

**PET FEEDER**

**O seu amigo bem alimentado!**

Mês 06/2023

Município Florianópolis/SC

**17/12/2015**

Florianópolis/SC

**SUMÁRIO**

[1. INTRODUÇÃO 3](#_Toc137495868)

[2. OBJETIVO 3](#_Toc137495869)

[3. DESENVOLVIMENTO 3](#_Toc137495870)

[3.1 Requisitos funcionais 4](#_Toc137495871)

[3.2 Requisitos não funcionais 4](#_Toc137495872)

[3.3 Componentes eletrônicos 4](#_Toc137495873)

[3.4 Sensores 5](#_Toc137495874)

[3.5 Atuadores 6](#_Toc137495875)

[OUTROS MATERIAS 7](#_Toc137495876)

[5.RESULTADO 8](#_Toc137495877)

[6. DIAGRAMA DE CASOS DE USO 9](#_Toc137495878)

[CODIGO DA APLICAÇÃO 10](#_Toc137495879)

[7.REFERÊNCIAS 10](#_Toc137495880)

[8.HISTÓRICOS DE ALTERAÇÕES 10](#_Toc137495881)

1. INTRODUÇÃO

A pratica de domesticação de animais já ocorre desde do início do período neolítico, quando o ser humano deixou de ser nômade e passou a se instalar raízes em um determinado lugar, desde então essa prática se tornou comum até os dias de hoje ,onde antes esses animais tinham como objetivos serem domesticado para o uso da sua força para trabalhos pesados e hoje eles já são vistos como membros de famílias e nossos amigos .Dentre os preferidos das famílias estão os cães e gatos, onde somente no Brasil o número desses animais representam um total de 80 milhões segundo dados do Gloogle.

Pensando nisso, e em um contexto tecnológico em que estamos vivendo nos dias de hoje, o porquê não encontrar uma solução tecnológica para alguns desses pais de pets que encontram dificuldades quanto a alimentação controlada de seus animais.

1. OBJETIVO

Pensando nisso a nossa aplicação vem com o objetivo de ajudar nessa questão de alimentação saudável e controlada, evitando alguns incomodo indesejado de ter que pedir um favor para seu vizinho ou até mesmo você ter que levar o seu bichinho para uma outra casa, causando estrese no animal e também controlando a obesidade do animal. Sendo assim, o usuário dessa aplicação terá a liberdade de sair de casa tranquilamente sem ter que se preocupar com o bem-estar de seu bichinho.

1. DESENVOLVIMENTO

O Pet Feeder foi desenvolvido na IDE ARDUINO em linguagem C com apoio de algumas bibliotecas nativas da ID, o código roda em cima da placa de desenvolvimento Wifi Bluetooth ESP32, semelhante a NodeMcu e ESP8266.

Para que todo o código fosse funcional, como dito anteriormente, foi preciso instalar algumas bibliotecas para auxiliar na calibragem da balança (SENSOR DE PESO).

Após finalizar o código do sensor de peso passamos para a montagem do dispositivo Micro Servo, que irá atuar com a função de liberar a comporta do reservatório de alimento, fazendo com que seja despejado quando atingir o peso predefinido pelo usuário, conforme a necessidade de seu pet.

Montado toda a protoboard com o código funcionando, passamos para a TagoIO.

Oque é TagoIO?

TagoIO e uma plataforma web 100% cloud e de alto nível para monitoramento de ambientes via dispositivos IoT conectados a sua rede.

Já na TagoIO foi criado um DEVICE para fazer a integração com a IDE Arduino.

Criado o DEVICE é gerado um token para concluir essa interação entre Tago e IDE. Feito isso, para que haja interação entre as duas plataformas usamos como ponte o protocolo MQTT que será responsável por receber e enviar requisições em formado JSON através de tópicos que serão coletados da placa Esp32 E armazenados nos buckts da DEVICE.

3.1 Requisitos funcionais

3.1 O dono do pet, terá informações diárias de peso/kg, que o pet consumiu por dia

3.2 O dono do pet, terá relatórios de média diária de consumo.

3.3 O dono do pet, terá autonomia para realizar a alimentação do pet

3.2 Requisitos não funcionais

* 1. O dono do pet sai de casa sem previsão de voltar para a casa
  2. O pet está com fome e tem acesso ao pet feeder
  3. O pet feeder capitando a informação que foi acionado o seu uso (o pet se alimentou)
  4. O pet feeder fica programado para fazer um próximo abastecimento do reservatório no tempo estipulado
  5. Caso o pet não se alimentou o sensor não ativa o próximo abastecimento e sim um abastecimento de reserva,
  6. Evitando que o pet fique muito tempo sem se alimentar
  7. O dono do pet tem controle remotamente dos relatórios armazenados pelo pet feeder, assim ficando tranquilo sabendo que seu pet está bem alimentado.

3.3 Componentes Eletrônicos

* O projeto foi desenvolvido em uma placa modelo Esp32, integrado a um sensor de peso utilizando uma célula de carga e um Módulo Conversor Amplificador HX711, e um Motor Micro Servo SG90.

E para fazer o abastecimento após ser identificada essa necessidade, teremos um motor Micro Servo SG90, atuando como atuador para fazer o despejo do alimento no recipiente.

3.4 Sensores

**Sensor de Peso**

É composto por uma ponte resistiva que altera sua resistência conforme o peso aplicado. Eletricamente quando o Sensor de peso entra em operação, ou seja, é aplicado determinado peso, ele envia uma tensão ao micro controlador, conforme o peso a tensão de saída vai variar.

**Especificações:**

- Carga nominal: 5 kg;  
- Potência nominal de saída: 1.0mv/v ± 0.15mv/v;  
- Tensão de funcionamento recomendada: 3 ~ 12 VDC;  
- Tensão máxima de operação: 15 VDC;  
- Impedância de entrada: 410 +/-30 ohm;  
- Impedância de saída: 350 +/-3 ohm;  
- Isolamento: > 2000 megaohm/50 VDC;  
- Faixa de compensação de temperatura: -10C ~ 40C;  
- Faixa de temperatura de operação: -20C ~ 60C;  
- Sobrecarga permitida: <120% da carga nominal;  
- Carga de ruptura: >150% da carga nominal;  
- Classe de proteção: IP65;  
- Material: liga de alumínio;  
- Dimensões(CxLxA):80mm,12,7mm,12,7mm;  
- Peso: 31g;  
- Ligação do fios: Vermelho(+),Preto(-),Verde(Sinal +),Branco(Sinal -);

****

3.5 Atuadores

**O Micro Servo SG90**

 É um motor compacto, pesa apenas 9g e oferece um torque máximo de ~1.6kg, mostrando-se ideal para as mais diversas aplicações em projetos robóticos e eletrônicos em geral. Acompanha ainda dois parafusos para fixação do servo, três cruzetas e um parafuso para fixação das cruzetas.

**Especificações:**  
- Material da engrenagem: Nylon;  
- Tensão de operação: 3 - 7.2 V;   
- Velocidade em 4.8 V e sem carga: 0.12 seg/60Graus;  
- Torque em 4.8 V:1.2 kg.cm;  
- Torque em 6 V: 1,6 kg.cm;  
- Temperatura de operação.: -30ºC a +60ºC;  
- Dimensões: 32 mm x 30 mm x 12 mm;  
- Tamanho cabo: 245 mm  
- Peso: 9 g.



OUTROS MATERIAS

**Protoboard, Placa Esp32 e Jumpers**

Protoboard

É uma placa de ensaio que serve como um protótipo de um aparelho eletrônico



**Placa Esp32**

O módulo ESP32 **é um módulo Wi-Fi de alta performance, com um baixíssimo consumo de energia**



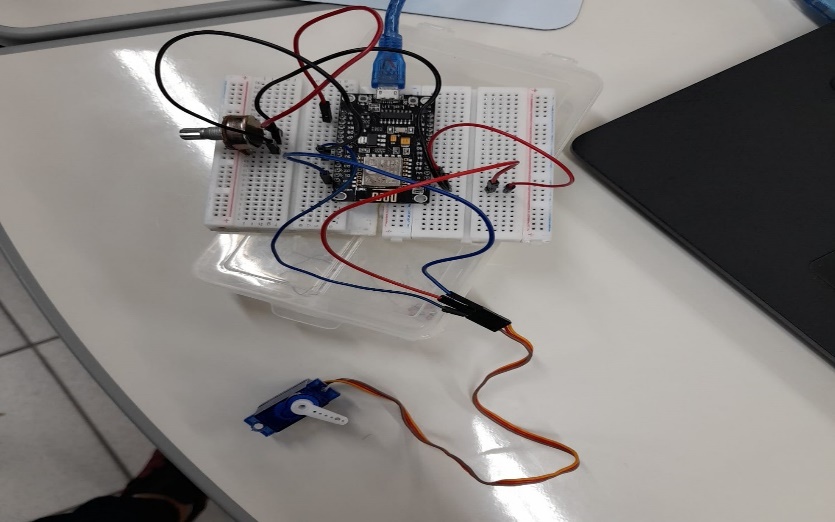
**Jumpers**

 Um jumper é uma peça plástica que contém um pequeno filamento de metal responsável pela condução de eletricidade.



5.RESULTADO

Após montado o protótipo utilizando todos os componentes anteriores, chegamos no resultado



6. DIAGRAMA DE CASOS DE USO

O sistema identifica que esta na hora de abastecer

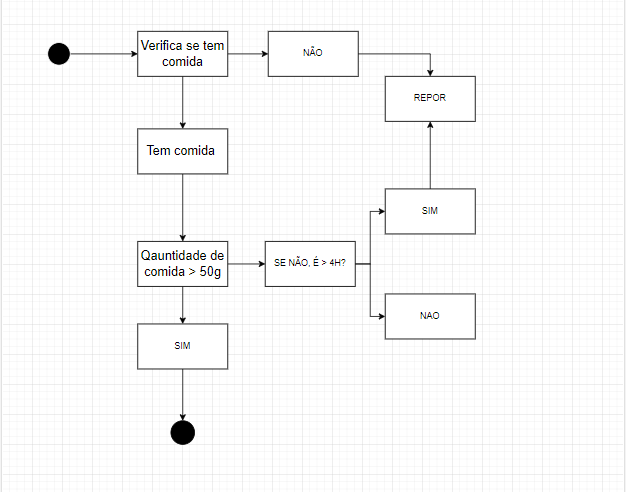
Verifica se há ração no recipiente

SE SIM

Havendo comida,o sistema completa o pote e finaliza e aguarda o próximo ciclo programado.

SE NÃO

Não havendo comida o sistema abastece a quantidade programado pelo usuário e aguarda o próximo ciclo programado.



Na aplicação foi utilizado um cabo USB para fazer a transferência de dados entre o pc e a placa Esp32 e, para o desenvolvimento do código. Foi utilizado a IDE Arduino em linguagem de programação C++, foi incluído algumas bibliotecas para ajudar na aplicação.

7.REFERÊNCIAS

**Nossa referência do projeto pode ser encontrada nos seguintes links;**

**Autor:** Innovative Ideas

**Ideia baseada em:**

[**https://www.youtube.com/watch?v=imtcdl9jHVQ**](https://www.youtube.com/watch?v=imtcdl9jHVQ)

**Referência de Sensor Célula de peso:**

[**https://www.casadarobotica.com/sensores-e-modulos/sensores/outros/celula-de-carga-5kg-sensor-de-peso-arduino#:~:text=Mecanicamente%20o%20Sensor%20de%20Peso,tens%C3%A3o%20de%20sa%C3%ADda%20vai%20variar**](https://www.casadarobotica.com/sensores-e-modulos/sensores/outros/celula-de-carga-5kg-sensor-de-peso-arduino#:~:text=Mecanicamente%20o%20Sensor%20de%20Peso,tens%C3%A3o%20de%20sa%C3%ADda%20vai%20variar)

**Referência de atuador Micro Servo:**

[**https://www.eletrogate.com/micro-servo-9g-sg90-towerpro**](https://www.eletrogate.com/micro-servo-9g-sg90-towerpro)

8.HISTÓRICOS DE ALTERAÇÕES

|  |  |  |  |
| --- | --- | --- | --- |
| DATA | DESCRIÇÃO | AUTOR |  |
| 03/05 | Descrição de requisitos | Daniel |  |
| 04/05 | Criação dos requisitos | Daniel |  |
| 19/05 | Criação do nome do projeto | Daniel |  |
| 20/05 | Iniciou o desenvolvimentos dos códigos | Rodrigo/João |  |
| 21/05 | Foi realizado mudanças logo do projeto | Daniel |  |
| 25/05 | Iniciou se a integração dos códigos do motor com o código da calibragem da balança | Rodrigo/João |  |
| 02/06 | Foi feito revisão nas telas de apresentação | Daniel |  |
| 05/06 | Todos os códigos foram unidos e funcionou | Rodrigo/João |  |
| 07/06 | Foi feito nova revisão nas telas e integração dos códigos no Brooker da tagoIO | João/Rodrigo/Daniel |  |