

# Alimentador automático de animais: Uma simulação realizada no Cooja

Rafael Tenfen<sup>1</sup>

<sup>1</sup> Programa de Pós-Graduação em Computação Aplicada (PPGCAP)  
Centro de Ciências Tecnológicas (CCT)  
Universidade do Estado de Santa Catarina (UDESC)

Joinville - SC - Brasil

rafaeltenfen.rt@gmail.com

**Abstract.** *Technology is increasingly present in our lives, to turn on a lamp you don't need to go to the switch, just ask for the lamp to turn it on, in some models you even choose the color, in addition to the devices being more technological, they are also more automated aiming to make human life easier. The objective of this study is to simulate and feed any animal, whether domestic or rural, being able to specify the initial amount of feed in the pot in grams, the amount to be disposed of in each device and how many times a day. Therefore, the automatic feeder in a simulated environment provided in this work is developed in Contiki OS using the Cooja simulator with MQTT to exchange messages and a public web interface to monitor the progress of the amount of food on each device.*

**Resumo.** *A tecnologia está cada vez mais presente em nossas vidas, para ligar uma lâmpada você não precisa ir até o interruptor, basta solicitar a lâmpada para ela ligar, em alguns modelos até escolher a cor, além dos dispositivos estarem mais tecnológicos, eles estão mais automatizados, visando facilitar a vida humana. O objetivo desse estudo é simular e alimentar os animais seja ele doméstico ou rural, podendo especificar a quantidade inicial de ração no pote em gramas, a quantidade a ser disposta em cada dispositivo e quantas vezes por dia. Assim, o alimentador automático em um ambiente simulado disposto nesse trabalho, é desenvolvido no Contiki OS utilizando o simulador Cooja com o MQTT para troca de mensagens e uma interface pública na web para acompanhar o andamento da quantidade de comida em cada dispositivo.*

**Palavras-chaves**— alimentador automático, automação, MQTT, Cooja

## 1. Introdução

A ideia principal da Internet das coisas é interconectar múltiplos eletrônicos analógicos e digitais dispositivos, homogêneos e heterogêneos na natureza, mas com em sua faixa de transmissão sobreposta entre si, então que eles podem comunicar as informações de forma eficiente. *IoT* (Internet of Things) é um campo em crescimento e continuaria a crescer exponencialmente nos anos futuros [Chaudhary et al. 2019]. A humanidade busca sempre estar em evolução para facilitar o seu dia a dia, uma ótima maneira de auxiliar nas atividades diárias é automatizar as tarefas básicas com a utilização de dispositivos inteligentes que auxiliam na interconexão de dispositivos.

Na busca de facilitadores para a vida, vários avanços científicos e também automação de muitos processos resolveram problemas industriais e sociais complexos para criarem produtos com soluções inovadores. A IoT é uma tendência atual de desenvolvimento tecnológico, com ela é possível criar novos aplicativos que gerem novas oportunidades, como na economia e na sociedade [Quiñonez et al. 2021]. Dessa forma, inúmeras aplicações surgem diariamente com a possibilidade de agregar espaço no cotidiano das pessoas para automatizar em campos como casa inteligente, saúde, agricultura entre outros.

Nesse sentido, esse trabalho busca apresentar uma proposta de simulação de dispositivos para facilitar a vida de pessoas que possuem animais para realizar a alimentação desses animais de modo automático em que necessita ser configurado apenas uma vez e pode assim dispensar alimento para o animal inúmeras vezes ao dia, de acordo com a porção indicada. A possibilidade de implementação será apresentada através do Contiki no simulador Cooja e utilizando ferramentas como MQTT (*Message Queuing Telemetry Transport*) para troca de mensagens, uma página pública de domínio aberto disponibilizado pelo GCP (*Google Cloud Platform*).

Os dispositivos de alimentação automática, devem ser destinados a qualquer animal que possa receber ração como alimento, contudo para propósitos de especificação desse trabalho, será utilizado cães e gatos. Ambos os cães e gatos domésticos são membros da ordem Carnívora, o que indica espécies que se especializaram no hábito alimentar carnívoro e por isso apresentam anatomia peculiar. No entanto, pertencem a diferentes ramos da ordem e, conseqüentemente, têm herdado distintos legados de preferências alimentares e comportamento de seleção de alimentos [Saad and França 2010].

Isso significa que esses dois animais, consomem quantidades distintas de alimento em cada uma de suas refeições, como em que gatos comem quantidades diferentes de acordo com idade, e peso do felino, conforme apresentado na Tabela 1. Já os cachorros possuem uma vasta variedade de quantidade de ração ideal diária para consumo diário definido com base no peso do animal, conforme apresentado na Tabela 2.

**Tabela 1. Tabela de alimentação felina diária proposta por [Souza 2020].**

Idade	Peso	Quantidade (g)
2 a 3 meses	x	40 g a 50 g
3 a 4 meses	x	50 g a 60 g
4 a 6 meses	x	60 g a 70 g
6 a 12 meses	x	70 g a 80 g
Adulto Pequeno	3 a 4kg	40 g a 55 g
Adulto Grandes	5 a 6kg	55 g a 75 g
Idoso Pequeno	3 a 4kg	45 g a 60 g
Idoso Grande	5 a 6kg	60 g a 75 g

## 2. Objetivo

O objetivo desse trabalho é demonstrar a possibilidade de facilitar a vida cotidiana de humanos que possuem animais com o desenvolvimento de alimentadores automáticos interconectados em

**Tabela 2. Quantidade de ingestão diária que um cão deve comer com base no peso [Quiñonez et al. 2021].**

Size	Weight (Kg)	Grams	Size	Weight (Kg)	Grams
Miniature	1	38.94	Big	26	448.27
	2	65.49		27	661.27
	3	88.77		28	474.02
	4	110.15		29	466.66
	5	130.21		30	499.20
Small	6	149.29		31	511.62
	7	167.59		32	523.95
	8	185.25		33	536.19
	9	202.35		34	548.33
	10	218.99		35	560.38
Medium	11	235.22	Giant	36	572.34
	12	251.08		37	583.23
	13	266.62		38	596.03
	14	281.86		39	607.76
	15	296.82		40	619.41
	16	311.54		41	630.98
	17	326.04		42	642.49
	18	340.32		43	653.93
	19	354.40		44	665.30
	20	368.30		45	676.61
Big	21	382.03		46	687.86
	22	395.59		47	699.04
	23	409.00		48	710.17
	24	422.27		49	721.24
	25	435.40		50	732.25

um ambiente simulado. Esse ambiente simulado será criado no Cooja acessado por uma máquina virtual com o Contiki e consiste em 2 dispositivos de alimentação automática, configurados com valores diferentes em que um deles será representado o uso para um cachorro e outro para o gato.

Os animais utilizados de referência nesse estudo podem ser domésticos, a alimentação de animais doméstico pode ser um certo desconforto aos donos, nos casos em que o animal possui fome antes do horário que o seu dono levanta, e então começa a fazer barulhos para chamar a atenção e acordar o dono para receber comida.

Além disso, donos de animais domésticos podem necessitar ir ao trabalho fora de casa e deixar os bichanos em casa. Nesses momentos, ter um alimentador automático de animais, pode auxiliar muito em manter a alimentação dos animais em porções adequadas para o seu porte com base em sua idade e tamanho mesmo com os donos fora de casa, além de que com a utilização do dispositivo os cuidadores podem ficar tranquilos a respeito da alimentação dos seus bichanos.

### 3. Trabalhos Relacionados

Atualmente, existem alguns trabalhos relacionados a alimentadores automáticos como dispositivos programáveis para animais de estimação, alimentadores inteligentes para cães e sistemas inteligentes de cuidados para animais de estimação. Há também um projeto de sistema programável para fornecer alimentação aos animais de estimação em horários programados automaticamente, utilizando microcontroladores, LCD (*Liquid-Crystal Display*) e motores [Berhan et al. 2014].

Além disso, uma outra pesquisa com foco em controlar e monitorar alimentação de animais doméstico, especificamente cães, foi proposto por [Quiñonez et al. 2021] em que foi criado um alimentador automático de ração para cachorros em que o usuário do dispositivo (responsável pela alimentação do animal) pode configurar a forma de alimentação através da rede social do twitter ou com a utilização de GSM (*Global System for Mobile Communications*).

Outro trabalho relacionado teve a implementação realizada de um alimentador automático para pequenos animais utilizando canos PVC e um ATMEGA32 microcontrolador [Koley et al. 2021]. Entretanto, esse alimentador foi projetado apenas para pequenos animais,

e sua configuração é somente definida através de um pequeno teclado, em que é possível setar somente a quantidade de porções que deseja ser dispensada por dia.

Já [Ibrahim et al. 2019], criou um protótipo de alimentador automático para pequenos animais utilizando os seguintes componentes: Relógio em tempo real, um visor LCD para observar, um pequeno motor para abrir e fechar a porta que despeja ração, uma campainha para realizar alertas, um pequeno teclado para configuração e um interruptor de limites para escutar instruções. Este protótipo também utiliza da criação de um produto real utilizando microcontroladores e plásticos para criar um alimentador automático para animais de pequeno porte.

Por fim, o ultimo trabalho relacionado encontrado apresenta uma proposta de design para alimentadores automáticos que mantém contato com um servidor WEB e autenticação para controlar e observar os dados relativos ao alimentador automático [Babu et al. 2019]. Porém, de acordo com o especificado pelos dispositivos e materiais apresentados para construir o alimentador automático, a proposta também engloba apenas pequenas porções de alimento a ser distribuído durante o dia, o que leva a conclusão de que apenas pequenos animais são atendidos por essa proposta.

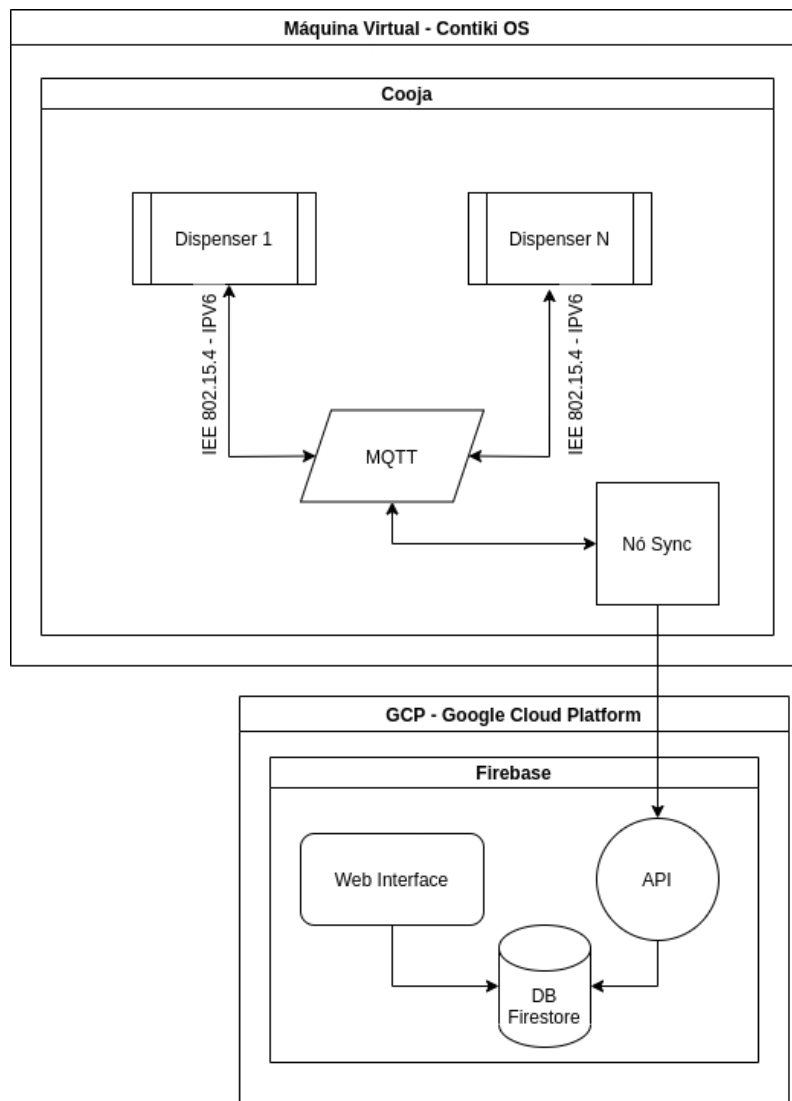
Contudo, o dispositivo proposto para alimentação automática de cachorros, tem sua configuração apenas para mamíferos caninos com aplicativo de seleção de acordo com o apresentado na Tabela 2, já o dispositivo no ambiente simulado apresentado nesse trabalho, deixa livre de determinação do usuário qual será o animal alimentado, não se limitando a apenas caninos, mas sim habilitando ao dono de qualquer animal que se alimenta diariamente de rações a possibilidade de configurar e utilizar o dispositivo do modo que mais se adapta ao estilo de vida do animal.

Além disso, inúmeros animais de pequeno e grande porte podem ser alimentados com o dispositivo da simulação proposta neste trabalho, necessitando apenas criar a configuração específica para cada animal. O acompanhamento dos dados relevantes do alimentador automático podem ser observados através de uma interface WEB (*World Wide Web*), disponível em: <https://animalfeeder-api-2wmdhpbuoq-uc.a.run.app/>.

## 4. Arquitetura

A arquitetura proposta consiste em utilizar de um ambiente simulado com uma máquina virtual com o Contiki no simulador Cooja com N dispositivos de alimentação automática que realiza publicações e subscrições em um *broker* do MQTT que contém um nó *sync* para realizar e requisições para o GCP na plataforma do Firebase através de uma API (*Application Programming Interface*) que então realiza a inserção de dados em uma banco de dados não relacional chamado de Firestore, que por fim pode ter seus dados visualizados pelo usuário através de uma interface pública na web disponibilizada pelo Firebase Hosting, conforme apresentado na Figura 1.

A plataforma utilizada no ambiente simulado Cooja tem seu código aberto para facilitar que demais pesquisadores a utilizem para fins acadêmicos ou testes de aparelhos de capacidades restritas como os de IoT. Já o Google Cloud Platform possui um ambiente mais empresarial em que diversas funcionalidades são pagas. Contudo, o GCP fornece incentivos para que pessoas iniciem a desenvolver com as suas propostas. Para o desenvolver desse trabalho, nenhum encargo foi realizado, mas para disponibilizar a aplicação WEB a todos, foi necessário incluir diversas informações e inserir os dados de cartão de crédito, para caso aconteça inúmeras requisições não cobertas pelo plano gratuito.



**Figura 1. Arquitetura proposta pelo autor.**

## 5. Desenvolvimento

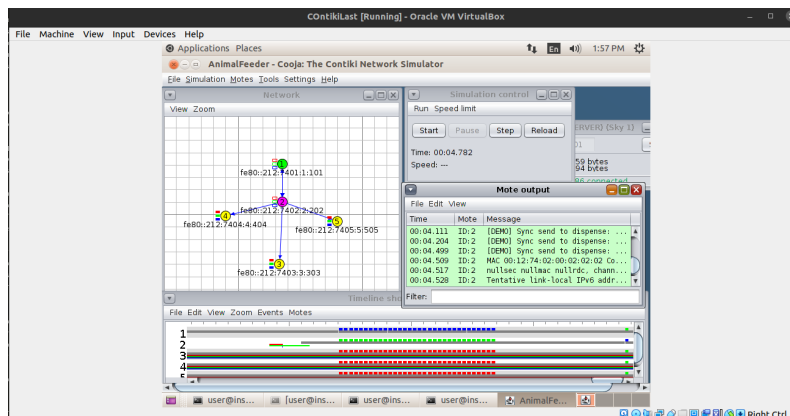
Para o desenvolvimento deste trabalho, várias tecnologias e linguagens de programação foram utilizadas, nesta seção será apresentada cada uma delas.

Começando pelo ambiente simulado do Cooja, para auxiliar a execução do simulador foi utilizado uma máquina virtual com algumas configurações já feitas com o sistema operacional Ubuntu 14.04.3 LTS - Contiki através de uma virtualização feita por uma aplicação chamada VirtualBox. Apesar do ambiente pré pronto para o desenvolvimento, alguns pacotes necessitaram ser instalados para a realização deste trabalho.

O pacote que possui a implementação do MQTT na versão SN(*Sensor Network*) disponível em <https://github.com/aignacio/mqtt-sn-tools> foi necessário ser instalado pois ele é utilizado na comunicação via publicação e subscrição entre os alimentadores e o nó sincronizador. Outro pacote utilizado de código aberto que precisou ser instalado foi o *libcurl* que é responsável por realizar requisições de dentro do simulador (Cooja) para o mundo externo (API do GCP), esses pacotes tiveram de ser instalados além dos já existentes no ambiente de simulação pré pronto.

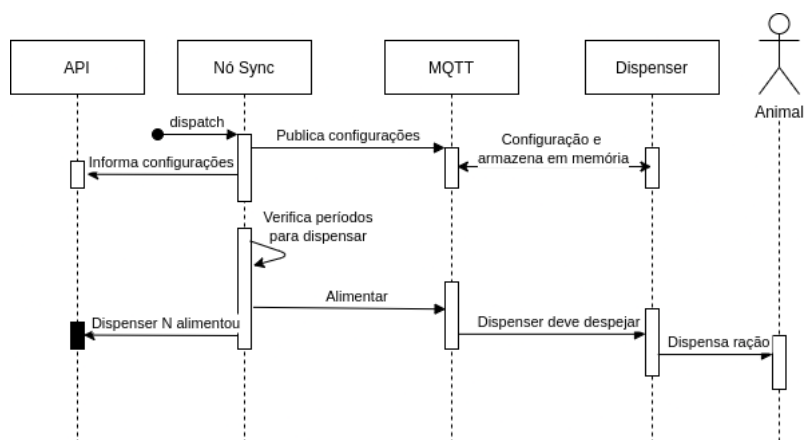
A simulação foi desenvolvida utilizando a linguagem de programação "C" em que foram criados o alimentador automático e também o nó sincronizador. Além disso, para a comunicação com IPV6 funcionar utilizando a rede de conexão sem fio IEEE 802.15.4, foi utilizado um mote "border-router" já disponível pelo próprio ambiente do Cooja.

Um exemplo do deste trabalho em funcionamento pode ser demonstrado através da utilização de 5 "Sky motes" em que o mote de identificador 1 é o "border-router", o identificador 2 é o nó sincronizador e os identificadores 3, 4 e 5 são alimentadores automáticos, os alimentadores automáticos foram configurados para três animais diferentes dois de pequeno porte e um de grande porte, o alimentador de identificador 3 é um cachorro, o identificador 4 é um gato e o identificador 5 é uma vaca e a sua representação gráfica pode ser observado na Figura 2.



**Figura 2. Simulação do alimentador automático no Cooja.**

O fluxo de funcionamento da comunicação entre o nó sincronizador e os demais alimentadores é ao iniciar os processos, os alimentadores automáticos de identificadores 3, 4 e 5 se inscrevem no tópico de configuração "/config", ao receber a configuração que o nó sincronizador de identificador 2 publicou nesse tópico eles armazenam esses em memória e já realizam a subscrição no tópico para realizar o despejo de ração "/dispenser", enquanto isso o nó sincronizador envia as configurações também à API para atualizar os dados no banco de dados, conforme representado no diagrama de sequência na Figura 3.



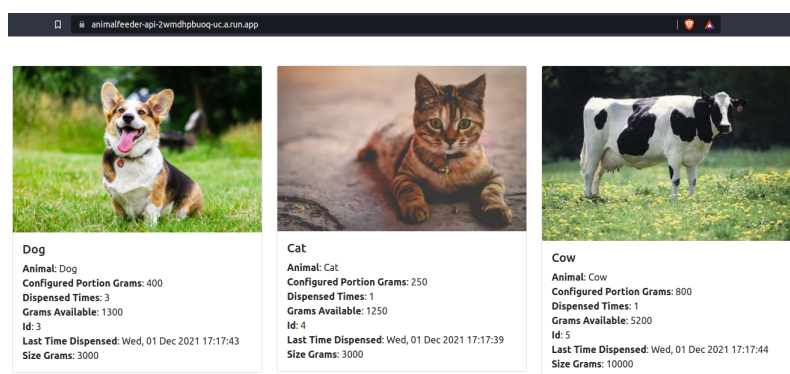
**Figura 3. Diagrama de sequência do controlador.**

Após ter a configuração realizada, o nó sincronizador começa a calcular o tempo para despejar para cada uma das configurações, e quando deve despejar é enviado uma mensagem

ao tópico `"/dispenser"` e também enviado a ordem de despejar ração para a API que atualiza os dados do banco. Os alimentadores automáticos ao receberem a mensagem no tópico `"/dispenser"`, realizam o despejo de ração de acordo com o configurado previamente para o animal.

Já para o desenvolvimento da aplicação da WEB, foi utilizado a linguagem de programação *TypeScript* que é basicamente *JavaScript* com tipagem de dados, o *TypeScript* compila para o *JavaScript* antes de ser executado, para desenvolver a API que recebe requisições do nó sincronizador e atualiza as informações no banco de dados. O banco de dados utilizando é o *Firestore* que é do tipo não relacional em que basicamente são armazenados os dados em formato JSON (*JavaScript Object Notation*).

Para a página WEB no *frontend* foi utilizado HTML (*HyperText Markup Language*) para montar a estrutura da página e o CSS (*Cascading Style Sheets*) para aplicar estilos e descrever a apresentação de um documento escrito em uma linguagem de marcação (HTML). A página é bem simples e consiste em apenas apresentar os dados relevantes dos alimentadores automáticos, então no caso do exemplo apresentado na Figura 2, a página WEB em algum momento da execução teria as seguintes informações: 3 objetos representado cada um dos animais, quantas vezes a ração foi despejada, o tamanho total da capacidade do alimentador, quantas gramas de ração são despejadas por vez, qual foi a ultima vez que a comida foi despejada e qual o identificador do alimentador conforme apresentado na Figura 4.



**Figura 4. Página WEB com as informações relevantes sobre alimentadores.**

## 6. Conclusão

A proposta de construir um alimentador automático de animais em um ambiente simulado foi realizado, é possível adicionar N alimentadores, apenas necessário adicionar uma configuração inicial no código do nó sincronizador para cada um dos novos alimentadores para começar a realizar o despejo automático de acordo com a configuração e começar a observar as informações relevantes de cada um dos alimentadores na página WEB.

Durante o desenvolver do trabalho, algumas dificuldades apareceram com instalar novos pacotes as funcionalidades já existentes do Cooja para realizar a comunicação do MQTT-SN e também a *libcurl* para realizar a requisição POST à API fora do ambiente simulado para a comunicação com internet. Além disso, desenvolver e testar aplicações dentro de uma máquina virtual tem os seus desafios e atrasa o desenvolvimento de código em relação a forma desenvolvimento convencional.

Este trabalho, apresentou a proposta de uma simulação de um alimentador automático em que existe um controlador para diversos aparelhos de alimentação automático de animais de diversos portes através da conexão de redes sem fio, mas partiu do pressuposto que esses aparelhos

compartilham a mesma interface de comunicação. Então, um possível trabalho futuro seria adicionar a funcionalidade de configuração dos alimentadores através da interface WEB e também analisar e realizar o design dos componentes necessários para que seja possível construir um alimentador automático que comporte diferentes portes de animais, e de fato integrar o controlador do ambiente simulado para a realidade.

## Referências

- Babu, B. R., Kumar, P. P., and Kuppusamy, P. (2019). Arduino mega based pet feeding automation. *IOSR Journal of Electronics and Communication Engineering*, 14(4):13–16.
- Berhan, T. G., Ahemed, W. T., and Birhan, T. Z. (2014). Programmable pet feeder. *Int. J. Sci. Eng. Res*, 3(11):99–104.
- Chaudhary, S., Johari, R., Bhatia, R., Gupta, K., and Bhatnagar, A. (2019). Craiot: concept, review and application (s) of iot. In *2019 4th international conference on internet of things: Smart innovation and usages (IoT-SIU)*, pages 1–4. IEEE.
- Ibrahim, M., Zakaria, H., and Xian, E. W. (2019). Pet food autofeeder by using arduino. In *IOP Conference Series: Materials Science and Engineering*, volume 670, page 012069. IOP Publishing.
- Koley, S., Srimani, S., Nandy, D., Pal, P., Biswas, S., and Sarkar, I. (2021). Smart pet feeder. In *Journal of Physics: Conference Series*, volume 1797, page 012018. IOP Publishing.
- Quiñonez, Y., Lizarraga, C., Aguayo, R., and Arredondo, D. (2021). Communication architecture based on iot technology to control and monitor pets feeding. *JUCS-Journal of Universal Computer Science*, 27:190.
- Saad, F. M. d. O. B. and França, J. (2010). Alimentação natural para cães e gatos. *Revista Brasileira de Zootecnia*, 39:52–59.
- Souza, L. (2020). Descubra como calcular a quantidade de ração para gato — petz. <https://www.petz.com.br/blog/nutricao/quantidade-de-racao-para-gato/>. (Accessed on 11/02/2021).