

Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América



Facultad de Ingeniería de Sistemas e Informática

**Dispensador de alimento para Mascotas basado en IoT controlado por el Asistente
de Google mediante NodeMCU**

Internet de las Cosas

Integrantes

Albert Williams Perez Santiago

Miguel Stalin Soria Villanueva

Erly Toribio Rivera Inche

Nicolas Alonso Rojas Gala

LIMA, PERÚ

2024

1. Introducción

En la era de la digitalización y la conectividad, el Internet de las Cosas (IoT) ha revolucionado la manera en que interactuamos con el mundo que nos rodea. Este avance tecnológico ha permitido la creación de dispositivos inteligentes capaces de comunicarse entre sí y con nosotros, facilitando la automatización de diversas tareas cotidianas. Una de las aplicaciones más prácticas y beneficiosas de IoT es la automatización de los cuidados para nuestras mascotas, permitiendo un manejo más eficiente y cómodo de sus necesidades.

2. Problemática

La falta de soluciones automatizadas accesibles y eficaces para el cuidado de mascotas en el mercado peruano contribuye a esta problemática. La mayoría de las opciones disponibles son costosas o requieren de un conocimiento técnico avanzado para su implementación, lo que las hace poco prácticas para el usuario promedio. Además, la falta de integración con tecnologías comunes, como asistentes de voz, limita la facilidad de uso y la adopción de estas soluciones. En este contexto nos planteamos la siguiente pregunta: ¿La construcción de un dispensador de alimento para mascotas basado en IoT, controlado mediante el asistente de google mediante Nodemcu, logrará alimentar las mascotas remotamente?

3. Objetivos

O. General:

Desarrollar un dispensador de alimento para Mascotas basado en IoT controlado por el Asistente de Google mediante NodeMCU.

O. Específicos:

- Diseñar el prototipo del dispensador de alimento para mascotas
- Implementar el sistema IoT utilizando NodeMCU
- Desarrollar la funcionalidad de control de voz mediante el Asistente de Google.

4. Antecedentes

Actualmente, la creciente tendencia a la automatización y la conectividad en dispositivos electrónicos ha generado avances significativos en la domótica aplicada a la alimentación de mascotas.

Entre los dispositivos desarrollados, se destaca el dosificador de alimentos sólido y líquido para gatos desarrollado por Velasco (2020) como parte de su trabajo de titulación en ingeniería electrónica en la Universidad Católica de Santiago de Guayaquil. Este dispositivo integra un Arduino MEGA, módulos como RTC y WIFI esp8266 para programación y conectividad, sensores FSR para detección, una micro bomba sumergible 5V para dispensación de líquidos, y un servo motor ES08MAX para el control del sistema. Con estos componentes, se facilita la supervisión remota y la gestión eficiente de la alimentación de mascotas, proporcionando datos cruciales para el cuidado veterinario.

García (2020) desarrolló un prototipo dispensador de alimento para mascotas, denominado PET-HOME ALBA, basado en Internet de las Cosas (IoT). Este proyecto tuvo como objetivo explorar las aplicaciones de IoT en el ámbito de los dispositivos domésticos. El prototipo diseñado utiliza un nodo central como servidor, un nodo lector y una aplicación móvil para permitir el control remoto y automatizado del dispensador de alimento. Este sistema innovador emplea tecnologías como el protocolo MQTT y un módulo NodeMcu ESP8266, ofreciendo así una solución eficiente y adaptable para la gestión de la alimentación de mascotas, específicamente diseñado para perros y gatos.

Reynoso (2023) realiza una aplicación móvil para monitoreo y control de un alimentador automático de mascotas, implementando una estructura física que integra componentes eléctricos conectados mediante el protocolo MQTT. El objetivo de la aplicación es facilitar la interacción con el alimentador desde cualquier parte del mundo mediante una conexión a internet. Para ello, se empleó un módulo WiFi NodeMCU ESP8266, una tarjeta de desarrollo similar a Arduino, diseñada específicamente para el Internet de las Cosas (IoT). El desarrollo de la aplicación se realizó en lenguaje Lua utilizando el framework de Arduino, aprovechando las librerías disponibles para Arduino y su entorno de desarrollo integrado (IDE). Las pruebas realizadas revelaron tiempos máximos de respuesta de 5 segundos para el servomotor y 1 segundo para los sensores, destacando así un avance significativo en la automatización y conectividad de dispositivos para mejorar el cuidado y bienestar de las mascotas domésticas.

5. Funcionalidades

- Control por Voz: Habilidad del control del comedero mediante comandos de voz a través del Asistente de Google, permitiendo la alimentación de la mascota desde cualquier lugar.
- Programación de Alimentación: Configuración de horarios de alimentación automáticos para dispensar comida a intervalos regulares.

6. Componentes del Sistema

NodeMCU ESP8266: Dispositivo conocido por su capacidad de conexión Wi-Fi, tanto en modo estación y también como punto de acceso, este módulo es utilizado para proyectos de bajo costo.

Servo Motor: Es el actuador encargado de controlar la apertura y cierre del dispensador de alimento. Este motor es controlado por el NodeMCU para liberar la cantidad de comida requerida.

Pantalla LCD (I2C): La pantalla LCD sirve para mostrar información relevante al usuario, como el estado del dispensador, cantidad de alimento dispensado, o mensajes de error. Esta pantalla está conectada al NodeMCU a través de una interfaz I2C.

Adafruit IO: Es una plataforma de servicios en la nube que permite gestionar y monitorizar dispositivos IoT. El NodeMCU se comunica con Adafruit IO para recibir comandos y enviar datos.

Google Assistant: El usuario interactúa con el sistema a través del Asistente de Google en su dispositivo móvil. El Asistente de Google interpreta los comandos de voz del usuario y los envía a Adafruit IO.

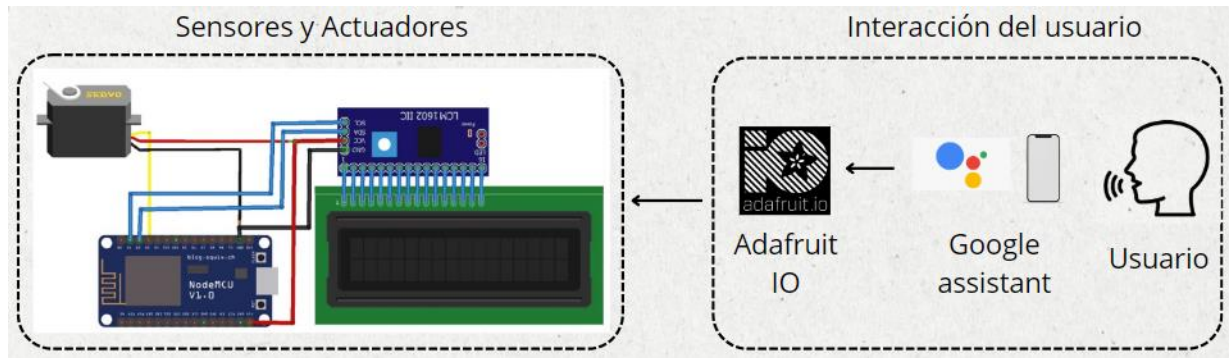
Servomotor:

Un servomotor es un dispositivo electromecánico que proporciona control preciso de posición angular, velocidad y aceleración

7. Arquitectura

En el siguiente diagrama se puede observar la composición del circuito del dispensador de alimentos para mascotas, en donde se evidencia el uso del NodeMCU8266, el servomotor y los módulos LCD. Además, de la implementación de

Adafruit IO y Google assistant para que mediante comandos de voz se logre accionar el servomotor.



8. Código Fuente

A continuación, se muestra el código fuente que permite la interacción con los distintos componentes del sistema.

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#include <Servo.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", 19800, 60000);

Servo servo;
LiquidCrystal_I2C lcd(0x27, 16, 2); // Dirección I2C y dimensiones del LCD

#define WIFI_SSID "Tilin"
#define WIFI_PASS "esotilin666"

#define MQTT_SERV "io.adafruit.com"
#define MQTT_PORT 1883
#define MQTT_NAME "EsoTilin666"
#define MQTT_PASS "aio_QQXQ090gCTrzvno4J2toh5fXVFbw"

int SERVO_PIN = D3; // Pin donde está conectado el servo
int CLOSE_ANGLE = 0; // Ángulo de cerrado del servo
int OPEN_ANGLE = 180; // Ángulo de apertura del servo
int hh, mm, ss;
int feed_hour = 0;
int feed_minute = 0;

// Configuración de clientes MQTT y WiFi
WiFiClient client;
```

```

Adafruit_MQTT_Client mqtt(&client, MQTT_SERV, MQTT_PORT, MQTT_NAME,
MQTT_PASS);

// Suscripción al feed onoff
Adafruit_MQTT_Subscribe onoff = Adafruit_MQTT_Subscribe(&mqtt, MQTT_NAME
"/f/onoff");
boolean feed = true; // Condición para la alarma

void setup() {
  Serial.begin(9600);
  timeClient.begin();
  Wire.begin(D2, D1); // Inicializar I2C

  // Inicializar LCD
  lcd.begin(16, 2);
  lcd.init();
  lcd.backlight();
  lcd.clear();

  // Conectar a WiFi
  Serial.print("\n\nConnecting Wifi... ");
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("OK!");

  // Suscribirse al feed onoff
  mqtt.subscribe(&onoff);
  servo.attach(SERVO_PIN);
  servo.write(CLOSE_ANGLE);
}

void loop() {
  MQTT_connect();
  timeClient.update();
  hh = timeClient.getHours();
  mm = timeClient.getMinutes();
  ss = timeClient.getSeconds();

  // Mostrar la hora actual
  lcd.setCursor(0, 0);
  lcd.print("Time: ");
  if (hh > 12) {
    hh = hh - 12;
    lcd.print(hh);
    lcd.print(":");
    lcd.print(mm);
    lcd.print(":");
    lcd.print(ss);
    lcd.print(" PM ");
  }
}

```

```

}
Else {
    lcd.print(hh);
    lcd.print(":");
    lcd.print(mm);
    lcd.print(":");
    lcd.print(ss);
    lcd.print(" AM ");
}

// Mostrar la hora de alimentación
lcd.setCursor(0, 1);
lcd.print("Feed Time: ");
lcd.print(feed_hour);
lcd.print(':');
lcd.print(feed_minute);

Adafruit_MQTT_Subscribe *subscription;
while ((subscription = mqtt.readSubscription(5000))) {
    if (subscription == &onoff) {
        Serial.println((char*) onoff.lastread);

        if (!strcmp((char*) onoff.lastread, "ON")) {
            open_door();
            delay(1000);
            close_door();
        }
        if (!strcmp((char*) onoff.lastread, "Morning")) {
            feed_hour = 10;
            feed_minute = 30;
        }
        if (!strcmp((char*) onoff.lastread, "Afternoon")) {
            feed_hour = 1;
            feed_minute = 30;
        }
        if (!strcmp((char*) onoff.lastread, "Evening")) {
            feed_hour = 6;
            feed_minute = 30;
        }
    }
}

if (hh == feed_hour && mm == feed_minute && feed == true) {
    open_door();
    delay(1000);
    close_door();
    feed = false;
}
}

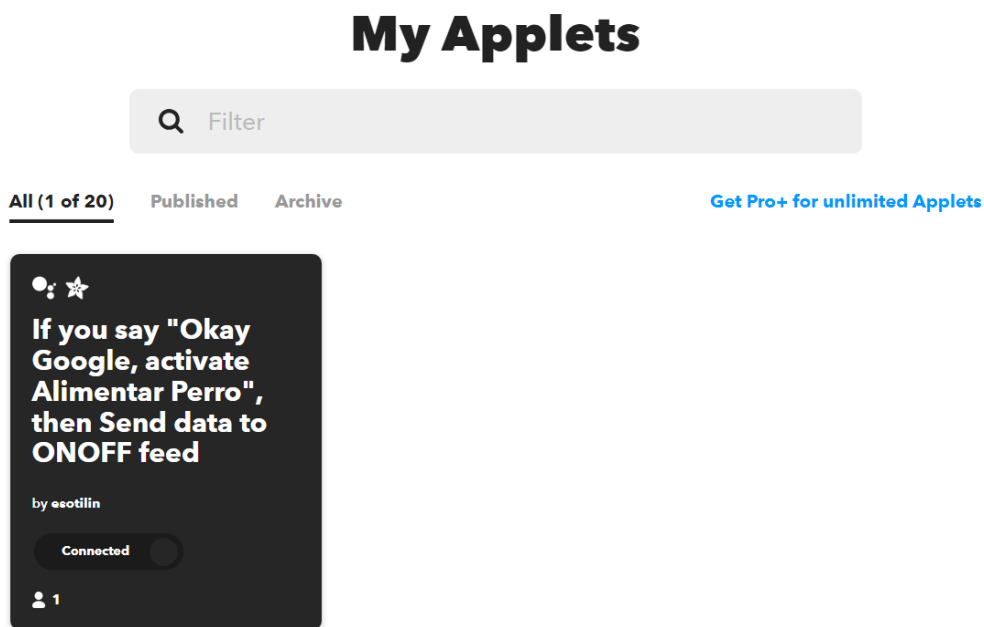
void MQTT_connect() {
    int8_t ret;

```

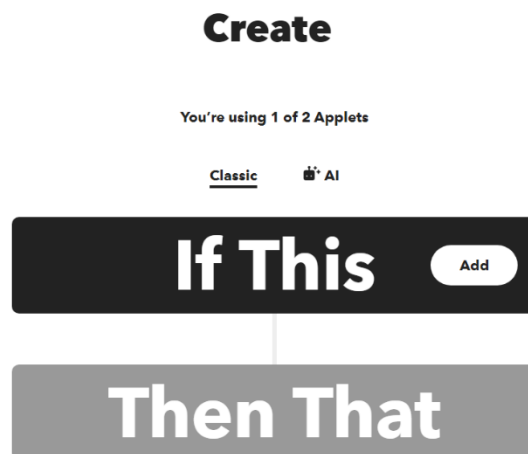
```
if (mqtt.connected()) {  
    return;  
}  
  
uint8_t retries = 3;  
while ((ret = mqtt.connect()) != 0) {  
    mqtt.disconnect();  
    delay(5000);  
    retries--;  
    if (retries == 0) {  
        while (1);  
    }  
}  
}  
  
void open_door() {  
    servo.write(OPEN_ANGLE);  
}  
  
void close_door() {  
    servo.write(CLOSE_ANGLE);  
}
```


9. Configuración de IFTT

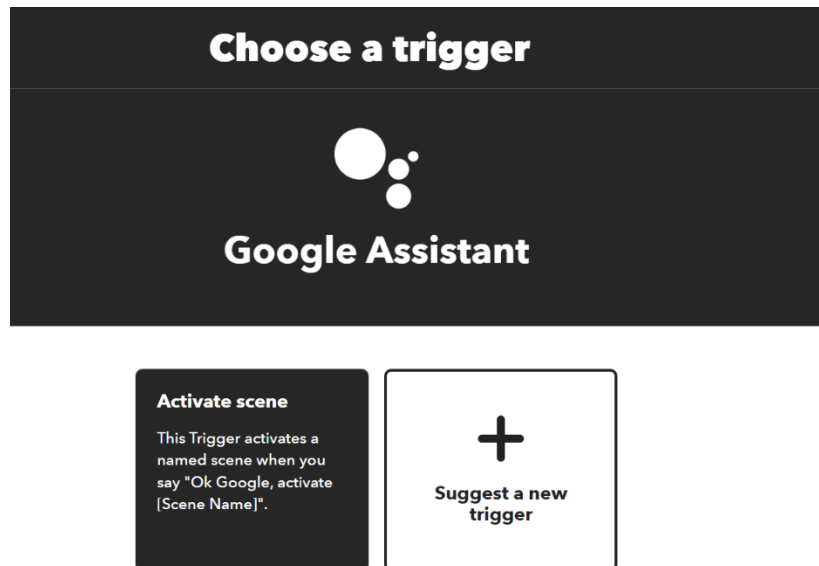
Usamos Reconocimiento de voz a través del asistente de google mediante el uso de “IFTTT” para detectar al momento que se dice “Okay Google, alimentar perro” se mande una señal a ADAFRUIT para que puede ser reconocido por la placa y poder liberar el alimento para la mascota



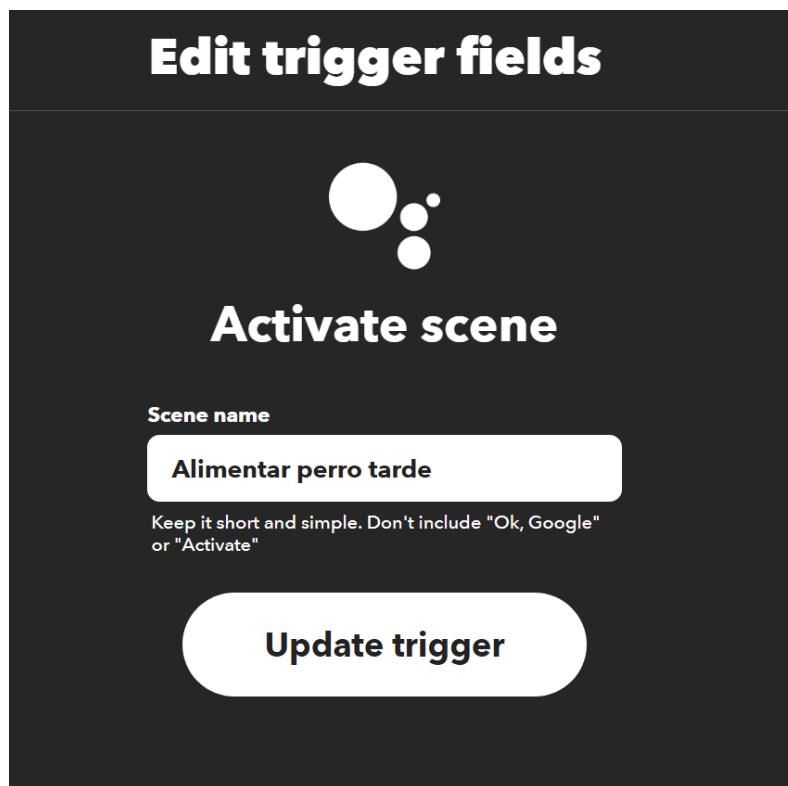
Primero elegimos crear un nuevo applet:



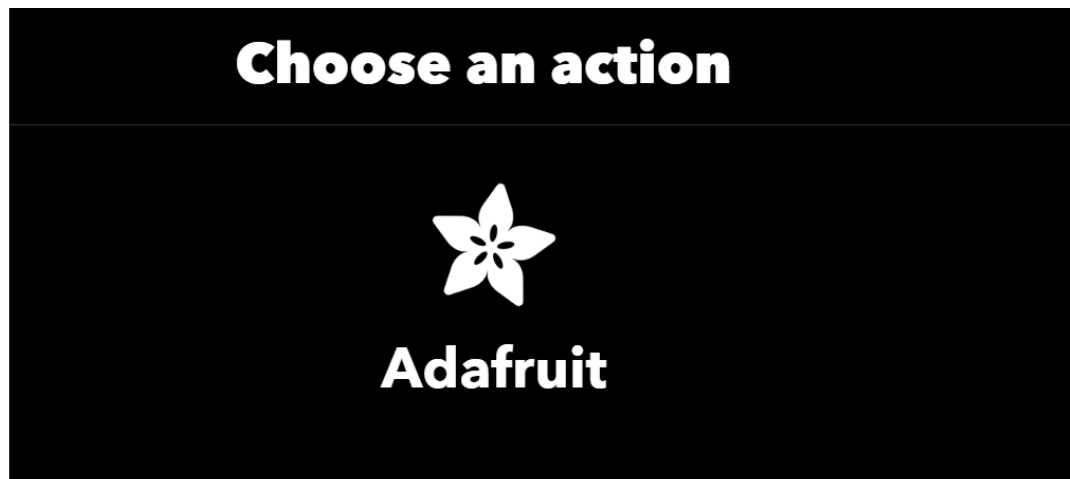
Seleccionamos la opción de asistente de google y creamos una nueva escena



Mencionamos la palabra de activación para que será recibida por el asistente de Google



Elegimos el servicio de ADAFRUIT que nos ayuda a la conexión entre el asistente de Google y el esp8266 e introducimos la data que se envía al momento de usar la palabra clave con el asistente de Google



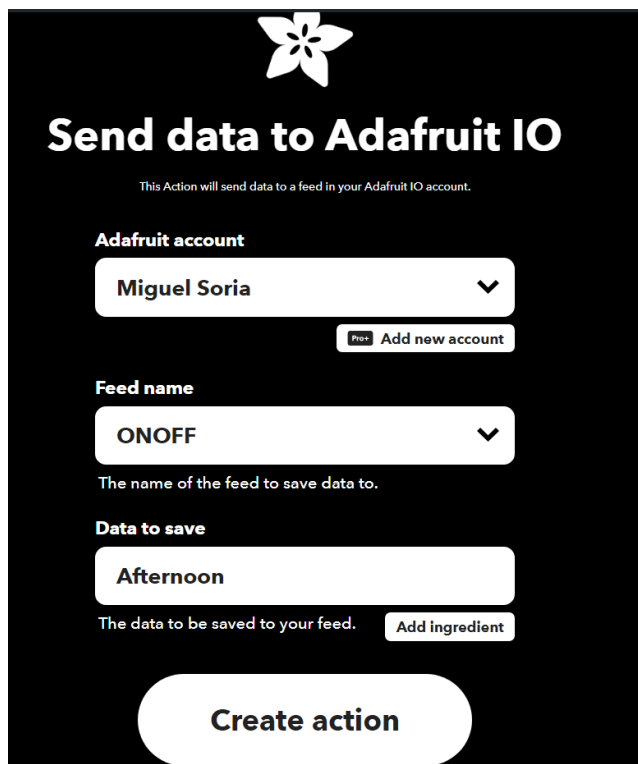
**Send data to
Adafruit IO**


This Action will send data
to a feed in your Adafruit
IO account.



**Suggest a new
action**

Seleccionamos la palabra clave que será enviada a ADAFRUIT





Send data to Adafruit IO

This Action will send data to a feed in your Adafruit IO account.

Adafruit account

Miguel Soria ▼

Free Add new account

Feed name

ONOFF ▼

The name of the feed to save data to.

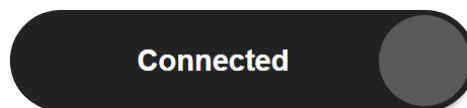
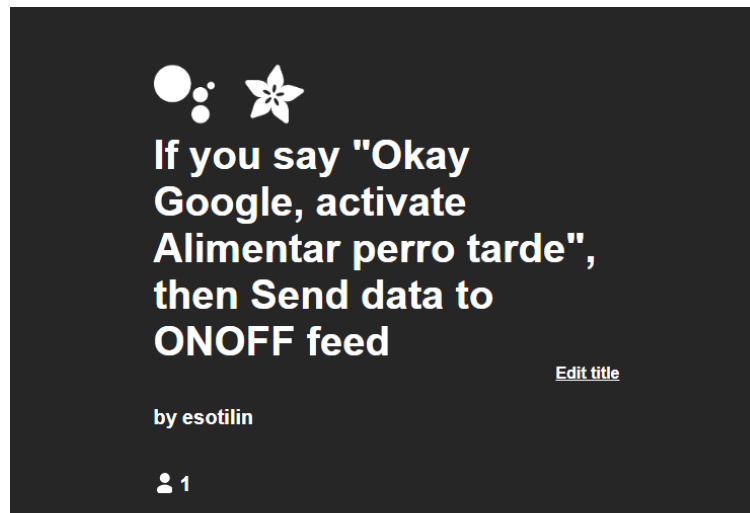
Data to save

Afternoon

The data to be saved to your feed. Add ingredient

Create action

Después de todo eso tendríamos el applet creado y previamente tendremos que conectar nuestra cuenta de Google a IFTTT para que se puede vincular con Google home y poder hacer uso del alimentador de mascotas



More details

Connected Jun 27, 2024

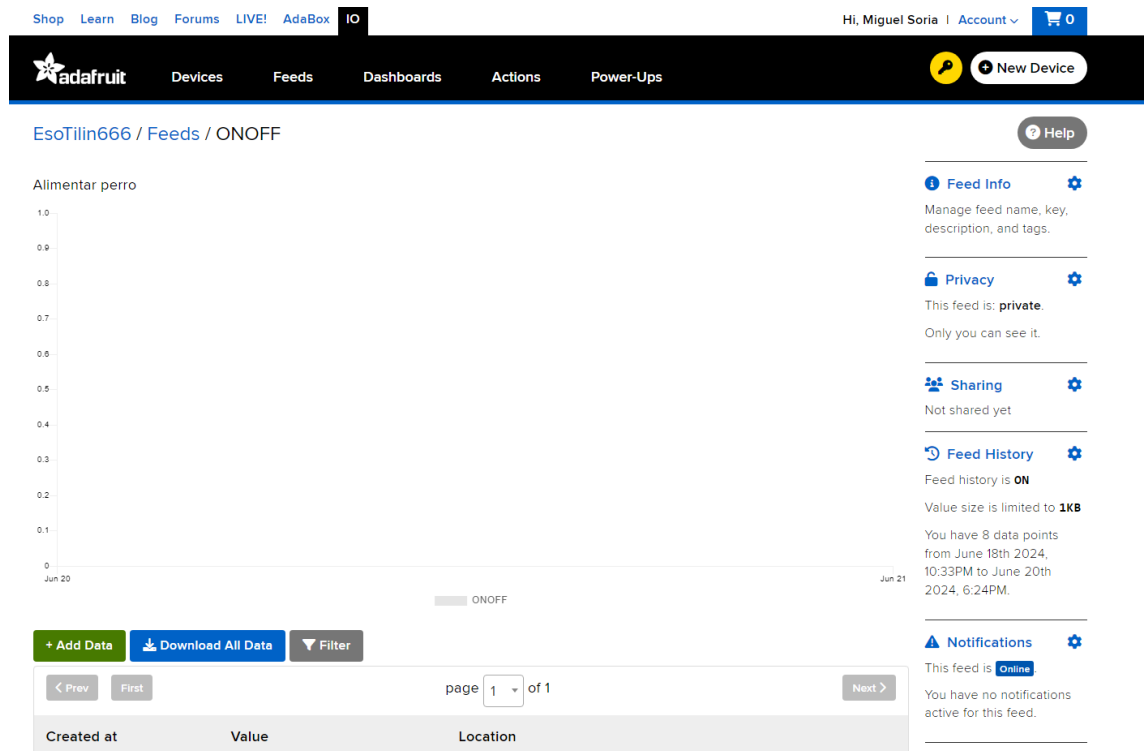
Never run

Notifv me when this runs



10. Configuración de ADAFRUIT

Usamos **ADAFRUIT** para una vez el asistente de google recibe el orden este le manda la una señal en este caso “ON” que va a ser almacena como un dato que sera almacenado y este asu vez envia el dato de “ON” para que activarlo y se puede alimentar a la mascota asi mismo nos brinda un dasboard donde se muestran las veces que fueron activadas



En ADRAFRUIT tenemos que crear un feed que recibe los datos para que puedan ser mostrados en el dasboard

adafruit Devices Feeds Dashboards Actions Power-Ups New Device

EsoTilin666 / Feeds Help

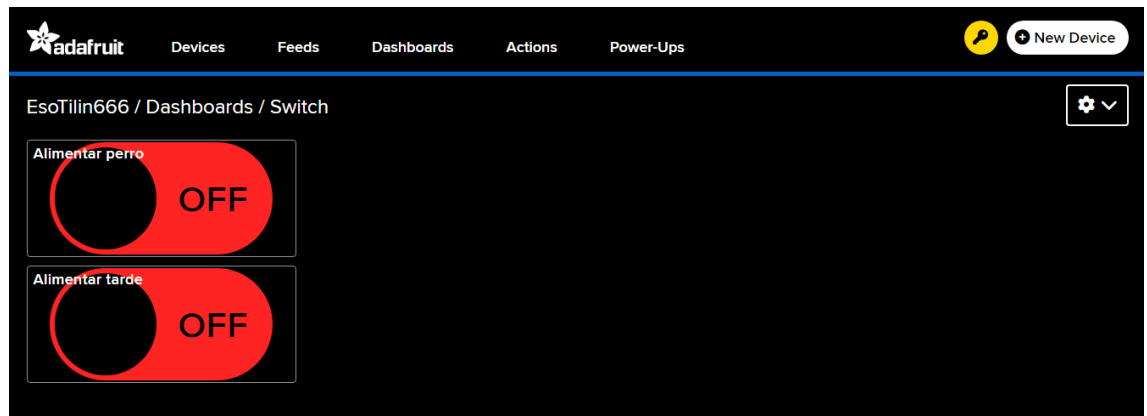
New Feed New Group

Feed Name	Key	Last value	Recorded
<input type="checkbox"/> ONOFF	onoff	OFF	1 minute ago
<input type="checkbox"/> Proyecto Iot Grupo 2	proyecto-iot-grupo-2		9 days ago
<input type="checkbox"/> Welcome Feed	welcome-feed	0	4 days ago

ONOFF

Feed Name	Key	Last value	Recorded
-----------	-----	------------	----------

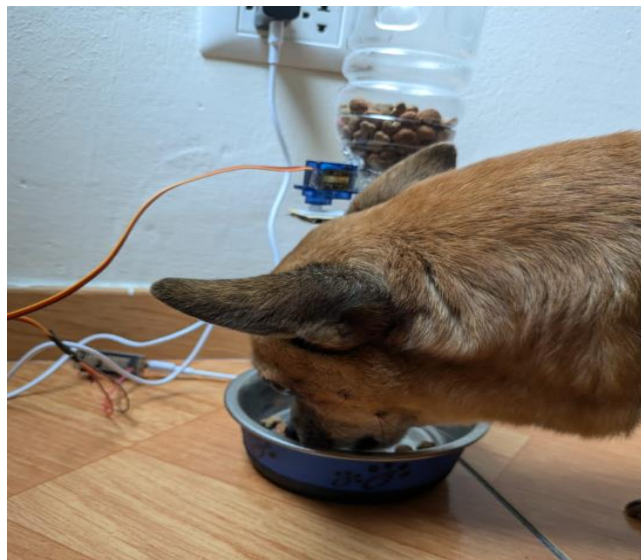
Así mismo crear los switches con las activación del alimentador de mascotas



11. Resultado

En las siguientes imágenes se muestra el resultado de este proyecto de IoT.





12. Conclusiones

- La selección adecuada de materiales y componentes ha permitido la creación de un dispositivo robusto y fiable, que puede soportar el uso diario y proporcionar una solución práctica para la alimentación automatizada de mascotas.
- La programación del NodeMCU y la integración de los sensores y actuadores necesarios han permitido que el dispositivo funcione de manera autónoma y precisa, demostrando la viabilidad de utilizar tecnologías IoT en aplicaciones domésticas para el cuidado de mascotas.

- La integración exitosa entre el NodeMCU y el Asistente de Google ha demostrado la capacidad de crear soluciones tecnológicas avanzadas que mejoran la interacción y la experiencia del usuario.

Referencias:

- García, A. (2020). *Desarrollo de prototipo dispensador de alimento para mascotas PET-HOME ALBA basado en IOT* (Tesis de licenciatura). Instituto Tecnológico Superior de Misantla. Recuperado de <https://rinacional.tecnm.mx/jspui/handle/TecNM/5242>
- Reynoso, E (2023). Aplicación móvil para monitoreo y control de un alimentador automático de mascotas (Reporte técnico No. 277). Recuperado de <http://cathi.uacj.mx/20.500.11961/26741>
- Velasco, C. (2020). *Dosificador de alimentos sólido y líquido para gatos con un sistema embebido con conexión a wifi y una base de datos* (Trabajo de titulación de ingeniería electrónica en control y automatismo). Universidad Católica de Santiago de Guayaquil. Recuperado de <http://repositorio.ucsg.edu.ec/handle/3317/14322>