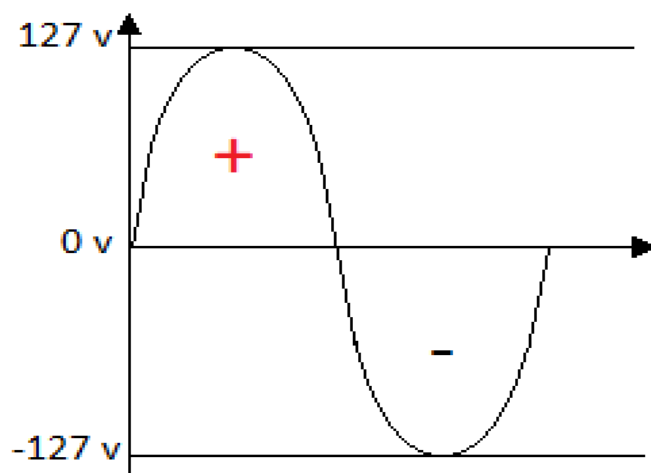
3.5 Motor

Para montagem do protótipo foi escolhido um motor do eixo do prato de uma sucata de forno micro-ondas, Figura 8, que tem uma baixa rotação, 6 RPM. Adaptado ao protótipo alimentador está ligado ao disco dosador, que ao receber o sinal de comando gira o disco por 10 segundos, e retorna à posição de espera do disco totalizando uma volta do disco.

Fotografia 3– Motor de Micro-ondas

Fonte: Autoria própria (2020)

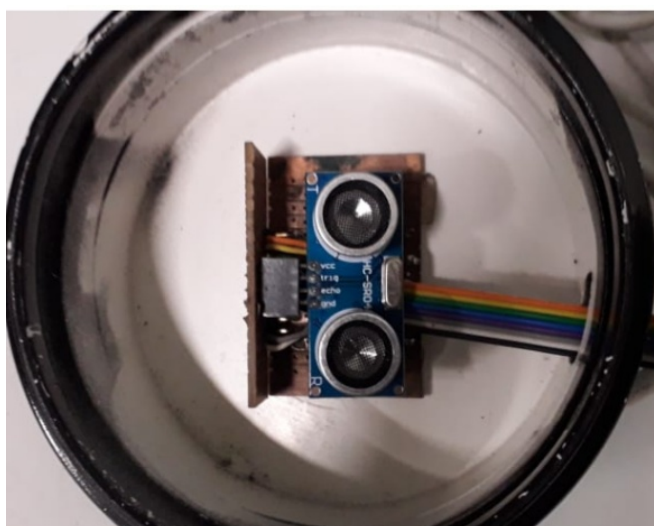
O motor trabalha com corrente alternada, Figura 9, conforme o momento que ele for ligado terá uma rotação horária ou anti-horária.

Figura 6 – Corrente alternada

Fonte: Autoria própria (2020)

3.6 Sensor de ultrassom

Utilizado para medir a quantidade de alimento dentro do cilindro, envia a informação da necessidade de reposição ou se há um nível satisfatório de ração para alimentar o pet

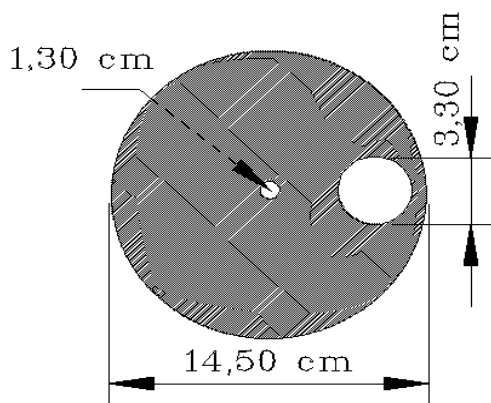
Fotografia 4 – Sensor ultrassônico fixado na tampa do dispenser

Fonte: Autoria própria (2020)

3.7 Disco Dosador

Responsável pela passagem do alimento do depósito ao prato, tem um diâmetro de e 14,5 centímetros, com uma abertura de 3,30 centímetros de forma excêntrica, onde o giro do disco faz a abertura se posicionar exatamente sobre o funil, liberando o alimento.

Figura 7 – Desenho com medidas do disco dosador



Fonte: Autoria própria (2020)

A cada rotação do disco, é liberada uma quantidade de ração 45 gramas aproximadamente, Figura 12. Alterando a abertura do disco, aumenta-se ou diminuiu a quantidade liberada.

Fotografia 5 – Peso da ração



Fonte: Autoria Própria (2020)

4 RESULTADOS

4.1 Protótipo

O alimentador pet tem a cor preta com desenho de patinhas na base. Há uma saída para o alimento que devem ser de baixa granularidade. No protótipo usaram-se as rações da marca Special Cat. Fabricada pela empresa SpecialDog, “Special Cat.” para gatos adultos castrados “(PETLOVE, 2018), provavelmente rações de outras marcas tanto para gatos como cães devem funcionar também, mas o teste do protótipo usou somente a marca já citada.

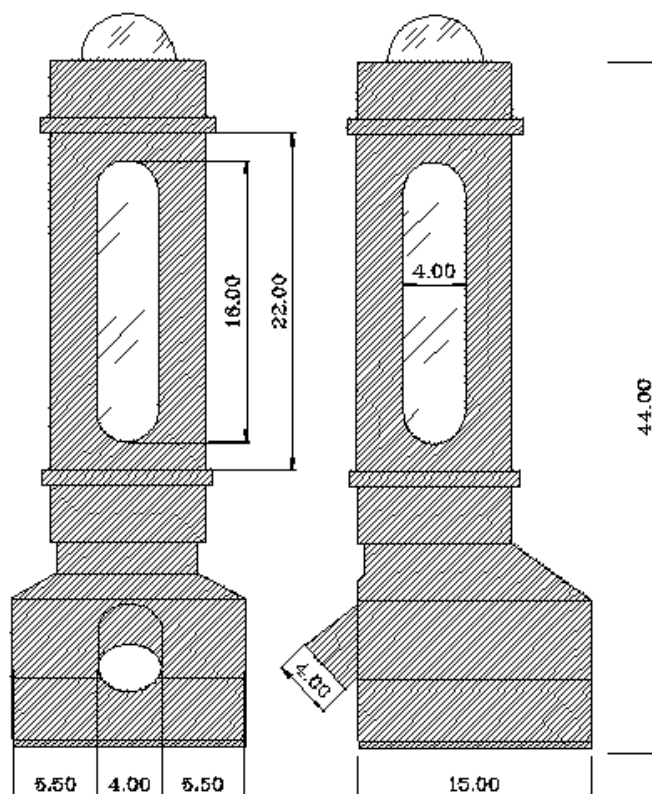
No topo do protótipo tem um domo luminoso que indica o nível de alimento no reservatório através de mudança de cor. Quando o domo está na cor azul o recipiente está cheio, cor verde nível satisfatório, cor amarela nível regular e vermelho quando o nível estiver vazio. O depósito cilíndrico de ração tem quatro aberturas protegidas por um plástico transparente, para visualização do conteúdo interno, com dimensões de 16 cm de altura por 4 cm largura cada, seu tamanho é de 22 cm de altura com 10 cm de diâmetro. Aparte inferior onde está localizado o motor e a parte eletrônica tem 14 cm de altura com 15 cm de diâmetro na parte cilíndrica, a base é um suporte quadrado 15 cm de lado com 0,5 cm de altura, a tensão de trabalho e de 127 volts, Figura 13. Desenho esquemático pode ser visualizado na Figura 14.

Fotografia 6– Alimentador PET.



Fonte: Autoria própria (2020)

Figura 8 – Desenho esquemático alimentador PET



Fonte: Autoria própria (2020)

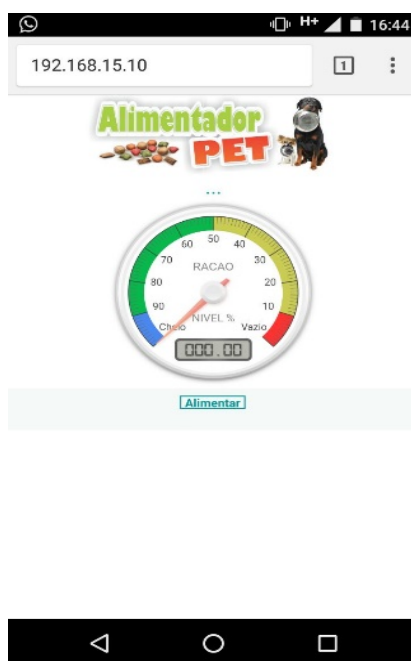
Este protótipo alimentador não possui botão liga/desliga, sendo ligado e desligado diretamente na tomada. O aparelho possui um botão de teste do dosador que está localizado abaixo do depósito cilíndrico, quando pressionado libera uma porção de ração.

Para configurar o aparelho protótipo na primeira vez, deve-se entrar na programação do Arduino, através de computador, e especificar um IP local válido.

```
(IPAddressip (192, 168, 25, 15); // IP address, may need to
change depending on network)
```

Uma vez configurado, o alimentador pet tem que acessar a página controlada por um aplicativo de celular, onde é possível ver o nível da ração disponível no depósito, Figura 15.

Fotografia 7– Imagem do gauge no celular



Fonte: Autoria própria (2020)

No Apêndice está o código fonte do dispositivo escrito na linguagem Java.

4.2 Testes

O Protótipo foi testado pelo meu gato Boreo, Figura 16. No teste foi ofertado alimento pelo aparelho por um período de quatro dias, no primeiro momento notou-se uma pequena aversão a forma que recebia o alimento, após este breve período inicial já estava habituado ao alimentador.

Fotografia 8– O gato Bório testando alimentador



Fonte: A autoria própria (2020)

Depois do período de testes, não tivemos nenhum problema com o funcionamento do aparelho, o animal já esperava ao lado do alimentador na hora de receber a ração. O protótipo teve que ser reabastecido a cada dois dias.

5 CONSIDERAÇÕES FINAIS

Este TCC se concentrou na construção do protótipo alimentador PET, e provou o funcionamento conceitual com somente um sujeito de teste, uma marca de ração. Mais testes são necessários para a coleta e antecipação de possíveis problemas com esta forma de alimentação do “pet”.

Boreo, o gato usado no teste, é relativamente calmo, mas há diferentes temperamentos entre os próprios gatos e cães. Há gatos mais ariscos, outros mais calmos, outros preguiçosos, ou mais exigentes na forma de ser alimentado. Um pet poderia estragar o alimentador. Estudar a adesão a forma de recebimento da ração por diferentes animais é uma característica importante na evolução do projeto.

Sugere-se deixar os animais interagirem de forma independente com o protótipo, monitorando-os a distância para se obter uma análise mais detalhada da interação entre o protótipo e outros animais domésticos.

Para funcionamento em longo prazo do protótipo, que tipo de manutenção seria necessária e periodicidade da limpeza também fazem parte de etapas futuras.

Um possível problema antecipado nesta versão única do protótipo poderia ser o fato de um animal mover o pote do lugar derramando a ração no chão e contaminando o alimento. Para resolver o problema a base do alimentador seria maior se prolongando até onde o pote seria encaixado.

Para o futuro, atualizações são necessários, mudar o designer deixar mais leve, testar novos sistemas mecânicos, adaptar um sistema de bebedouro junto ao alimentador, com um circuito independente acionado por sensor de movimento.

O protótipo foi idealizado para ser um produto de fácil montagem e simples utilização. Os testes mostraram uma ótima eficiência seguindo a idéia principal. Com modificações na montagem é possível adaptar para os outros tipos de animais porte menores como hamsters, coelhos, como um sistema de alimentação para animais em granjas, também com bombas hidráulicas e sensores de umidade para sistemas de irrigação.

Este protótipo poderia ser usado como base de comunicação de uma empresa que forneceria a ração para reposição e poderia monitorar as variações de consumo do animal, fornecendo mais dados ao dono ou a um profissional de saúde veterinária.

REFERÊNCIAS

- ALMEIDA, M. N. de B. **Comedouro automático para cães**. 2016. Projeto de monografia - Disponível em: <http://lyceumononline.usf.edu.br/salavirtual/documentos/2954.pdf>. Acesso em: 12 set. 2021.
- ARDUINO, **Ethernet library**. 2019. Disponível em: <https://www.arduino.cc/en/Reference/SD>. Acesso em: 25 mar. 2019.
- ARDUINO, **SD library**. 2019. Disponível em: <https://www.arduino.cc/en/Reference/Ethernet>. Acesso em: 25 mar. 2019
- BANZI, Massimo. **Primeiros passos com o Arduino**. 1. ed. São Paulo: Editora Novatec. 2012.
- BUOGO, D. R. **Protótipo de Alimentador Automático Para animais Domesticos-Gingapets**. 2017. Cursor de mecatrônica - Disponível em: <http://joinville.ifsc.edu.br/~bibliotecajoi/arquivos/tcc/mecind/180273.pdf>. Acesso em 12set. 2021.
- CARCIOFI, A. C. **Progresso científico sobre nutrição de animais de companhia na primeira década do século XXI**, 2010. Trabalho de Zootecnia - Disponível em: <https://repositorio.unesp.br/bitstream/handle/11449/1837/S1516-35982010001300005.pdf?sequence=1&isAllowed=y>. Acesso em 03 jan. 2020.
- DESSBESELL, E. H.; VALDIERO, A. C.; KLEVERSON, O. L.; RASIA, L. A. **Desenvolvimento e construção de Máquina Para Alimentação Automática de Pequenos Animais**. 2014. Disponível em [http://www.simmer.unijui.edu.br/Periodicos/R0568-1\[1\].pdf](http://www.simmer.unijui.edu.br/Periodicos/R0568-1[1].pdf). Acesso em 09 set. 2021.
- FARHAT, N. de A.; KLINGENFUS, V. de S.. **Alimentador Para Pequenos Animais Controlado Por Aplicativo Móvel**. 2019. Trabalho de conclusão de curso - Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/12758/1/CT_DAELN_2019_1_08.pdf. Acesso em 08 set. 2021.
- GONÇALVES, A. L. D. **Gatos Ficam Sem Comida Após Falha em Alimentadores Inteligentes**, 2020, Disponível em: <https://m.tecmundo.com.br/amp/software/150562-gatos-comida-falha-alimentadores-inteligentes.htm>. Acesso em: 01 mai. 2020
- LÓPEZ, L. J. R.; LÓPEZ, A. F. M.; SANABRIA, Y. P. C., **Aplicación de Labiotemetría para Três Signos Vitales**, 2015. Disponível em: <https://publicacionesfac.com/index.php/cienciaypoderaereo/article/view/428>. Acesso em: 03 jan. 2020.
- MOREIRA, M. I., Revista Startups, **Como Empresas Brasileiras Estão Inovando no Setor de Hardware**, 2019. Disponível em:

<https://revistapegn.globo.com/Startups/noticia/2019/03/como-empresas-brasileiras-estao-inovando-no-setor-de-hardware.html>. Acesso em 04 fev. 2020.

OLIVEIRA, R. C. B. de. **Alimentador de Animais Domésticos Automático com Comunicação a Distância**. 2017. Trabalho de monografia - Disponível em: <http://monografias.poli.ufrj.br/monografias/monopoli10022726.pdf>. Acesso em 10 set. 2021.

PETER, A. V.; KLOCNER, D.; PEREIRA, L. A. C. **A Alimentação de Animais Domésticos Quando Sozinhos**. 2015. Trabalho curso de administração - Disponível em: <http://www.projetos.unijui.edu.br/moeducitec/moeducitec/principal/151.pdf>. Acesso em 10 set. 2021.

PETLOVE. **Ração seca Nutrire Monello Cat Salmão, Atum & Frango para Gatos Adultos**. 2020. Disponível em: <https://www.petlove.com.br/racao-seca-nutrire-monello-cat-salmaa-atum-frango-para-gatos-adultos/p?sku=2422909>. Acesso em 14 fev. 2021.

PETLOVE. **Ração Special Cat Premium para gatos Adultos**, 2018, Disponível em: <https://www.petlove.com.br/racao-special-cat-premium-para-gatos-adultos-castrados/p>. Acesso em; 04 jan. 2020.

PETNET, THE PETNET SMART FEEDER, **Automatic pet feeder for cats and dogs**. 2020, Disponível em: <https://www.petnet.io/pages/smartfeeder>. Acesso em 31 Jan. 2020.

PURINA. **Ração para Gatos Castrados – Megamix de Carnes**. 2020. Disponível em: <https://www.purina-latam.com/pt-br/br/friskies/produtos/megamix-adultos-e-castrados>. Acesso em 14 fev. 2021.

SAAD, F. M. de O. B. **ASPECTOS NUTRICIONAIS DE CÃES E GATOS EM VÁRIAS FASES FISIOLÓGICAS**. 2009. Disponível em: <https://www.researchgate.net/publication/270508665>. Acesso em: 25 Jan. 2020.

SANTOS, B. F. de B. S. **Alimentador automático Para animais Utilizando Arduino**. 2016. Trabalho de conclusão de curso em engenharia - Disponível em: <https://repositorio.unesp.br/bitstream/handle/11449/139072/000864824.pdf?sequence=1&isAllowed=y>. Acesso em 12 set. 2021.

SANTOS, F. de A. M. dos, **PROJETO DE IRRIGAÇÃO INTELIGENTE**, 2010, Disponível em: <https://docs.google.com/viewerng/viewer?url=http://www2.ifrn.edu.br/ojs/index.php/HOLOS/article/viewFile/553/410>. Acesso em 04 jan. 2020.

SHEAD, S., **Government plans new laws for smart gadgets sold in UK**, 2020. Disponível em: <https://www.bbc.com/news/technology-51271717>. Acesso em 05 fev. 2020.

ULRICH, A. P. de A.; NOVAK, J. L.. **Alimentador Voltado Para Animais Domésticos**. 2012. Trabalho de conclusão de curso - Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/1302/1/CT_CODES_2012_1_13.pdf. Acesso em 08 Set. 2021.

WHISKAS. **Ração Seca para Gatos adulto Sabor Carne 10,1kg**. 2020. Disponível em: <https://www.whiskas.com.br/nossos-produtos/secos/whiskasandreg-sabor-carne-10kg>. Acesso em 14 fev. 2021.

WILLIAMS, O. **The internet of Things almost starved My Cat**, 2020, Disponível em: <https://onezero.medium.com/the-internet-of-things-almost-starved-my-cat-517ca7554b61>. Acesso em: 28 fev. 2020.

ZANIN, M.; MANCINI, S. D. **Resíduos Plásticos e Reciclagem**. 2015. Disponível em: https://www.google.com.br/books/edition/Res%C3%ADduos_pl%C3%A1sticos_e_reciclagem/rSIXDQAAQBAJ?hl=pt-BR&gbpv=1&dq=gratuitos+reciclagem&printsec=frontcover. Acesso em 24 set. 2021.

ZEMPET. **Peace of Mind for Petlovers**, 2018. Disponível em: <https://zen.pet/#!produtos>. Acesso em 24 Mai. 2020.

APÊNDICE A - Código fonte

```

#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>
#include <Ultrasonic.h>

//Define os pinos para o trigger e echo
#define pino_trigger 5
#define pino_echo 3

//Inicializa o sensor nos pinos definidos acima
Ultrasonic ultrassom(5,3);
// size of buffer used to capture HTTP requests
#define REQ_BUF_SZ 50

// MAC address from Ethernet shield sticker under board
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 25, 15); // IP address, may need to change depending on network
EthernetServer server(80); // create a server at port 80
File webFile; // the web page file on the SD card
char HTTP_req[REQ_BUF_SZ] = {0}; // buffered HTTP request stored as null terminated string
char req_index = 0; // index into HTTP_req buffer

const int carga1 = 7;
int flag1 = 0;
const int ledAZ = 14;
const int ledVD = 15;
const int ledAM = 16;
const int ledVM = 17;

long distancia;
EthernetClient client;

boolean currentLineIsBlank;
char c;
int botao = 2;
int estadoBotao = LOW;

void setup()
{
    pinMode(carga1, OUTPUT);
    pinMode(ledAZ, OUTPUT);
    pinMode(ledVD, OUTPUT);
    pinMode(ledAM, OUTPUT);
    pinMode(ledVM, OUTPUT);
    pinMode(botao, INPUT);

    // disable Ethernet chip
    pinMode(10, OUTPUT);
    digitalWrite(10, HIGH);

    Serial.begin(9600); // for debugging
    //Serial.print(distancia); //imprime o valor da variável distancia
    // Serial.println("cm");
    // initialize SD card
    //Serial.println("Initializing SD card...");
    if (!SD.begin(4)) {
        Serial.println("ERROR - SD card initialization failed!");
        return; // init failed
    }
    Serial.println("SUCCESS - SD card initialized.");
    // check for index.htm file
    if (!SD.exists("index.htm")) {
        Serial.println("ERROR - Can't find index.htm file!");
        return; // can't find index file
    }
    //Serial.println("SUCCESS - Found index.htm file.");

    Ethernet.begin(mac, ip); // initialize Ethernet device
    server.begin(); // start to listen for cls
}

void loop()
{

```



```

estadoBotao = digitalRead(botao);
if (estadoBotao == HIGH) {
  SetCargaBotao();
}
client = server.available(); // try to get client

if (StrContains(HTTP_req, "ajax_carga1")) {
  SetCarga1();
}
//Le as informacoes do sensor, em cm e pol

distancia = ultrassom.Ranging(CM);

led_nivel();

if (client) { // got client?
  currentLineIsBlank = true;
  while (client.connected()) {
    if (client.available()) { // client data available to read
      c = client.read(); // read 1 byte (character) from client
      // buffer first part of HTTP request in HTTP_req array (string)
      // leave last element in array as 0 to null terminate string (REQ_BUF_SZ - 1)
      if (req_index < (REQ_BUF_SZ - 1)) {
        HTTP_req[req_index] = c; // save HTTP request character
        req_index++;
      }
      // last line of client request is blank and ends with \n
      // respond to client only after last line received
      if (c == '\n' && currentLineIsBlank) {
        // send a standard http response header
        client.println("HTTP/1.1 200 OK");
        // remainder of header follows below, depending on if
        // web page or XML page is requested
        // Ajax request - send XML file

        if (StrContains(HTTP_req, "ajax_carga1")) {
          SetCarga1();
          Serial.println(HTTP_req);
        }

        if (StrContains(HTTP_req, "ajax_inputs")) {
          // send rest of HTTP header
          client.println("Content-Type: text/xml");
          client.println("Connection: keep-alive");
          client.println();

          // send XML file containing input states
          XML_response(client);
        }
        else { // web page request
          // send rest of HTTP header
          client.println("Content-Type: text/html");
          client.println("Connection: keep-alive");
          client.println();

          // send web page
          webFile = SD.open("index.htm"); // open web page file
          if (webFile) {
            while (webFile.available()) {
              client.write(webFile.read()); // send web page to client
            }
            webFile.close();
          }
        }
        // display received HTTP request on serial port
        //Serial.println(HTTP_req);
        // reset buffer index and all buffer elements to 0
        req_index = 0;
        StrClear(HTTP_req, REQ_BUF_SZ);
      }
    }
  }
}

```

```

        break;
    }
    // every line of text received from the client ends with \r\n
    if (c == '\n') {

        // last character on line of received text
        // starting new line with next character read
        currentLineIsBlank = true;
    }
    else if (c != '\r') {
        // a text character was received from client
        currentLineIsBlank = false;
    }
    } // end if (client.available())
} // end while (client.connected())
//delay(1); // give the web browser time to receive the data
client.stop(); // close the connection
} // end if (client)

}

void led_nivel() {
    if (distancia < 6.5) {
        digitalWrite (ledAZ, HIGH);
        digitalWrite (ledVD, LOW);
        digitalWrite (ledAM, LOW);
        digitalWrite (ledVM, LOW);
    }

    if (distancia >= 6.5 && distancia < 16) {
        digitalWrite (ledAZ, LOW);
        digitalWrite (ledVD, HIGH);
        digitalWrite (ledAM, LOW);
        digitalWrite (ledVM, LOW);
    }

    if (distancia >= 16 && distancia < 26.2) {
        digitalWrite (ledAZ, LOW);
        digitalWrite (ledVD, LOW);
        digitalWrite (ledAM, HIGH);
        digitalWrite (ledVM, LOW);
    }

    if (distancia >= 26.2) {
        digitalWrite (ledAZ, LOW);
        digitalWrite (ledVD, LOW);
        digitalWrite (ledAM, LOW);
        digitalWrite (ledVM, HIGH);
    }
    Serial.println(distancia);
    return;
}

// send the XML file containing analog value
void XML_response(EthernetClient cl)
{
    float analog_val;
    // long microsec = ultrasonic.timing();

    cl.print("<?xml version = '1.0' ?>");
    cl.print("<inputs>");
    // read analog pin A2

    cl.print("<analog>");
    cl.print(analog_val);
    cl.print("</analog>");
    cl.print("</inputs>");

}

void SetCarga1() {
    if (flag1 == 0) {
        digitalWrite(carga1, LOW);
        flag1 = 1;
        delay(10250);
        digitalWrite(carga1, HIGH);
    }
}

```

```
    flag1 = 0;
  }
}

void SetCargaBotao() {
  digitalWrite(carga1, LOW);
  delay(10250);
  digitalWrite(carga1, HIGH);
}

// sets every element of str to 0 (clears array)
void StrClear(char *str, char length)
{
  for (int i = 0; i < length; i++) {
    str[i] = 0;
  }
}

// searches for the string sfind in the string str
// returns 1 if string found
// returns 0 if string not found
char StrContains(char *str, char *sfind)
{
  char found = 0;
  char index = 0;
  char len;

  len = strlen(str);

  if (strlen(sfind) > len) {
    return 0;
  }
  while (index < len) {
    if (str[index] == sfind[found]) {
      found++;
      if (strlen(sfind) == found) {
        return 1;
      }
    }
    else {
      found = 0;
    }
    index++;
  }
  return 0;
}
```