

**UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA DO MAR, CIÊNCIA E TECNOLOGIA
CURSO DE ENGENHARIA DE COMPUTAÇÃO
PROJETO DE SISTEMAS EMBARCADOS**

ALIMENTADOR DE GATOS POR ASSISTENTE DE VOZ

por

João Vitor Specht Kogut
Nicole Migliorini Magagnin

Itajaí (SC), novembro de 2022

**UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA DO MAR, CIÊNCIA E TECNOLOGIA
CURSO DE ENGENHARIA DE COMPUTAÇÃO
PROJETO DE SISTEMAS EMBARCADOS**

ALIMENTADOR DE GATOS POR ASSISTENTE DE VOZ

por

João Vitor Specht Kogut
Nicole Migliorini Magagnin

Relatório apresentado como requisito parcial da disciplina Projeto de Sistemas Embarcados do Curso de Engenharia de Computação para análise e aprovação.

Professores Responsáveis: Cesar Albenes Zeferino
Paulo Roberto Valim
Felipe Viel

Itajaí (SC), novembro de 2022

SUMÁRIO

1 INTRODUÇÃO.....	5
1.1 DEFINIÇÃO DO PROBLEMA.....	5
1.2 SOLUÇÕES EXISTENTES.....	5
1.3 SOLUÇÃO PROPOSTA	9
1.4 MERCADO	10
1.5 ANÁLISE DE VIABILIDADE	10
 2 PROJETO	 12
2.1 VISÃO GERAL.....	12
2.2 PREMISSAS.....	14
2.3 ANÁLISE DE REQUISITOS.....	15
2.4 ARQUITETURA DE HARDWARE.....	17
2.4.1 ESPCam.....	18
2.4.2 LM2596S.....	19
2.4.3 ULN2003	21
2.5 ARQUITETURA DE SOFTWARE.....	21
2.5.1 ASSISTENTE DE VOZ	22
2.5.2 DIAGRAMA DE CASOS DE USO.....	23
2.6 estrutura	28
2.7 LISTA DE MATERIAIS E ORÇAMENTO	29
2.8 PLANEJAMENTO	30
2.8.1 PLANO DE VERIFICAÇÃO DO DISPOSITIVO ALIMENTADOR.....	31
2.8.2 PLANO DE VERIFICAÇÃO DA ASSISTENTE DE VOZ.....	31
2.8.4 PLANO DE AVALIAÇÃO.....	31
2.9 CRONOGRAMA	32
2.10 ANÁLISE DE RISCOS.....	32
 3 DESENVOLVIMENTO	 33
3.1 IMPLEMENTAÇÃO	33
3.1.1 IMPLEMENTAÇÃO NO ESP32-CAM	33
3.1.2 IMPLEMENTAÇÃO NA ASSISTENTE DE VOZ	37
3.2 VERIFICAÇÃO.....	39
3.2.1 VERIFICAÇÃO DO DISPOSITIVO ALIMENTADOR.....	39
3.2.2 VERIFICAÇÃO DA ASSISTENTE DE VOZ.....	39
3.2.3 AVALIAÇÃO.....	40
 3 RESULTADOS	 41

4 CONSIDERAÇÕES FINAIS / CONCLUSÕES	44
--	-----------

1 INTRODUÇÃO

1.1 DEFINIÇÃO DO PROBLEMA

As pessoas na era da tecnologia tendem a ter a vida cada vez mais agitada e ocupada, seja com trabalho, estudos ou até mesmo lazer. Porém, para muitos ainda é indispensável a presença de um bichinho de estimação em sua casa.

Popularmente, gatos são ditos como independentes e que “podem passar o dia todo sozinhos”, sendo assim, até mesmo as pessoas com menos tempo livre tendem a adotar esse animal, o que gera uma responsabilidade as vezes maior do que o tutor do gato pode esperar. Pensando nisso, é proposto o alimentador de gatos por assistente de voz, que tem como objetivo auxiliar essas pessoas que mesmo muito ocupadas, ainda assim desejam ter um bichinho de estimação como companhia.

1.2 SOLUÇÕES EXISTENTES

As soluções no mercado de alimentadores inteligentes para cães e gatos ainda não são populares, tendo assim poucas opções em geral e ainda menos quando se trata da ativação por voz, nos itens abaixo são apresentadas as principais soluções relacionadas encontradas.

1.2.1 CAT FEEDER

O primeiro produto a chamar a atenção como concorrente em potencial ainda é um projeto científico chamado Cat Feeder. O Cat Feeder foi proposto por Anggraini et. al (2020) e seu protótipo utiliza um microcontrolador *Raspberry Pi*, além de um módulo de webcam para auxiliar ao usuário verificar quando o seu gato se aproxima do alimentador, para a partir de um botão por aplicativo mobile ele possa ativar a alimentação para o animalzinho. A ideia é que o usuário consiga visualizar o animal pedindo comida em qualquer lugar através do seu celular, via aplicativo conectado a *web server* e então, alimentar o gato a partir do clique de um botão. O produto Cat Feeder encontra-se apenas como um protótipo de pesquisa e não foi levado ao âmbito comercial até então.

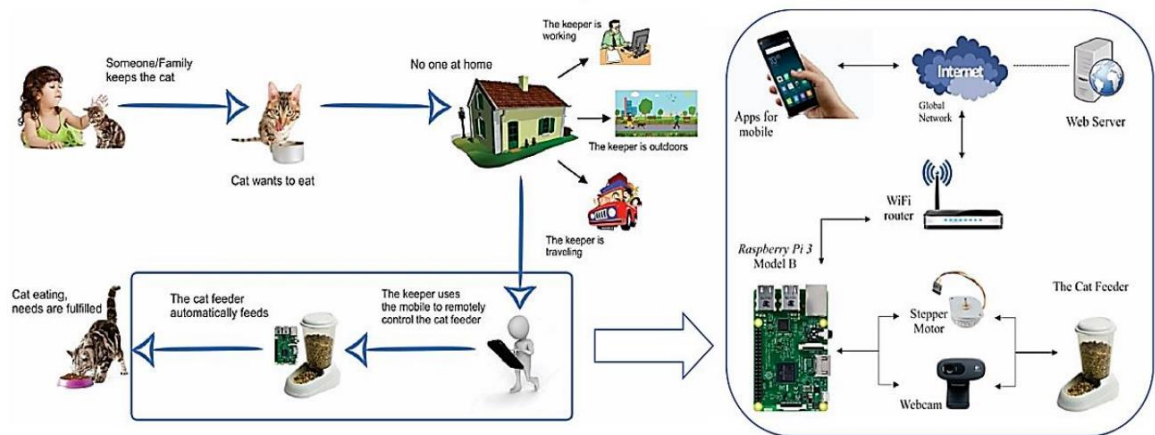


Figura 1 - Funcionamento Cat Feeder

1.2.2 CAT MATE C3000

O segundo alimentador para gatos inteligente a chamar atenção através de pesquisas de concorrentes foi o Cat Mate C3000, esse alimentador possui algumas versões e pode ser encontrado em sites de comércio popular como a Amazon, por cerca de 1.825,00 reais (AMAZON, 2022). O produto Cat Mate não funciona de maneira inteligente e sim automatizada, sendo possível programar horários para a alimentação do cão ou gato.



Figura 2 - Cat Mate C3000

1.2.3 ZEN PET FEEDER

Outro produto similar é o Zen Pet Feeder, um alimentador para cães e gatos de fabricação nacional que ainda não teve seu lançamento consolidado, mas possui patente registrada e pode ser reservado no próprio site da empresa (ZENPET, 2022). Existem três modelos do Zen Pet Feeder, sendo um deles uma fonte de água inteligente e dois alimentadores, um para cães de pequeno porte e outro para cães de médio e grande porte. Ao adquirir um dos

produtos, o usuário tem acesso ao Zen App, um aplicativo para controlar o dispositivo que permite o agendamento de horários e porções, interações com o *pet* por áudio e vídeo, notificações em tempo real, histórico de alimentação, interação em tempo real via webcam e agenda de saúde do animal, para cadastrar informações sobre vacinas, consultas, entre outras. O valor do produto não é publicamente divulgado.

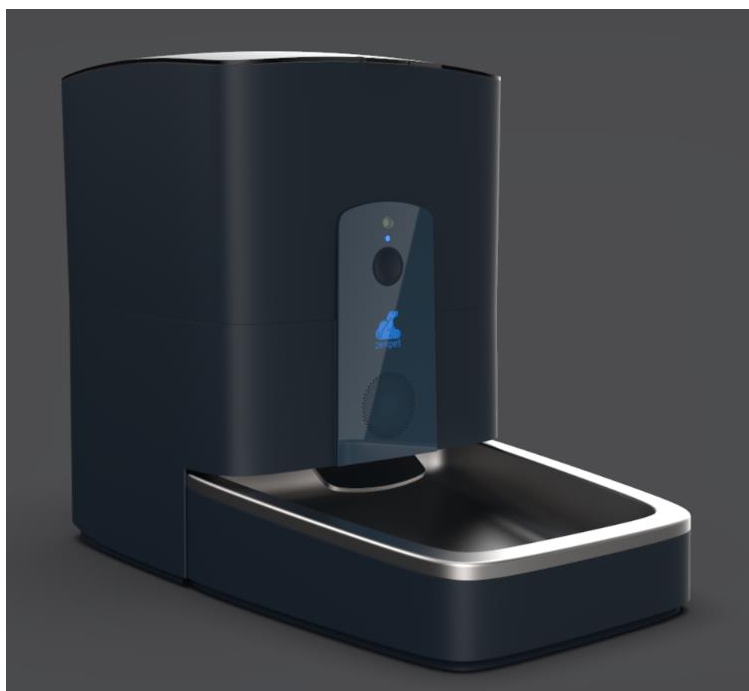


Figura 3 - Zen Pet Feeder

1.2.4 SURE FEED

O alimentador Sure Feed é um alimentador inteligente por microchip, o que significa que ele tem seu funcionamento baseado no microchip presente na coleira do animal, que irá liberar o alimentador ao se aproximar e ser compatível. Assim, o animal pode se alimentar quando desejar e outros animais não poderão chegar ao alimento, incluindo outros gatos ou até mesmo outras espécies. O Sure Feed permite o cadastro de até 32 animais diferentes e permite ao tutor selecionar também o número de refeições diárias disponíveis. O produto se encontra no site Pet Love e custa R\$ 989,91 reais a vista (PET LOVE, 2022).



Figura 4 - Sure Feed

1.2.5 ALIMENTADOR POR VOZ AUTOMÁTICO

Foi possível localizar em buscas online apenas um alimentador por voz e assistente virtual Alexa, atualmente a venda, esse alimentador genérico pode ser encontrado em sites de venda de produtos vindos da China, como Aliexpress e Shopee e custa cerca de R\$450,00, ele permite também tirar fotos e gravar vídeos do *pet* a distância. O alimentador também possui um sensor infravermelho para evitar que a ração transborde no pote. Entretanto, produto não possui avaliações ou compradores (SHOPEE, 2022).



Figura 5 - Alimentador por voz genérico

1.3 SOLUÇÃO PROPOSTA

A solução proposta por este projeto se dá na criação e implementação de um alimentador para cães e gatos, porém com foco comercial nos felinos, que possua conexão com a assistente de voz Alexa para proporcionar melhor uso do alimentador. O alimentador também terá uma câmera embutida com a proposta de fotografar o animal no ato da alimentação e enviar ao dono e uma opção de porcionar a ração, permitindo que o tutor tenha certeza de que seu bichinho está se alimentando.

Diferente da maioria dos produtos encontrados no mercado por utilizar a assistente de voz Alexa como ativação, o produto também se difere do alimentador por voz encontrado pois terá seu funcionamento consolidado através de testes, tornando-se mais confiável, uma vez que a única alternativa do mercado com esse tipo de ativação não possui dados.

Quadro 1 – Comparativo da solução proposta com as soluções existentes

Nome	Inteligente	Ativação por voz	Ativação por aplicativo	Possui câmera
Car Feeder	Sim	Não	Sim	Sim
Cat Mate	Não	Não	Não	Não
Zen Pet	Sim	Não	Sim	Sim
Sure Feeder	Sim	Não	Não	Não
Alimentador com ativação por voz	Sim	Sim	Não	Sim
Solução proposta	Sim	Sim	Não	Sim

1.4 MERCADO

O mercado Pet é um grande atrativo comercial, uma vez que bichinhos de estimação são a paixão de muitas pessoas. Segundo a Associação Brasileira da Indústria de Produtos Para Animais de Estimação (ABINPET) o Brasil possui a segunda maior população mundial de cães, gatos e aves canoras e ornamentais e a terceira maior população mundial em questão de todos os tipos de animais (ABINPET, 2022). Sendo assim, o nicho de mercado em questão apresenta grande movimento econômico, tendo movimentado cerca de R\$ 20,3 bilhões no ano de 2018, sendo esse valor quase sete vezes maior do que a movimentação no ano de 2006, essa de R\$ 3,3 bilhões.

Ao mesmo tempo que o mercado Pet está sempre em ascensão, o mercado de produtos inteligentes e que utilizam tecnologias IoT (internet das coisas) também está. Estimativas preveem a movimentação de cerca de R\$ 8,5 bilhões em dispositivos e funcionalidades abrangendo a internet das coisas (“IoT: um mercado de R\$ 8,5 bilhões no Brasil”, 2022).

Sendo assim, ao reunir dois mercados muito fortes e em constante crescimento, é esperado que o produto sugerido por este projeto seja bem aceito pelo mercado e consiga entrar em cena em um bom momento, onde se tornará uma inovação e possua poucos concorrentes diretos.

1.5 ANÁLISE DE VIABILIDADE

A execução do presente trabalho dentro dos prazos é viável, uma vez que foram apresentados conteúdos de base para a implementação durante a disciplina e inclusive foram realizados trabalhos semelhantes, permitindo dessa maneira que houvesse uma base pré-definida para o presente trabalho.

Quanto ao protótipo de prova, todos os componentes já se encontram com os alunos e há tempo hábil para a execução e testes, garantindo assim um bom funcionamento.

2 PROJETO

A seção 2 é responsável por apresentar o projeto do protótipo de alimentador inteligente por comandos de voz. Nesta seção é possível encontrar uma visão geral na subseção 2.1, as premissas na seção 2.2 e a análise de requisitos na seção 2.3. A arquitetura de hardware se encontra na seção 2.4 e a arquitetura de software na seção 2.5, por fim, a estrutura do protótipo está na seção 2.6.

A seção 2.7 se refere a lista de materiais e a partir da seção 2.8 está o planejamento para o desenvolvimento do protótipo, incluindo cronograma (seção 2.9) e análise de risco (seção 2.10).

2.1 VISÃO GERAL

O produto será composto por um microcontrolador ESP32 contendo o módulo de câmera, esse será ligado a um motor de passo responsável por mover os componentes que serão impressos em filamento 3D, para realizar a alimentação do felino. O dispositivo também terá possibilidade de conectar-se com a assistente de voz Alexa, utilizando um banco de dados em tempo real Firebase para interação de comandos, fazendo com que a Alexa envie a ação de alimentar e o número de porções ao Firebase e o ESP32 seja capaz de ler os comandos e executá-los.

Após a alimentação, será enviado um e-mail ao tutor do animal utilizando *SMTP Server* e permitindo que ele possa visualizar imagens do ponto de vista do alimentador alguns segundos após a disponibilização de ração, uma vez que é importante ao tutor saber se seu animal está se alimentando corretamente.

No diagrama de blocos abaixo é possível visualizar uma visão geral do alimentador de gatos e de sua implementação futura. A conexão por voz pode ser realizada via dispositivo Alexa echo dot ou aplicativo *mobile* Alexa.



Figura 6 - Visão geral

Quanto a parte da assistente de voz, será possível realizar o porcionamento da ração de acordo com a necessidade do animal, além de solicitar horários de alimentação, funções já previstas na Alexa para suas atividades pré-definidas.

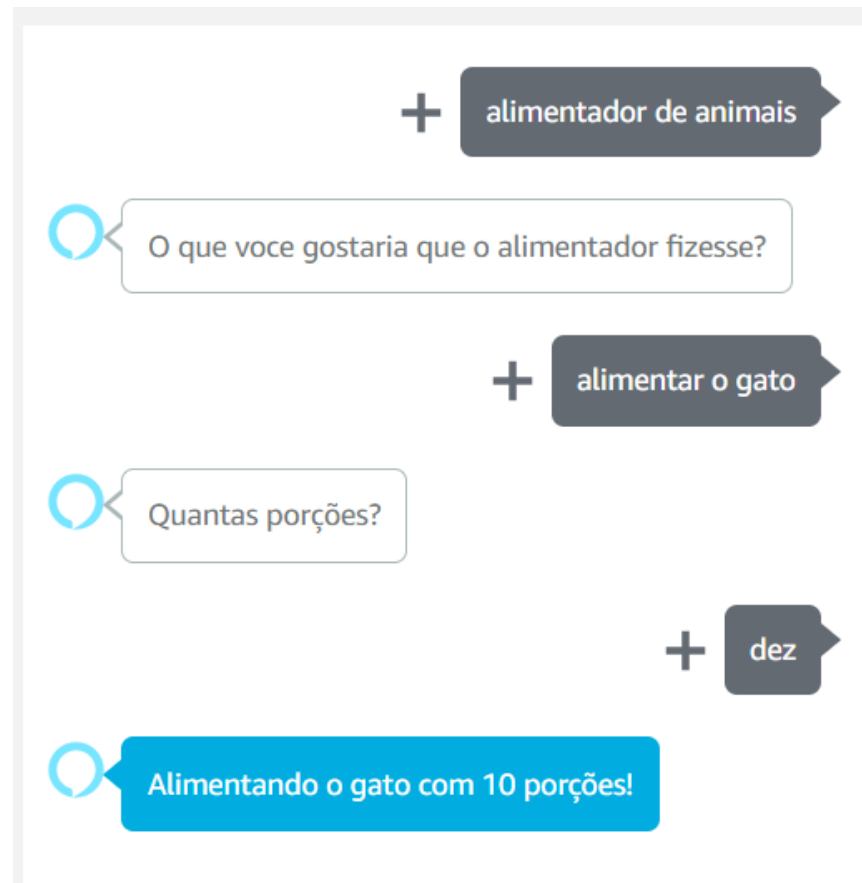


Figura 7 - Exemplo de interação com a Alexa

2.2 PREMISSAS

As afirmações a seguir são as premissas para definir o andamento deste projeto.

2.2.1 Premissas para o dispositivo alimentador

- Deverá ser utilizado o microcontrolador ESP-32 com módulo ESPCam.
- O software deverá ser escrito em linguagem C.
- O microcontrolador será responsável por realizar a conexão com a assistente de voz Alexa.
- Serão utilizados os dois núcleos do microcontrolador, realizando *multithreading*.
- O microcontrolador deverá interagir com a assistente de voz Alexa e com a aplicação *mobile* Alexa.
- O microcontrolador deverá interagir com um banco de dados em tempo real no Firebase.
- O microcontrolador deverá ser conectado via wi-fi.

2.2.2 Premissas para a assistente de voz

- A assistente de voz utilizada deverá ser uma Alexa Echo Dot ou em um *smartphone*.
- O Alexa Developer Console deverá ser utilizado para realizar a interação do protótipo com a assistente.
- A assistente de voz deverá ser capaz de agendar horários e rotinas de alimentação.
- A assistente de voz deverá ser capaz de ligar o alimentador.

2.3 ANÁLISE DE REQUISITOS

2.3.1 Requisitos funcionais do dispositivo alimentador

- RF01: O sistema deverá ativar a câmera após 20 segundos da execução do alimentador.
- RF02: O sistema deverá ter a divisão de tarefas entre busca por comandos para execução no Firebase em um núcleo e execução do motor em outro;
- RF04: O sistema deverá conectar-se a um servidor SMTP;
- RF03: O sistema deverá permitir ao usuário a alimentação imediata do pet;

- RF05: O sistema deverá permitir o envio de fotos via servidor SMTP para um e-mail;
- RF06: O sistema deverá permitir a escolha de número de porções para alimentar o pet;
- RF07: O sistema deverá se conectar com a assistente de voz.

2.3.2 Requisitos não funcionais do dispositivo alimentador

- RNF01: O sistema será prototipado no microcontrolador ESP32;
- RNF02: O sistema será prototipado no kit de desenvolvimento ESP32 dev kit;
- RNF03: O código da aplicação será escrito em linguagem C;
- RNF04: Será utilizado o ambiente de desenvolvimento Arduino IDE para programação no microcontrolador;
- RNF05: Deverá ser utilizado um módulo de câmera do ESP32;
- RNF06: O sistema deverá ser *multithreading*;
- RNF07: O sistema deverá permitir a conexão wi-fi.
- RNF08: O protótipo deverá custar no máximo R\$300,00;
- RNF09: Será usado um motor de passo para realizar os comandos mecânicos do alimentador;
- RNF10: O protótipo deverá ter no máximo 900 gramas;
- RNF11: O protótipo deverá ter no máximo 1000 mm, somadas as suas três dimensões físicas (largura, altura e profundidade);
- RNF12: O protótipo deverá apresentar um tempo de resposta máximo de 5 segundos;
- RNF13: A parte física do alimentador será impressa em 3D;
- RNF14: O banco de dados deverá ser programado para o Firebase.

2.3.3 Regras de negócio do dispositivo alimentador

- RN01: O produto deverá ter foco de mercado na qualidade de vida do animal de estimação;
- RN02: O sistema deverá poder ser conectado em qualquer rede Wi-Fi, permitindo ao cliente seu uso em qualquer lugar;

- RN03: O sistema deverá rodar em dois núcleos do microcontrolador para a obtenção de um bom produto.

2.3.4 Requisitos funcionais da assistente de voz

- RF01: O sistema do alimentador deverá permitir a interação com a assistente de voz Alexa do usuário via aplicação *mobile*;
- RF02: O sistema do alimentador deverá permitir a interação com a assistente de voz Alexa via dispositivo echo dot;
- RF03: O sistema deverá permitir o agendamento de horários para a alimentação via assistente de voz;
- RF04: O sistema deverá permitir ao usuário solicitar o número de porções para alimentar o gato via comando de voz;
- RF05: A Alexa deverá enviar dados de comando ao banco de dados em tempo real no Firebase.

2.3.5 Requisitos não funcionais da assistente de voz

- RNF01: Deverá ser utilizada a ferramenta Alexa Developer Console para o desenvolvimento da conexão com a Alexa;
- RNF02: Deverá ser utilizado o aplicativo mobile Alexa para testes iniciais;
- RNF03: Deverá ser utilizada uma Alexa Echo Dot terceira geração para os testes finais;
- RNF04: A programação no Alexa Developer Console deve ser realizada na linguagem Python.

2.3.6 Regras de negócio da assistente de voz

- RN01: O usuário deverá conseguir conectar o dispositivo a assistente de voz de maneira demonstrada no manual do produto;

2.4 ARQUITETURA DE HARDWARE

O hardware do alimentador de gatos é composto por um ESP32 de módulo câmera, denominado ESPCam, que é o núcleo principal do projeto. Esse microcontrolador possui dois

núcleos que serão utilizados. A nível de projeto, o ESP32 será conectado a um motor de passo ULN2003 com um driver 28BYJ-48 que irá ser responsável pela movimentação do alimentador, juntamente com uma polia e uma correia. A alimentação externa do protótipo será de 12V, passando por um conversor LM2596S, que é capaz de transformar esses 12V de entrada em 5V, sendo uma entrada segura para o microcontrolador.

No diagrama da figura 8 são representados os componentes de hardware e suas ligações, além de uma representação da conexão via wi-fi a ser realizada pelo ESP32 com os dispositivos externos, o aplicativo *mobile* Alexa e a Echo dot.

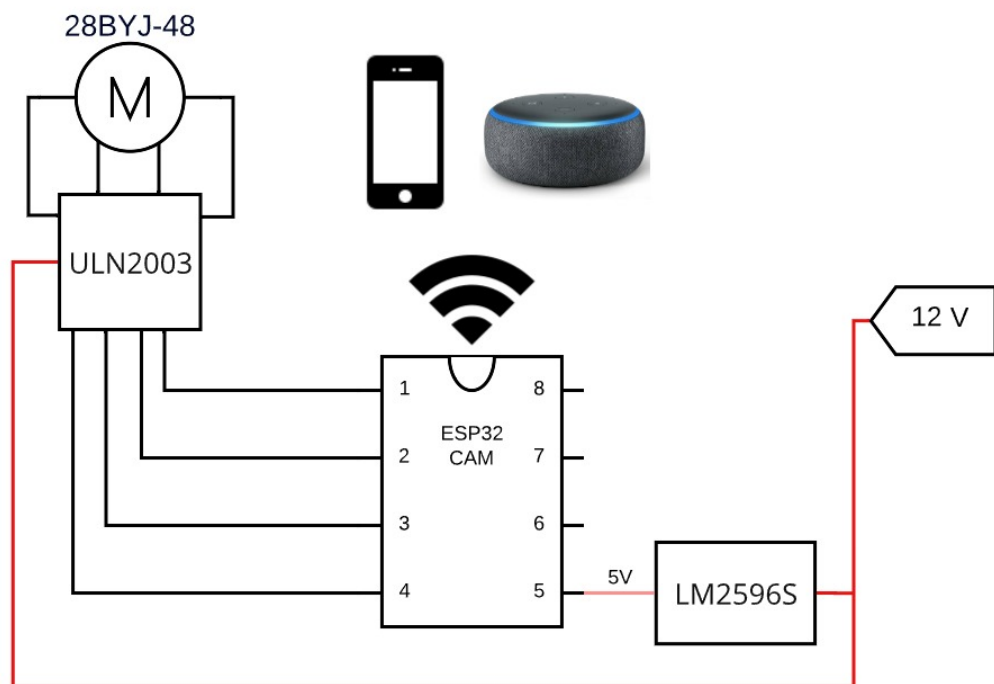


Figura 8 - Diagrama de Hardware

Nas subseções abaixo são realizadas descrições mais detalhadas dos principais componentes do projeto.

2.4.1 ESPCam

O ESP32 Cam é uma placa de desenvolvimento com câmera embutida de baixo custo. Essa placa possibilita a integração com Wi-Fi e Bluetooth e possui duas CPUs LX6 de 32-bits e alta performance. A ESP32 Cam também possui um *clock* de até 160MHz, 520KB de SRam

e um 4MPSRAM externo, além do suporte a *upload* de imagens via wi-fi, múltiplos modos de desligamento, sistema operacional FreeRTOS e LWIP embarcados e suportes UART/SPI/I2C/PWM/ADC e DAC (“ESP32-Cam Development board”, 2019).

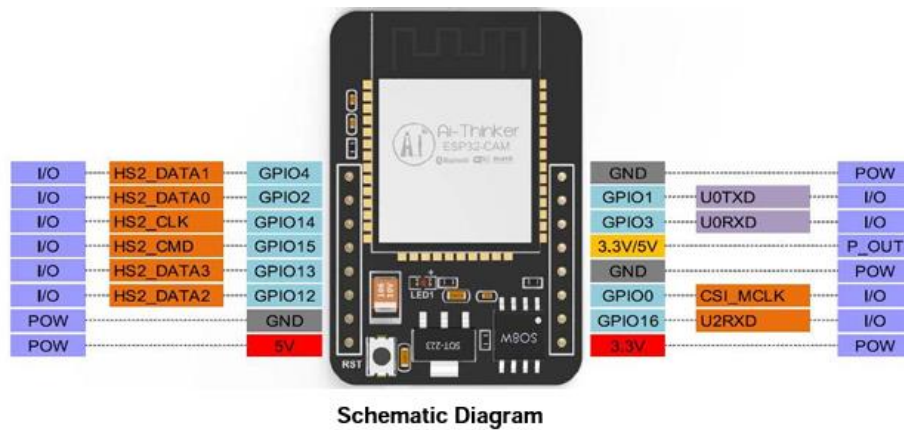


Figura 9 - Pinout ESP32 Cam

A fonte de energia do ESP32-Cam é de 5V e sua temperatura de operação varia entre - 20 graus celsius e 85 graus celsius. Seu peso é de apenas 10 gramas e sua câmera possui um flash embutido de 180mA.



Figura 10 - ESP32-Cam

2.4.2 LM2596S

O LM2596S é um regulador de tensão DC-para-DC capaz de converter uma entrada de 12V para valores de 3.3V, 5V ou 12V de maneira fixa e outras saídas customizáveis, da maneira que seu usuário preferir entre 1.2V e 37V com uma margem de erro de 4%. Seu intervalo de

tensão é de 40V e seu oscilador interno tem uma frequência de 150kHz. O conversor também possui proteção térmica de desligamento e corrente (WILSON, 2020).

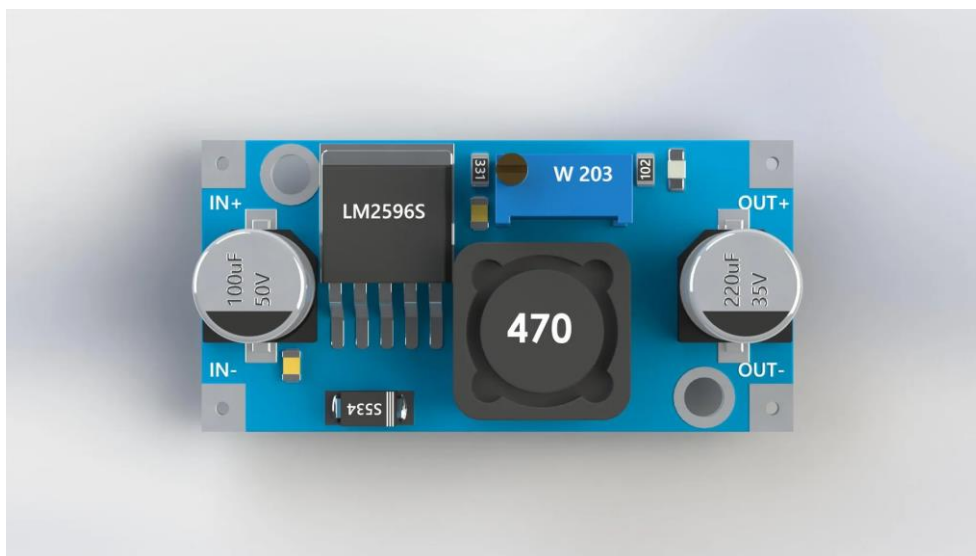


Figura 11 - Conversor LM2596S

2.4.3 ULN2003

O ULN2003 é um driver que permite a integração entre o motor de passo e um microcontrolador, utilizando quatro entradas para conexão. Ele possui um ângulo de passo de 1/64 ou 5.625 graus e uma tensão de 5V (WIKI, 2022).



Figura 12 - Motor de passo e driver

2.5 ARQUITETURA DE SOFTWARE

O software a ser implementado para a consolidação do alimentador de gatos visa a utilização dos dois núcleos do ESP32, incluir o módulo de câmera integrado e realizar a implementação em sua totalidade utilizando a Arduino IDE e a linguagem C. A separação do sistema em dois núcleos tem como objetivo a separação de tarefas, fazendo que um dos núcleos do microcontrolador verifique sempre se há comandos para executar e o outro núcleo os execute. O sistema do alimentador de gatos pode ser operado via assistente de voz e via aplicação *mobile* Alexa. Sendo assim, as seções a seguir descrevem o funcionamento de cada um dos módulos de funcionamento.

2.5.1 ASSISTENTE DE VOZ

Ao utilizar da aplicação do alimentador de gatos utilizando a assistente de voz, o usuário irá solicitar que a Alexa ative o alimentador através dos dizeres “Alexa, ativar alimentador”, posteriormente a assistente pedirá quantas porções serão dadas ao pet e ativará o alimentador. Esse módulo pode ser utilizado com a assistente de voz física ou pelo aplicativo da assistente no celular, uma vez que ambos funcionam da mesma maneira.



Figura 13 - Interação com a assistente de voz

A interação com a assistente de voz é realizada via envio de comandos ao banco de dados de tempo real Firebase, assim com uma *flag* de execução em 0, o microcontrolador irá

coletar os dados do banco de dados e realizar a execução, colocando a *flag* em 1 para que seja apenas executado uma vez. Assim, o alimentador irá ser ativado e fornecer alimento ao gato nas porções solicitadas e após vinte segundos tirar uma foto do animal ou do que estiver em frente a câmera. A foto será enviada via e-mail utilizando SMTP *server*.

A assistente de voz permite criar rotinas de horários de alimentação e realizar a tarefa todo dia igualmente, o alimentador também pode participar dessas rotinas, uma vez que elas são ferramentas da Alexa.

2.5.2 DIAGRAMA DE CASOS DE USO

Levando em conta as opções do sistema e sua complexidade, foi definida como uma boa maneira de demonstrar seu funcionamento um diagrama de casos de uso. No diagrama abaixo é possível visualizar os atores Usuário, ESPCam, Alimentador, Assistente de e suas tarefas nesse sistema.

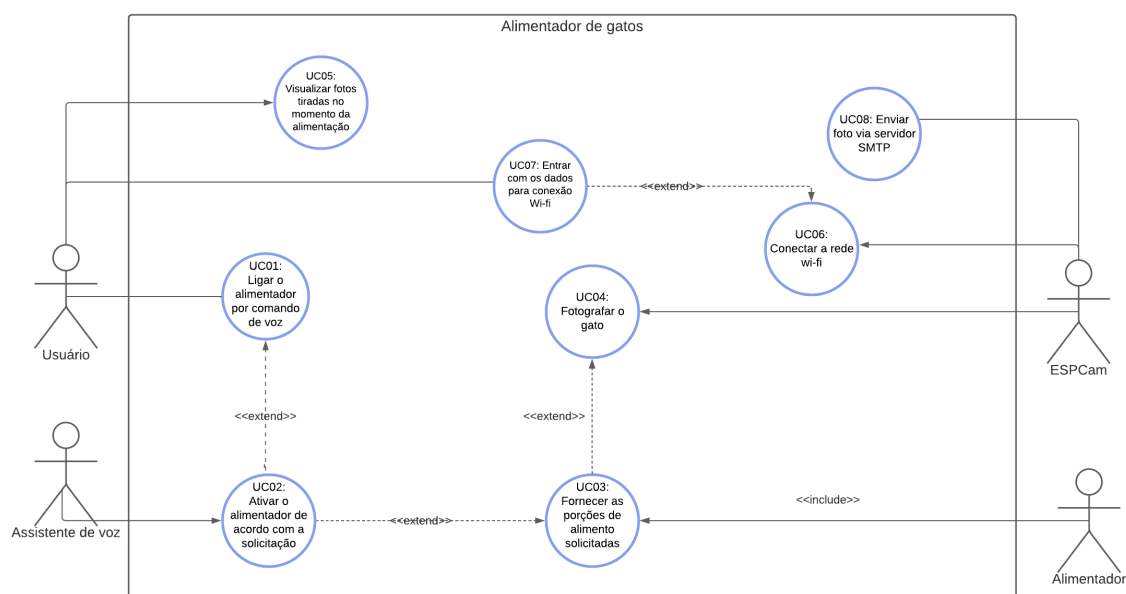


Figura 14 - Diagrama de casos de uso

Quadro 2 - UC01

Caso de uso	Nome do caso de uso
UC01	Ligar o alimentador por comando de voz
Pré-condição	-
Pós-condição	UC02

Requisito	RF01 e RF02 – Assistente de voz
Objetivos	Permitir o uso do alimentador via assistente de voz.
Atores	Usuário
Fluxo principal	O usuário solicita a alimentação utilizando a assistente de voz e ela o responde, perguntando quantas porções.
Fluxo alternado	-
Fluxo de exceção	-

Quadro 3 - UC02

Caso de uso	Nome do caso de uso
UC02	Ativar o alimentador de acordo com a solicitação
Pré-condição	UC01
Pós-condição	UC03
Requisito	RF03 – Assistente de voz
Objetivos	Perguntar o número de porções ao usuário e ativar o alimentador.
Atores	Assistente de voz
Fluxo principal	O usuário pede o número de porções e o alimentador é ativado.
Fluxo alternado	-
Fluxo de exceção	-

Quadro 4 - UC03

Caso de uso	Nome do caso de uso
UC03	Fornecer as porções de alimento solicitadas
Pré-condição	UC02
Pós-condição	UC04
Requisito	RF06 – Dispositivo alimentador

Objetivos	Alimentar o pet no número de porções selecionadas.
Atores	Alimentador
Fluxo principal	O alimentador fornece a quantidade certa de alimento e o gato se alimenta.
Fluxo alternado	-
Fluxo de exceção	-

Quadro 5 - UC04

Caso de uso	Nome do caso de uso
UC04	Fotografar o gato
Pré-condição	UC03
Pós-condição	UC08
Requisito	RF01 – Dispositivo alimentador
Objetivos	Fotografar o pet após 20 segundos para que o tutor consiga ver se ele foi até o prato se alimentar.
Atores	ESPCam
Fluxo principal	O dispositivo tira foto e o gato aparece se alimentando, posteriormente envia via <i>SMTP server</i> .
Fluxo alternado	O dispositivo tira foto e o gato não aparece se alimentando, posteriormente envia via <i>SMTP server</i> .
Fluxo de exceção	-

Quadro 6 - UC05

Caso de uso	Nome do caso de uso
UC05	Visualizar fotos tiradas no momento da alimentação
Pré-condição	UC04

Pós-condição	-
Requisito	RF05 – Alimentador
Objetivos	Mostrar ao usuário as imagens do animal se alimentando.
Atores	Usuário
Fluxo principal	O dispositivo fotografa o gato, envia as fotos via <i>SMTP server</i> e o usuário as visualiza em um e-mail
Fluxo alternado	-
Fluxo de exceção	-

Quadro 7 - UC06

Caso de uso	Nome do caso de uso
UC06	Conectar à rede wi-fi
Pré-condição	UC07
Pós-condição	-
Requisito	RNF07 – Dispositivo alimentador
Objetivos	Conectar com a rede wi-fi e permitir o uso do alimentador.
Atores	ESPCam
Fluxo principal	O usuário se conecta à rede wi-fi e é permitido o uso do alimentador.
Fluxo alternado	O usuário não se conecta à rede wi-fi e é necessário tentar novamente.
Fluxo de exceção	-

Quadro 8 – UC07

Caso de uso	Nome do caso de uso
UC07	Entrar com os dados para conexão Wi-fi
Pré-condição	-

Pós-condição	UC09
Requisito	RNF07 – Dispositivo alimentador
Objetivos	Conectar com a rede wi-fi e permitir o uso do alimentador.
Atores	Usuário
Fluxo principal	O usuário se conecta à rede wi-fi e é permitido o uso do alimentador.
Fluxo alternado	O usuário não se conecta à rede wi-fi e é necessário tentar novamente.
Fluxo de exceção	Conectar com a rede wi-fi e permitir o uso do alimentador.

Quadro 9 – UC08

Caso de uso	Nome do caso de uso
UC12	Enviar foto via servidor SMTP
Pré-condição	UC04
Pós-condição	UC05
Requisito	RF05 – Dispositivo alimentador
Objetivos	Permitir ao usuário a visualização das imagens via discord ou aplicativo.
Atores	ESPCam
Fluxo principal	O dispositivo fotografa o gato, envia as fotos via <i>SMTP server</i> e o usuário as visualiza em um e-mail.
Fluxo alternado	-
Fluxo de exceção	-

2.6 ESTRUTURA

A estrutura do alimentador será inteiramente impressa em 3D, com o design desenvolvido no *software Fusion*. A estrutura conta com uma base para sustentação, seguida pelo módulo que irá realizar o porcionamento de acordo com o número de porções solicitado. O tambor de porcionamento irá girar para alinhar o furo quadrado com o da tampa de cima para recolher a quantidade de ração e posteriormente, gira para alinhar com a base, fazendo com que seja disponibilizado o alimento em um pote externo ao alimentador.

O porcionador possui 4cm X 2,5cm X 1,9cm e corresponde a uma porção, o usuário poderá escolher o número de porções que desejar. O *dispenser* de ração terá o mesmo tamanho que o pote porcionador, uma vez que ele servirá para derrubar a ração para que seja medida sua quantidade.

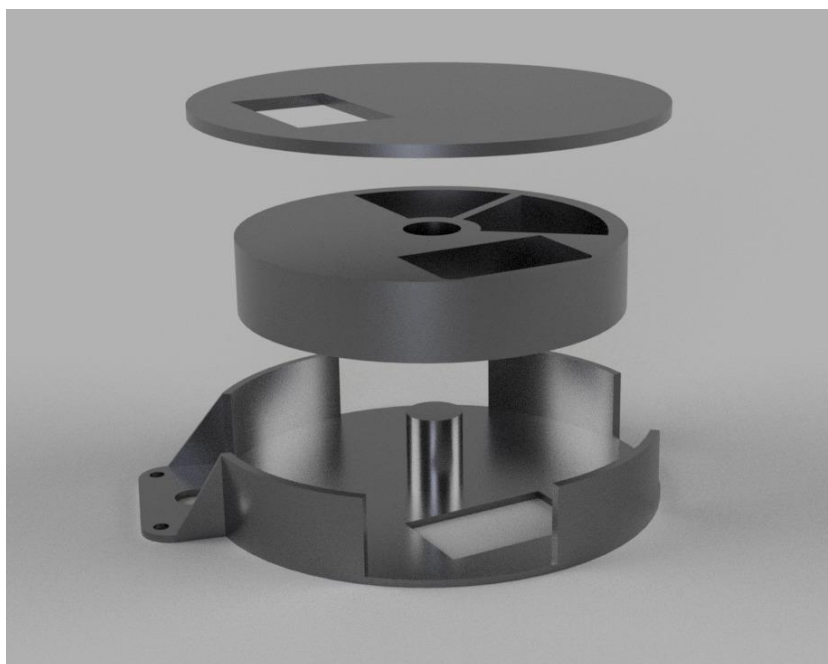


Figura 15 - Design do alimentador em 3D

Na impressão 3D abaixo, é possível visualizar o módulo de porcionamento e a base, com o motor de passo já acoplado.



Figura 16 - Impressão do módulo de porcionamento

2.7 LISTA DE MATERIAIS E ORÇAMENTO

Nesta seção serão apresentados os materiais a serem utilizados para a implementação do alimentador para gatos por assistente de voz, incluindo componentes eletrônicos, microcontrolador e componentes físicos impressos em filamento 3D. Será usado um microcontrolador ESPCam, com um adaptador FTDI para possibilitar a programação do microcontrolador e um motor de passo para permitir o funcionamento do alimentador. Uma correia e uma polia serão usadas para girar a peça principal do alimentador, separando porções e colocando no pote do alimentador.

Quadro 10 – Lista de materiais

Item	Qtd	Disponível	Responsável(is)
Kit de desenvolvimento ESP32 com ESP cam	01	Sim	Alunos
Adaptador FTDI	01	Sim	Alunos
Motor de passo 28BYJ-48 com driver ULN2003	01	Sim	Alunos
Polia 20 dentes para eixo 5mm	01	Sim	Alunos
Correia GT2	01	Sim	Alunos
Impressão 3D	01	Sim	Alunos
Cabo USB tipo mini B	01	Sim	Professores

2.8 PLANEJAMENTO

Nesta seção é apresentado o plano de verificação e testes do protótipo de prova, de acordo com seus requisitos. Será demonstrado em duas tabelas, uma para o dispositivo alimentador e outra para a assistente de voz.

2.8.1 PLANO DE VERIFICAÇÃO DO DISPOSITIVO ALIMENTADOR

Quadro 15 - Plano de verificação do dispositivo alimentador

Requisito	Procedimento de verificação/teste	Resultado esperado
RF01	Cronometrar o tempo de acionamento da câmera	Câmera acionada após 20 segundos
RF02	Verificação da execução das tarefas de software	Tarefas executadas em dois núcleos
RF03	Verificar se a solicitação de alimentação sem agendamento é funcional	Alimentação imediata permitida
RF04	Verificar o envio e recebimento das fotos	Fotos recebidas via e-mail
RF05	Verificar o envio e recebimento das fotos	Fotos recebidas via e-mail
RF06	Selecionar um número de porções e conferir seu atendimento	Número de porções igual ao requisitado
RF07	Testar a conexão de voz via Alexa	Conexão funcional

2.8.2 PLANO DE VERIFICAÇÃO DA ASSISTENTE DE VOZ

Quadro 16 – Plano de verificação da assistente de voz

Requisito	Procedimento de verificação/teste	Resultado esperado
RF01	Testar a interação com Alexa via App	O alimentador responde a comandos da Alexa via App
RF02	Testar a interação com Alexa via echo dot	O alimentador responde a comandos da Alexa via echo dot
RF03	Testar a função de agendamento da Alexa para o protótipo	A Alexa realiza agendamentos de alimentação
RF04	Solicitar à Alexa um número de porções e aguardar resposta	Número de porções servido ser igual ao solicitado
RF05	Verificar se os dados são enviados no Firebase	Firebase recebendo dados em tempo real

2.8.4 PLANO DE AVALIAÇÃO

Na tabela abaixo são demonstrados os meios de avaliação do protótipo em termos de funcionamento eletrônico.

Quadro 17 – Plano de avaliação

Métrica	Procedimento de avaliação	Resultado esperado
Corrente	Multímetro	< 1 ^a
Potência	Cálculos utilizando a fórmula Corrente x Tensão	Menos de 1W
Energia	Multímetro	5V
Tempo de resposta	Teste de software	Menos de 5s
Peso	Uso de balança	900g
Altura	Uso de trena	200 mm
Largura	Uso de trena	200 mm
Profundidade	Uso de trena	200mm

2.9 CRONOGRAMA

Na tabela abaixo é estruturado um cronograma para a entrega do protótipo de prova funcional e sua documentação.

Quadro 18 – Cronograma de execução

Atividade	Sem. 1	Sem. 2	Sem. 3	Sem. 4	Sem. 5	Entregável
Projeto	X	X				07/11
Implementação		X	X	X		28/11
Verificação				X	X	28/11
Avaliação				X	X	28/11
Documentação	X	X	X	X	X	28/11

2.10 ANÁLISE DE RISCOS

Nesta seção são analisados os possíveis riscos que podem afetar a entrega deste trabalho dentro do prazo e em sua total integridade. Os riscos são divididos entre suas probabilidades e impacto, podendo ser essas altas, médias ou baixas. Há sempre um gatilho para o risco e a equipe se faz responsável a um plano de contingência.

Quadro 19 – Análise de riscos

Risco	Probabilidade	Impacto	Gatilho	Plano de contingência
Não ser possível a integração do dispositivo com a Alexa	Baixa	Alto	A Alexa não reconhecer o ESP32-CAM	Trocar o dispositivo de interação por voz para um aplicativo <i>mobile</i>

3 DESENVOLVIMENTO

Este capítulo é responsável por apresentar a implementação do protótipo na seção 3.1, sua verificação na seção 3.2 e seus resultados na seção 3.3.

3.1 IMPLEMENTAÇÃO

A implementação do alimentador controlado via assistente de voz foi realizada utilizando a ferramenta Arduino IDE na linguagem C para controlar o ESP32-Cam. A ferramenta Alexa Developer Console foi utilizada para a programação da assistente de voz em linguagem Python e o banco de dados de tempo real Firebase foi destinado ao armazenamento de comandos.

3.1.1 IMPLEMENTAÇÃO NO ESP32-CAM

Para as interações do microcontrolador, foi realizada uma conexão com a rede wi-fi, dado o módulo wi-fi da ESP32-Cam e posteriormente foram configurados o servidor SMTP e a câmera, conforme demonstrado na figura 17.

```

42
43 // Config do sistema SMTP
44 #define emailSenderAccount "botfotogato@gmail.com"
45 #define emailSenderPassword "kjsiuopfjuqbkgkk"
46 #define smtpServer "smtp.gmail.com"
47 #define smtpServerPort 465
48 #define emailSubject "Foto"
49 #define emailRecipient "vitorkogut@gmail.com"
50
51 // CONFIG CAMERA
52 #define CAMERA_MODEL_AI_THINKER
53 #if defined(CAMERA_MODEL_AI_THINKER)
54 #define PWDN_GPIO_NUM 32
55 #define RESET_GPIO_NUM -1
56 #define XCLK_GPIO_NUM 0
57 #define SIOD_GPIO_NUM 26
58 #define SIOC_GPIO_NUM 27
59 #define Y9_GPIO_NUM 35
60 #define Y8_GPIO_NUM 34
61 #define Y7_GPIO_NUM 39
62 #define Y6_GPIO_NUM 36
63 #define Y5_GPIO_NUM 21
64 #define Y4_GPIO_NUM 19
65 #define Y3_GPIO_NUM 18
66 #define Y2_GPIO_NUM 5
67 #define VSYNC_GPIO_NUM 25
68 #define HREF_GPIO_NUM 23
69 #define PCLK_GPIO_NUM 22
70 #else
71 #error "Camera model not selected"
72 #endif
73

```

Figura 17 - Configuração do ESPCam

Além das inicializações de câmera e wi-fi são inicializados o Firebase e os núcleos do ESP32-Cam. As tarefas são divididas de maneira que o núcleo 0 irá sempre checar se existem comandos a serem executados no banco de dados de tempo real do Firebase, esses comandos são configurados pela assistente de voz. Enquanto isso, o núcleo 1 tem como tarefa o envio de comandos ao motor. Na figura 18 é possível verificar a tarefa de alimentar sendo designada ao núcleo 1.

```

//Inicialização das Tasks
xTaskCreatePinnedToCore(
    TaskDarComida, /* Tas
    "Task1", /* name of
    10000, /* Stack s
    NULL, /* paramet
    1, /* priorit
    &TaskAlimentarHandler
    1); /* pin tas
}

```

Figura 18 - Núcleo 1

Seguindo nas tarefas, a tarefa do núcleo 1 correspondente ao motor, é responsável por alimentar o gato ligando o motor de passos, isso é feito a partir de uma verificação de comando não

executado, com a variável “dar comida” e assim, a função executa o motor de passos em um número de giros correspondente ao número de porções que o usuário solicitou. Na figura 19 é possível verificar essa tarefa sendo executada.

```
// Task alimentar
void TaskDarComida(void * pvParameters){
    while(1==1){
        if( dar_Comida == true){
            dar_Comida = false;
            Serial.printf("\n Dando %d porcoes", quantidade_comida);
            for(int i=0; i<quantidade_comida; i++){
                myStepper.setSpeed(300);
                myStepper.step(4500);
                Serial.print("rodando");
                myStepper.step(-4500);
            }
            delay(20000);
            capturePhotoSaveSpiffs();
            sendPhoto();
        }
        delay(100);
    }
}
```

Figura 19 - Função do núcleo 1

Para que o alimentador saiba as vontades do usuário quanto a alimentação de seu animal de estimação, o alimentador recorre ao banco de dados Firebase para verificar se existem novos comandos. Os comandos são sempre atualizados de maneira a se sobrepor no banco de dados, sendo assim, uma *flag* denominada executado irá sempre estar em 0 quando há um novo comando. O microcontrolador realiza essa checagem ao banco de dados durante o seu loop principal, onde o banco de dados é acessado e caso haja um comando não executado, é adicionada uma *flag* de execução ao banco de dados e lida a quantidade de porções que será dada ao animal, assim, é colocada em afirmativa a opção de alimentar o animal e o núcleo responsável pelo motor irá realizar a função. Na figura 20 tem-se essa demonstração de loop e na figura 21 pode ser visto o banco de dados de tempo real do Firebase.

```

74
75
76 void loop(){
77     if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 10000 || sendDataPrevMillis == 0)) {
78         Serial.write("\n\nChecando comandos pendentes: ");
79         sendDataPrevMillis = millis();
80         if (Firebase.RTDB.getInt(&fbdo, "/comandos/comando/executado")) {
81             int newCommand = fbdo.intData();
82             if (newCommand == 0){
83                 esp_task_wdt_reset();
84                 Serial.write("Comando pendente encontrado!");
85                 if (Firebase.RTDB.getString(&fbdo, "/comandos/comando/quantidade")) {
86                     int newquantidade = fbdo.stringData().toInt();
87                     Firebase.RTDB.setInt(&fbdo, "/comandos/comando/executado", 1);
88                     quantidade_comida = newquantidade;
89                     dar_Comida = true;
90                 }else{
91                     Serial.println(fbdo.errorReason());
92                 }
93             }else{
94                 Serial.write("nenhum comando pendente");
95             }
96         }
97         else {
98             Serial.println(fbdo.errorReason());
99         }
100     }
101 }
102 }

```

Figura 20 - Loop em busca de comando

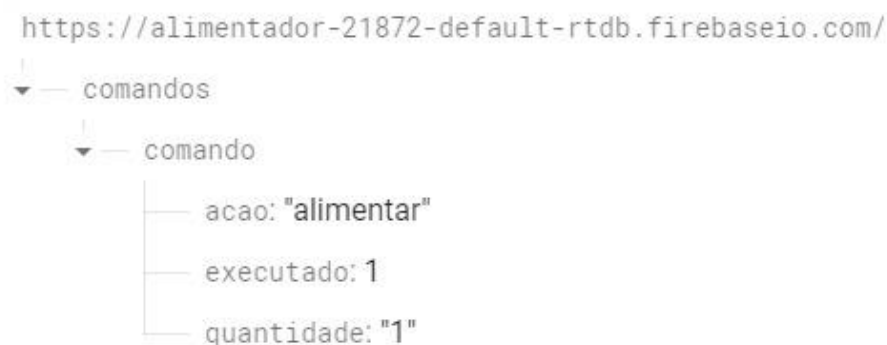


Figura 21 - Banco de dados Firebase

Retornando a tarefa realizada pelo motor, é possível verificar na figura 19 que após a execução da alimentação, há um delay de 20 segundos antes de uma nova função ser ativada. Essa função é chamada de *capturePhotoSaveSpiffs()* e chama a ação da câmera com flash, salvando a imagem para envio. Posteriormente, é chamada a função de envio da foto, essa é enviada via servidor SMTP ao e-mail do usuário que é inicializado juntamente com a conexão a internet no código.

```

void sendPhoto( void ) {
    // Preparing email
    Serial.println("Sending email...");
    // Set the SMTP Server Email host, port, account and password
    smtpData.setLogin(smtpServer, smtpServerPort, emailSenderAccount, emailSenderPassword);
    // Set the sender name and Email
    smtpData.setSender("PetFeed", emailSenderAccount);
    // Set Email priority or importance High, Normal, Low or 1 to 5 (1 is highest)
    smtpData.setPriority("High");
    // Set the subject
    smtpData.setSubject(emailSubject);
    // Set the email message in HTML format
    smtpData.setMessage("<h2>Foto de seu gatito tirada!.</h2>", true);
    // Set the email message in text format
    //smtpData.setMessage("Photo captured with ESP32-CAM and attached in this email.", false);
    //Add recipients, can add more than one recipient
    smtpData.addRecipient(emailRecipient);
    //smtpData.addRecipient(emailRecipient2);
    // Add attach files from SPIFFS
    smtpData.addAttachFile(FILE_PHOTO, "image/jpg");
    // Set the storage type to attach files in your email (SPIFFS)
    smtpData.setFileStorageType(MailClientStorageType::SPIFFS);
    smtpData.setSendCallback(sendCallback);
    // Start sending Email, can be set callback function to track the status
    if (!MailClient.sendMail(smtpData))
    |   Serial.println("Error sending Email, " + MailClient.smtpErrorReason());
    // Clear all data from Email object to free memory
    smtpData.empty();
}

```

Figura 22 - Envio de foto

3.1.2 IMPLEMENTAÇÃO NA ASSISTENTE DE VOZ

Para a implementação da assistente de voz em Python, foi realizada uma conexão com o banco de dados onde os comandos dados a assistente a partir da fala “Alexa, ligar alimentador” são salvos para posterior execução. Após esse comando, a assistente pergunta o que fazer com o alimentador, o usuário deve dizer “Alimentar o gato” e assim ela perguntará o número de porções para armazenar no Firebase, além de enviar o comando de “executado” como 0.

```

30 class AlimentarIntentHandler(AbstractRequestHandler):
31     def can_handle(self, handler_input):
32         return ask_utils.is_intent_name("alimentar")(handler_input)
33
34     def handle(self, handler_input):
35         speak_output = "Alimentando o gato com " + str(handler_input.request_envelope.request.intent.slots["porcoes"].value) + " porções!"
36         ref = db.reference('/')
37         ref.set({
38             'comandos':
39             {
40                 'comando': {
41                     'acao': 'alimentar',
42                     'quantidade': handler_input.request_envelope.request.intent.slots["porcoes"].value,
43                     'executado': 0
44                 }
45             }
46         })
47
48         return (
49             handler_input.response_builder
50                 .speak(speak_output)
51                 .ask(speak_output)
52                 .response
53         )
54

```

Figura 23 - Interação da Alexa com o Firebase

No diagrama a seguir, é possível visualizar melhor o funcionamento da aplicação e sua interação com o banco de dados e assistente de voz.

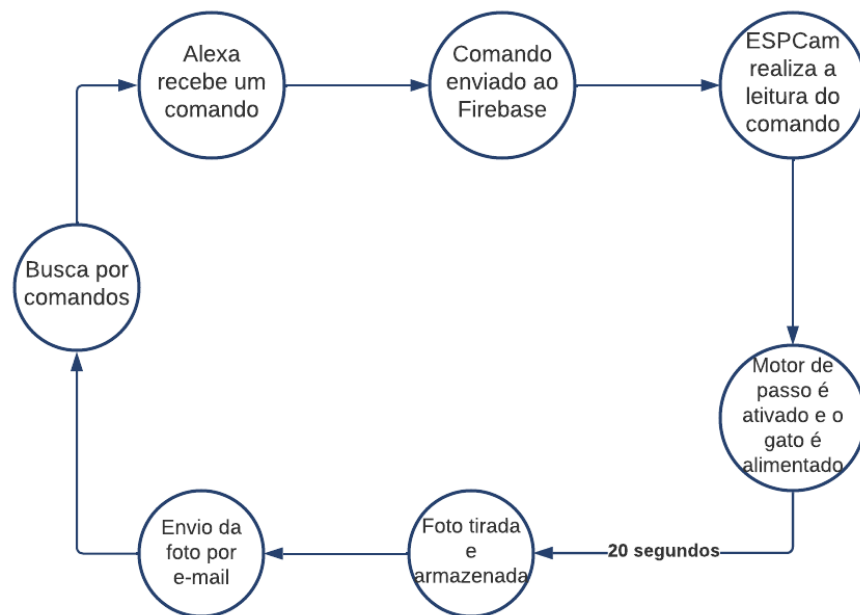


Figura 24 - Funcionamento da aplicação

3.2 VERIFICAÇÃO

Nesta seção são demonstrados os resultados obtidos a partir das verificações e avaliações propostas durante o projeto, no capítulo 2 deste documento.

3.2.1 VERIFICAÇÃO DO DISPOSITIVO ALIMENTADOR

Quadro 20 - Plano de verificação do dispositivo alimentador

Requisito	Procedimento de verificação/teste	Resultado esperado	Resultado obtido
RF01	Cronometrar o tempo de acionamento da câmera	Câmera acionada após 20 segundos	A câmera acionou em 20 segundos
RF02	Verificação da execução das tarefas de software	Tarefas executadas em dois núcleos	As tarefas são realizadas nos dois núcleos
RF03	Verificar se a solicitação de alimentação sem agendamento é funcional	Alimentação imediata permitida	A alimentação sem agendamento é funcional
RF04	Verificar o envio e recebimento das fotos	Fotos recebidas via e-mail	As fotos são recebidas via e-mail
RF05	Verificar o envio e recebimento das fotos	Fotos recebidas via e-mail	As fotos são recebidas via e-mail
RF06	Selecionar um número de porções e conferir seu atendimento	Número de porções igual ao requisitado	O número de porções é dado automaticamente
RF07	Testar a conexão de voz via Alexa	Conexão funcional	A conexão com a assistente de voz é funcional

3.2.2 VERIFICAÇÃO DA ASSISTENTE DE VOZ

Quadro 21 – Plano de verificação da assistente de voz

Requisito	Procedimento de verificação/teste	Resultado esperado	Resultado obtido
RF01	Testar a interação com Alexa via App	O alimentador responde a comandos da Alexa via App	O alimentador recebe e executa comandos via Alexa App
RF02	Testar a interação com Alexa via echo dot	O alimentador responde a comandos da Alexa via echo dot	O alimentador recebe e executa comandos via echo dot
RF03	Testar a função de agendamento da Alexa para o protótipo	A Alexa realiza agendamentos de alimentação	A Alexa aceita agendamento de rotinas, mas não alimenta corretamente
RF04	Solicitar à Alexa um número de porções e aguardar resposta	Número de porções servido ser igual ao solicitado	A Alexa envia o comando correto do número de porções
RF05	Verificar se os dados são enviados no Firebase	Firebase recebendo dados em tempo real	O Firebase recebe os comandos

3.2.3 AVALIAÇÃO

Quadro 22 - Avaliação de métricas

Métrica	Procedimento de avaliação	Resultado esperado	Resultado obtido
Corrente	Multímetro	< 1 ^a	650 mA com o motor em Stall
Potência	Cálculos utilizando a fórmula Corrente x Tensão	< 1W	3,2 W
Energia	Multímetro	< 5V	5V
Tempo de resposta	Teste de software	Menos de 5s	Inferior a 5s
Peso	Uso de balança	900g	Inferior
Altura	Uso de trena	200 mm	Inferior
Largura	Uso de trena	200 mm	Inferior
Profundidade	Uso de trena	200mm	Inferior

Os valores obtidos a partir das verificações e avaliações foram dentro do esperado e o único requisito que não pode ser validado é o requisito da assistente de voz RF03, uma vez que a Alexa permite o agendamento, mas o alimentador não se comporta de acordo com o esperado. Teriam de ser feitas reprogramações no alimentador para que fosse possível o agendamento de refeições.

3 RESULTADOS

A partir da implementação deste projeto foram obtidos os resultados esperados, tendo ao fim desta um protótipo de prova funcional via assistente de voz Echo Dot e a partir do aplicativo *mobile* Alexa. O microcontrolador disposto no núcleo do alimentador permite que sejam buscados comandos no banco de dados de tempo real Firebase e obtém as informações do mesmo em número de porções para fazer com que o motor faça sua rotação e forneça o alimento o número de vezes necessário. Enquanto isso, a assistente de voz permite ao usuário ditar o número de porções para alimentar o gato e salva no banco de dados esse valor. Na figura é possível observar a interação funcional com o dispositivo de assistente de voz.



Figura 25 - Assistente de voz comandando alimentação

Nas figuras 26 e 27 é possível observar a interação do microcontrolador com a conexão com a internet, busca de comandos e execução dos mesmos, além do envio da foto do animal sendo alimentado.

```

SPIFFS mounted successfully
IP Address: http://192.168.0.102
Conectado ao Firebase

Checando comandos pendentes: nenhum comando pendente

Checando comandos pendentes: Comando pendente encontrado!
Dando 1 porcoes

Checando comandos pendentes: nenhum comando pendenterodando

Checando comandos pendentes: nenhum comando pendente

Checando comandos pendentes: nenhum comando pendente

Checando comandos pendentes: nenhum comando pendente

Checando comandos pendentes: nenhum comando pendenteTaking a photo...
Picture file name: /photo.jpg
The picture has been saved in /photo.jpg - Size: 0 bytes
Taking a photo...
Picture file name: /photo.jpg
The picture has been saved in /photo.jpg - Size: 0 bytes
Taking a photo...
Picture file name: /photo.jpg
The picture has been saved in /photo.jpg - Size: 0 bytes
Taking a photo...
Picture file name: /photo.jpg

Checando comandos pendentes: nenhum comando pendenteThe picture has been saved in /photo.jpg - Size: 120832 bytes
Sending email...
Connecting to SMTP server...
SMTP server connected, wait for response...
Identification...
Authentication...
Sign in...
Sending Email header...

```

Figura 26 - Interação do microcontrolador

```

Checando comandos pendentes: nenhum comando pendenteSending Email body...
Sending attachments...
/photo.jpg
Finalize...
Finished
Email sent successfully

Checando comandos pendentes: nenhum comando pendente

Checando comandos pendentes: nenhum comando pendente

Checando comandos pendentes: nenhum comando pendente

```

Figura 27 - Interação do microcontrolador

Por fim, tem-se a estrutura montada utilizando o microcontrolador, o motor de passos e a impressão 3D, gerando um protótipo funcional do alimentador.

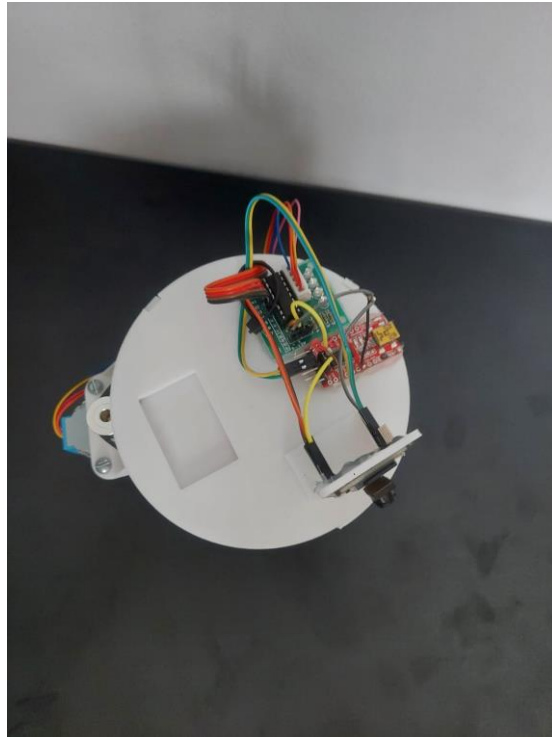


Figura 28 - Dispositivo alimentador finalizado



Figura 29 - Dispositivo alimentador finalizado

4 CONSIDERAÇÕES FINAIS / CONCLUSÕES

Dado o mercado crescente de produtos para animais de estimação e o desenvolvimento da Internet das Coisas, é possível notar uma relevância em um projeto reunindo os dois conceitos, sendo assim o alimentador inteligente por assistente de voz foi projetado. Com a utilização das ferramentas disponibilizadas pela própria desenvolvedora da assistente de voz Alexa e com o uso de um microcontrolador ESP32-Cam, foi desenvolvido um protótipo de prova.

O protótipo de prova apresentado neste documento permite a alimentação de um animal por meio de aplicativo Alexa ou dispositivo físico Echo Dot, utilizando comandos de voz. É possível ao tutor definir também o número de porções que dará ao animal. Dessa forma, após a alimentação o microcontrolador com módulo de câmera também aguarda 20 segundos e dispara a câmera, capturando o animal se alimentando e enviando ao tutor via e-mail.

Para projetos futuros, pretende-se disponibilizar uma atualização da aplicação que agende a alimentação do animal via assistente de voz, utilizando ainda o Alexa Developer Console.

REFERÊNCIAS

ABINPET. **Informações gerais do setor.** Disponível em: <https://abinpet.org.br/infos_gerais/>.

Alimentador Inteligente Sure PetCare SureFeed para Cachorros e Gatos. Disponível em: <<https://www.petlove.com.br/alimentador-inteligente-sure-petcare-surefeed-para-cachorros-e-gatos/p>>. Acesso em: 27 out. 2022.

ANGGRAINI, N. et al. **Mobile based monitoring system for an automatic cat feeder using Raspberry Pi.** TELKOMNIKA (Telecommunication Computing Electronics and Control), v. 18, n. 2, p. 1038, 1 abr. 2020.

Cat Mate C3000 alimentador automático para alimentos secos com 3 refeições, livre de BPA para gatos e cães pequenos | Amazon.com.br. Disponível em: <<https://www.amazon.com.br/Alimentador-autom%C3%A1tico-estima%C3%A7%C3%A3o-Cat-Mate/dp/B004BLVFSK>>. Acesso em: 27 out. 2022.

ESP32-Cam Development board. Disponível em: <https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0602_Web.pdf>. Acesso em: 3 nov. 2022.

GRAHL, C. A. **Simulando dispositivos WeMo com Alexa.** Disponível em: <<https://medium.com/@carloaugustograhl/simulando-dispositivos-wemo-com-alexa-9221a3c9e3f5>>. Acesso em: 12 nov. 2022.

IoT: um mercado de R\$ 8,5 bilhões no Brasil. Disponível em: <<https://www.convergenciadigital.com.br/Internet-Movel/IoT%3A-um-mercado-de-R%24-8%2C5-bilhoes-no-Brasil-59415.html?UserActiveTemplate=mobile>>. Acesso em: 27 out. 2022.

SHOPEE. Disponível em: <<https://shopee.com.br/Alimentador-Inteligente-com-Comando-de-Voz-Autom%C3%A1tico-i.587134166.17878484308>>. Acesso em: 27 out. 2022.

WIKI, G. **Stepper Motor 5V 4-Phase 5-Wire & ULN2003 Driver Board for Arduino.** [s.l: s.n.]. Disponível em: <<http://eeshop.unl.edu/pdf/Stepper+Driver.pdf>>.

WILSON, J. **LM2596 Buck Converter Datasheet, Pinout, Features, Applications**. Disponível em: <<https://www.theengineeringprojects.com/2020/09/lm2596-buck-converter-datasheet-pinout-features-applications.html>>.

Zenpet | **Peace of mind for petlovers** - Zenpet. Disponível em: <<https://zen.pet/>>. Acesso em: 27 out. 2022.

APÊNDICE

O código para a implementação se encontra no Github e pode ser acessado em
<https://github.com/magagninicole/AlimentadorInteligente_M3>