

Título del Proyecto de Investigación
al que corresponde el Reporte Técnico:

Aplicación móvil para monitoreo y control de un alimentador de mascotas

Tipo de financiamiento

Sin financiamiento

Fecha de Inicio: 01/01/2022
Fecha de Término: 30/04/2023

Tipo de Reporte

Parcial

Final

Autor (es) del reporte técnico:

Manuel de Jesús Nandayapa Alfaro
Francesco José García Luna
Osslan Osiris Vergara Villegas
Elva Lilia Reynoso Jardón
Quirino Estrada Barbosa
Oscar Tenango Pirin

TÍTULO DEL REPORTE TÉCNICO

Resumen del reporte técnico en español (máximo 250 palabras)

El presente proyecto se centra en el desarrollo de una aplicación para dispositivos móviles con el funcionamiento de control y monitoreo de un alimentador automático de mascotas. El documento presenta un diseño de una estructura física que porte todos los componentes eléctricos y logre realizar una conexión por medio de un servidor, utilizando el protocolo MQTT para mensajería en tiempo con la aplicación desde cualquier parte del mundo utilizando una red Wifi. Mediante una serie de pruebas presentan los tiempos de respuesta del alimentador después de entrar a la aplicación y presionar el botón y utilizando la misma red en la que está conectado el alimentador. Los tiempos máximos de respuesta en el servomotor fueron de 5 segundos mientras que los tiempos de respuesta para los sensores fueron de 1 segundos.

Resumen del reporte técnico en inglés (máximo 250 palabras):

This project focuses on the development of an application for mobile devices with the control and monitoring operation of an automatic pet feeder. The document presents a design of a physical structure that carries all the electrical components and manages to make a connection through a server, using the MQTT protocol for messaging in time with the application from anywhere in the world using a Wi-Fi network. Through a series of tests, they present the response times of the feeder after entering the application, pressing the button, and using the same network in which the feeder is connected. The maximum response times in the servo motor were 5 seconds while the response times for the sensors were 1 second.

Palabras clave: MQTT, App, alimentador de mascotas, IoT.

Usuarios potenciales (del proyecto de investigación)

Aquellas personas con poca disponibilidad de tiempo y aquellas con algún impedimento de movimiento temporal o permanente.

Reconocimientos

No aplica.

1. Introducción

Con el rápido desarrollo de la ciencia y la tecnología en los últimos años, las personas se han rodeado de dispositivos tecnológicos como los teléfonos inteligentes y los dispositivos se han convertido en una parte indispensable de la vida. Los programas de aplicación (apps) en los teléfonos inteligentes brindan una gran cantidad de utilidades [1-2]. Para que estas aplicaciones brinden dichos servicios, requieren datos de usuario como nombre, correo electrónico, dirección, sexo, etcétera. [3-4]. Estas aplicaciones pueden conectarse a través de módulos bluetooth o wifi a través de microcontroladores como Arduino. Así, se pueden mejorar proyectos complejos y las aplicaciones.

El desarrollo de aplicaciones móviles tiene muchísimas maneras de ser utilizadas, como el entretenimiento, aprendizaje, navegación web, música, entre otros. Se puede apreciar su amplia aplicación en diferentes proyectos como plantea [5], desarrolló una aplicación móvil teniendo como objetivo lograr la interacción con un Arduino Uno de manera inalámbrica por medio de bluetooth.

2. Planteamiento

El tiempo muchas veces es un factor realmente importante la vida de las personas, tanto que la mayoría de las ocasiones faltan horas para realizar algunas de las actividades planeadas. A consecuencia de eso se descuida a las mascotas y se olvida rellenar su plato de comida y de agua. Existen también la ocasión donde la familia quiere darse el lujo de salir de vacaciones por cierto tiempo, pero no existe quien pueda alimentar sus mascotas en su ausencia.

Desafortunadamente las personas están propensas a sufrir accidentes desde leves hasta graves, en las peores situaciones terminan con dificultades para caminar o realizar algunos movimientos físicos se vuelve bastante complicado realizar algunas tareas como atender a sus mascotas. Las cantidades de comida que una mascota debe de consumir está definida según la especie, raza, peso y edad. Muchas veces la persona a cargo de la mascota desconoce dicha información, como por ejemplo no sabe cuánto alimento debe

consumir y cuantas veces al día, por lo que la mascota puede pasar hambre o comer de más.

2.1 Antecedentes

Desde que el Internet empezó a ser importante para el mundo se realizaron intentos para conectar cosas con el Internet. De acuerdo con [6] se pudo conectar por primera vez una tostadora que podía encenderse y apagarse a través de internet. Por otro lado, en [7] dieron la primera descripción del funcionamiento sensores y dispositivos similares conectados a la Internet. El internet de las cosas (IoT) se entiende como la conexión entre humanos, computadoras y cosas. La tecnología IoT utiliza la identificación de elementos, sensores, sistemas de posicionamiento global, escáner, laser, Arduino, entre otros., para conectar cualquier artículo bajo protocolo de Internet (IP). Con una dirección IP única pude lograrse posicionamiento, seguimiento, monitoreo y administración inteligente de artículos. La administración de estos puede hacerse en tiempo real. Por ejemplo, controlar los interruptores eléctricos en una casa inteligentes desde un dispositivo móvil de manera remota por medio de programas de aplicación (apps) [8].

Las apps en los teléfonos inteligentes han brindado una gran cantidad de utilidades [9]. Las aplicaciones móviles pueden llegar a ser una gran herramienta como señaló [10] desarrollando una aplicación para dispositivos móviles con la capacidad de mover un brazo robótico que al mismo tiempo transmitiera imágenes en tiempo real desde una cámara. No solo en la industria se puede aplicar el uso de aplicaciones móviles para optimizar algún proceso como señala [11] con un novedoso sistema inalámbrico basado en el desarrollo de una aplicación móvil que se conectaba a un Arduino y sensores que adquirirían datos del movimiento de un molino, procesaba y enviaba a la oficina donde se trabajaba con la información. De igual manera como lo hace notar [12] se desarrolló una aplicación móvil que ayudaba a los usuarios a tener el control y monitoreo del clima dentro de una habitación y del gasto innecesario de energía al dejar dispositivos encendidos por largos lapsos de tiempo. Para que la IoT pueda trabajar en tiempo real se mantiene utilizando servicios en la nube en varios servidores informáticos locales conectados que luego transferirán la accesibilidad de la información a los centros de datos interconectados en todo el mundo para todos los teléfonos conectados con ubicaciones en todo el mundo. Se presenta un caso de IoT desarrollando una aplicación

móvil para el monitoreo y control de un alimentador de mascotas como una alternativa de ayuda para personas que presentan problemas de movilidad o ausencia [13].

2.2 Marco teórico

2.2.1 Protocolo MQTT

El protocolo Message Queuing Telemetry Transport (MQTT Por sus siglas en inglés), conecta las redes y los dispositivos por medio de middleware y aplicaciones. La conexión se establece utilizando patrones de comunicación máquina-servidor, servidor-servidor, maquina-maquina y mecanismos de enrutamiento. El puerto MQTT que con el que trabaja por defecto es TCP/IP 1883. MQTT tiene diferentes tipos de servidores como mosquito, chiven, y paho MQTT. Los protocolos de seguridad que utiliza en la capa de transporte son (TLS)/ Secure Sockets Layer (SSL) y proporcionan seguridad en las comunicaciones a través de la red informática que es utilizada en diferentes aplicaciones como el correo electrónico, la navegación web, el envío de faxes por internet, VoIP y la mensajería instantánea.

MQTT es considerado el protocolo de conexión más favorable para la conexión máquina-maquina dentro de la tecnología IoT. Utiliza el patrón publicación/suscripción para proporcionar una implementación más sencilla y una transición flexible [14]. En la Figura 1 se muestra cómo funciona el patrón mencionado de MQTT.

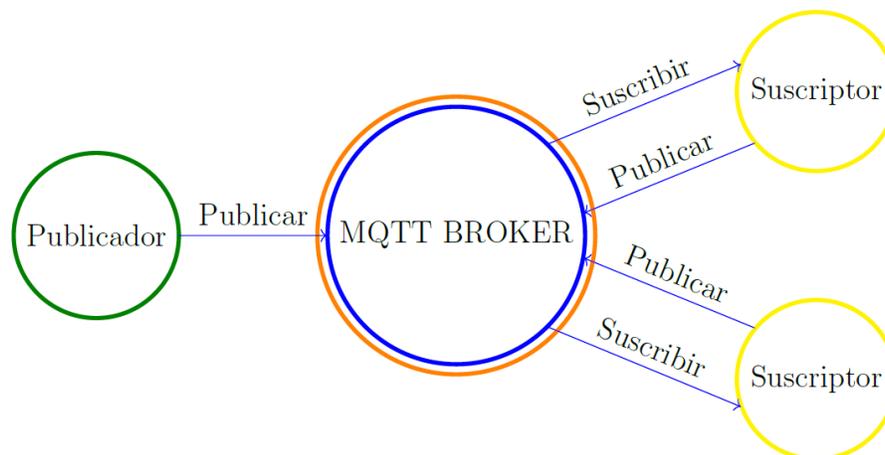


Figura1. Proceso de publicación/suscripción utilizada por MQTT.

2.2.2 Plataforma Arduino

Los circuitos de Arduino integran todos los componentes necesarios que permiten un uso poco complejo y rápido del microcontrolador. La simplificación que genera el microcontrolador se orienta a hacer accesible a toda la generación y programación de dispositivos interactivos. Los objetos mencionados pueden contener captadores, indicadores luminosos o interruptores deseados.

Las tarjetas son equipadas con una serie de conectores estandarizados para la conexión de módulos compatibles de nombre *shields*. Las extensiones de hardware que ofrece permiten al añadir funciones originales a el proyecto. Las tarjetas cuentan con conectividad de tipo USB para incorporar fácilmente el programa realizado. Arduino cuenta con un botón de *reset* el cual es una función física que le da la oportunidad al microcontrolador de reiniciar su estado. El microcontrolador ejecuta cíclica e infinitamente el programa cargado en su memoria. La función de este botón es interrumpir el programa y regresarlo al inicio para que comience desde cero.

El microcontrolador tiene dos formas de comunicarse con el mundo exterior, las entradas y salidas. Comúnmente se conoce como entradas y salidas a los pines de los chips electrónicos, estas pines reciben o emiten señales lógicas que son interpretadas por el microcontrolador. En las tarjetas se distribuyen las entradas y salidas alrededor de su circuito haciendo más sencillo añadir o retirar los componentes sin obligar al usuario a soldarlas. El número de entradas y salidas varía dependiendo del modelo del Arduino. Los conectores funcionan con 5V y solamente pueden recibir y entregar 40mA de corriente.

2.2.3 Aplicaciones para teléfonos móviles

Android es un sistema operativo para dispositivos móviles basado en Linux y está enfocado para ser utilizado en teléfonos inteligentes, tabletas, Google TV, entre otros dispositivos. Este sistema operativo es desarrollado por Open Handset Alliance, que es liderada por la compañía de Google.

La estructura del sistema operativo de Android está compuesta de aplicaciones que son ejecutadas desde un *framework* Java de aplicaciones. Las bibliotecas son escritas en el lenguaje C e incluyen un administrador de interfaz gráfica, un *framework* OpneCore, una base de datos relacional SQLite, una interfaz de programación de API grafica OpneGL

ES 2.0 3D, un motor de renderizado Webkit, un motor grafico SGL, SSL y una biblioteca estándar de C Bionic.

Las aplicaciones son normalmente desarrolladas en el lenguaje Java con Android Software Development kit, pero existen otras herramientas que pueden ayudar en el desarrollo de aplicaciones móviles, como el kit de desarrollo nativo para aplicaciones o algunas extensiones en C, C++, entre otros lenguajes de programación [15].

La interfaz de usuario de la página web del MIT app inventor incluye dos editores principales: editor de diseño es una interfaz con mucha ayuda visual donde solamente se necesita arrastrar y soltar para diseñar los elementos de la interfaz de usuario (UI) de la aplicación. El editor de bloques es un entorno donde los inventores de aplicaciones pueden realizar diseños visuales de la lógica de sus aplicaciones utilizando bloques que están codificados por colores que encajan como una pieza de rompecabezas para describir el programa. En la Figura 2 se muestra un ejemplo del editor de bloques.

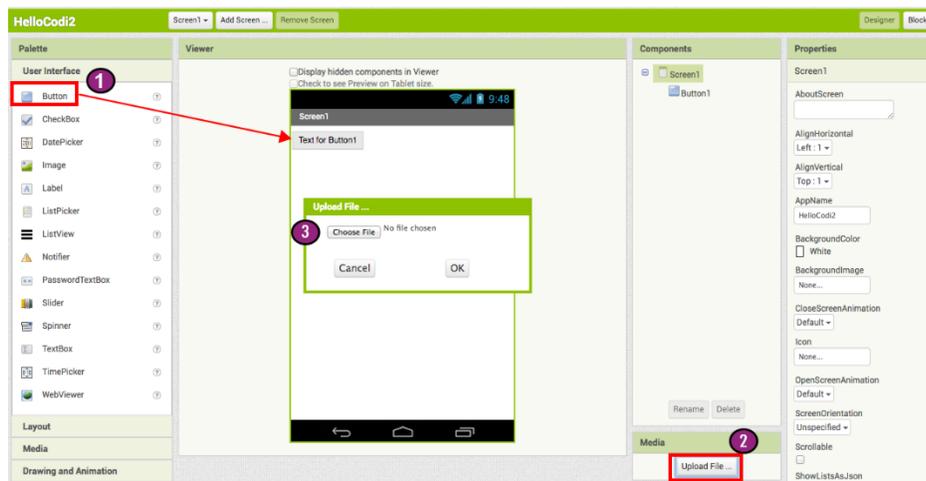


Figura 2. Editor de Bloques [16].

Para facilitar a los usuarios el desarrollo y pruebas, app inventor ofrece una aplicación móvil con el nombre de APP Inventor Companion donde los desarrolladores pueden probar y ajustar los componentes de sus aplicaciones en tiempo real. De esta forma es más fácil para los usuarios crear rápidamente una aplicación móvil [16].

3 Objetivos (general y específicos)

Desarrollar un sistema de alimentación de mascotas en un ambiente de IoT monitoreado y controlado por una aplicación móvil.

3.1 Objetivos específicos

1. Desarrollar una aplicación móvil con la información suficientes para el monitoreo y control del alimentador.
2. Diseñar un alimentador de mascotas que pueda contener el circuito eléctrico y un contenedor de comida.
3. Conectar un módulo de *NodeMCU ESP8266a* un servido *MQTT* para el control de sensores y actuadores.
4. Diseñar un programa en *Arduino* donde pueda extraer datos de los sensores y manipular actuadores.

5. Metodología

En la Figura 3 muestra un mapa conceptual donde se describe los pasos que se tomaron para el desarrollo de este trabajo.



Figura 3. Pasos del diseño del alimentador y app de alimento.

5.1 Modelo físico

La estructura principal se diseñó en el software de SolidWorks, la cual es una estructura que puede cargar un contenedor de alimento y adaptar todos los componentes del circuito eléctrico. Una vez realizado el diseño se consideraron los materiales con los que se construiría el alimentador, tomando en cuenta que fuera resistente, aspecto presentable y apto para la mascota. El diseño del alimentador se puede observar en la Figura 4.

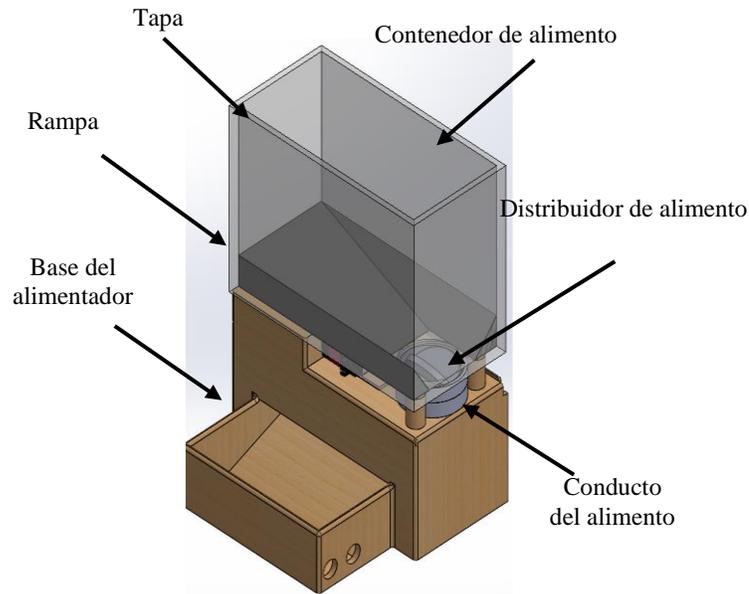


Figura 4. Alimentador de mascotas.

Para la construcción de la estructura física del alimentador de mascotas se buscó imprimir en 3D la base, tapa, distribuidor de alimento, rampa y conducto para el contenedor de alimento con la finalidad de tener calidad en las piezas para ensamblar a la perfección y reducir fallas. El material con el que se imprimió fue ABS-M30 (acrilonitrilo butadieno estireno) los materiales son ideales para modelos de concepto y piezas con es la opción ideal para modelos de concepto y piezas con requisitos moderados. Además, se caracteriza por su fuerza y dureza, al mismo tiempo que es liviana y resistente. Por otro lado, el contenedor fue hecho con acrílico.

5.2 Diseño del circuito eléctrico

El objetivo de la aplicación web es lograr la interacción con el alimentador desde cualquier parte del mundo mientras se tenga una conexión a internet. Se utilizó un módulo de wifi NodeMCU ESP8266, una tarjeta de desarrollo similar a Arduino, especialmente orientada al Internet de las cosas (IoT). Tiene 1 pin de entradas análoga y 13 digitales. Para el desarrollo de aplicaciones, se puede escribir el código en lenguaje Lua usando el *framework* de Arduino. Además, permite utilizar las librerías desarrolladas para Arduino y trabajar dentro de su entorno de desarrollo integrado (IDE), que es la plataforma para escribir y cargar los programas. En la Figura 5 se puede apreciar el chip y sus entradas análogas y digitales.

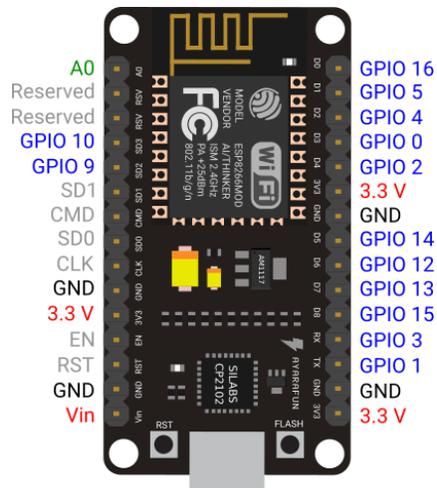


Figura 5. Módulo NodeMCU ESP8266 [17].

Para realizar el movimiento del distribuidor de alimento se utilizó un servomotor de 20 kg para mover el peso de las croquetas. En la Figura 6 se puede apreciar el motor acoplado al distribuidor del alimento. Las características del motor se pueden observar en la Tabla 1.

Tabla 1. Características físicas del servomotor.

Descripción del servomotor	Características
Voltaje mínimo - Voltaje máximo	4.8V - 6.8V
Torque	18 kg - 21.5 kg.
Velocidad	0.16 sec/60° a 5v y .14 sec/60° a 6.8v

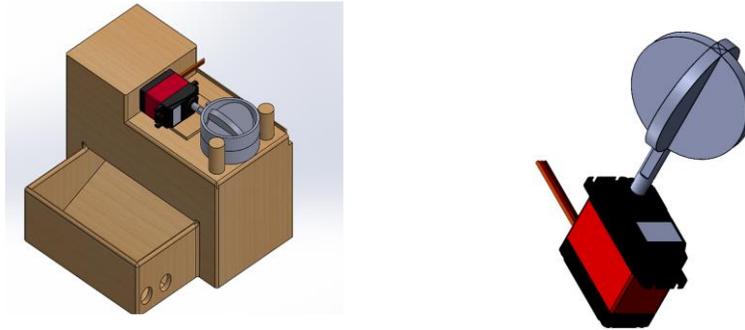


Figura 6 . Acoplamiento de servomotor y distribuidor de alimento

Como se menciona con anterioridad, el servomotor necesita 4.8 - 6.8 volts para funcionar, el chip NodeMCU ESP8266 solamente energiza con 3.3 volts por lo que no tiene la capacidad de energizar el servomotor. Para la alimentación de potencia al servomotor se utilizó una fuente de 5V.

La aplicación mostrará al usuario cuando el plato este lleno o vacío y el nivel de comida del contenedor, para conseguir eso se utilizará 2 sensores ultrasónicos HC-SR04 para medir la distancia en el contenedor y en el plato. Los sensores mandaran la señal al chip NodeMCU ESP8266. La Figura 7 muestra la apariencia física que tiene el sensor HC-SR04.



Figura 7. Sensor ultrasónico HC-SR04

En la Figura 8 se muestra la topología de todo el circuito eléctrico, donde se muestra cómo se conectaron los sensores HC-SR04 y el servomotor al NodeMCU ESP8266 y la forma en la que se alimenta el servomotor con el Arduino uno.

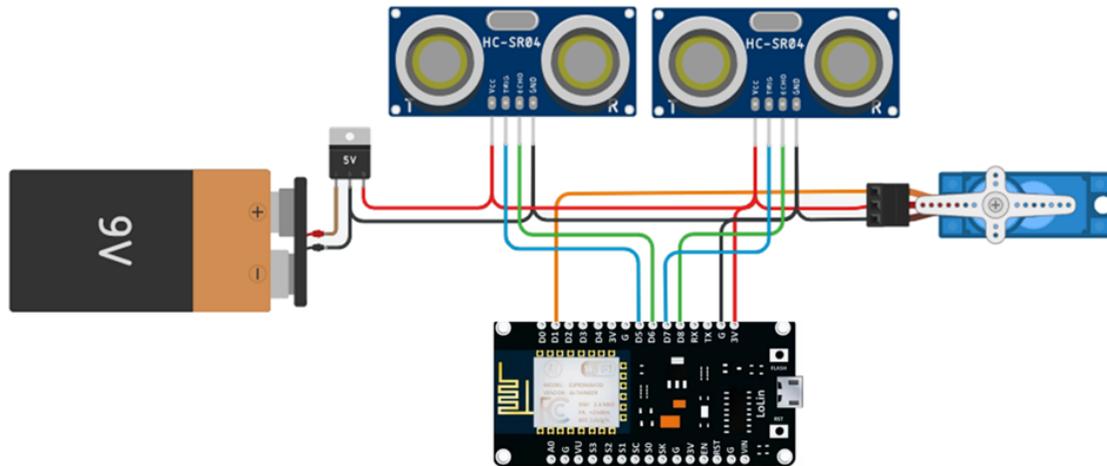


Figura 8. Circuito electrónico del control

5.3 Desarrollo de la programación

En la etapa de programación del protocolo de comunicación en el NodeMCU ESP8266, primero se utilizó el software “MQTT explorer” como monitor del servidor, donde se configura el servidor “test.mosquitto.org” en el puerto 1883. El software tiene diferentes funciones como mandar mensajes al servidor, escoger el nombre del tópico, configurar diferentes servidores de pago, ver el historial de mensajes enviados al servidor y tener la capacidad de revisarlo desde cualquier parte del mundo solamente usando una red wifi. En la Figura 9 se muestra la interfaz del software.

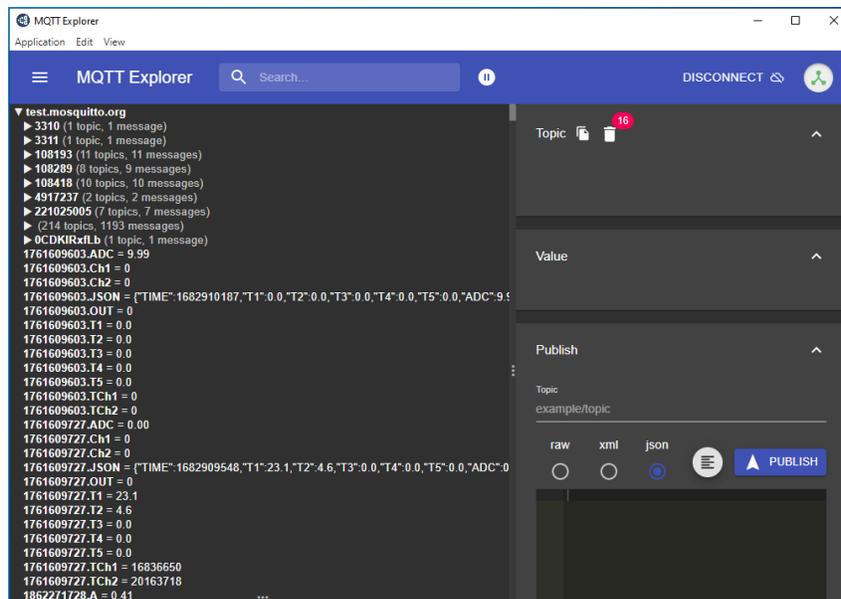


Figura 9. Interfaz del software MQTT explorer

Para la programación del NodeMCU ESP8266 se utilizó el software de IDE de Arduino. El programa inicia incluyendo las librerías necesarias para realizar la programación completa. También se definen las variables que se utilizaron a lo largo del programa para realizar las diferentes operaciones.

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
// WiFi
const char *ssid = ""; // Introducir el nombre de la red wifi
const char *password = ""; // Introducir la contraseña
// MQTT Broker
const char *mqtt_broker = "test.mosquitto.org"; // Introducir el nombre o IP del broker
const char *topic = "AlimentadorDeMascotas/";
const int mqtt_port = 1883;
WiFiClient espClient;
PubSubClient client(espClient);
void setup() {
  Serial.begin(115200); // velocidad serial 115200;
  WiFi.begin(ssid, password); // Conexion al wifi
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Conectandose al WiFi..");
  }
  Serial.println("Conectado");
  //Conexion al mqtt broker
  client.setServer(mqtt_broker, mqtt_port);
  client.setCallback(callback);
  while (!client.connected()) {
    String client_id = "esp8266-client-";
    client_id += String(WiFi.macAddress());
    Serial.printf("Este cliente %s se conecto al Broker test.mosquitto.org\n", client_id.c_str());
    if (client.connect(client_id.c_str())) {
      Serial.println("Broker conectado");
    } else {
      Serial.print("Fallo la conexion ");
      Serial.print(client.state());
      delay(2000);
    }
  }
}
// Publica y suscribe
client.publish(topic, "Hola Alimentador de mascotas ESP8266!");
client.subscribe(topic);
}
void callback(char *topic, byte *payload, unsigned int length) {
  Serial.print("Mensaje recibido en el topico: ");
  Serial.println(topic);
  Serial.print("Mensaje:");
  for (int i = 0; i < length; i++) { Serial.print((char) payload[i]); }
  Serial.println();
  Serial.println("-----");
}
void loop() {
  client.loop();
}

```

Programa de arduino NodeMCU ESP8266

5.4 Aplicación móvil

La aplicación móvil fue programada en la página web <http://ai2.appinventor.mit.edu>, donde se diseñó la interfaz de la aplicación web, la programación y la conectividad con el servidor utilizando el protocolo MQTT. En la interfaz de la aplicación primero presenta una caja de texto donde el usuario escribe el nombre del servidor y un botón que al ser presionado comienza el proceso de conectividad de la aplicación con el servidor y al mismo tiempo cambia el texto del botón a “conectando”. Si el usuario escribe de manera errónea el nombre del servidor o el servidor no está activo, no se logrará la conexión y lanzara una notificación indicando que no se ha logrado establecer la conexión. En la Figura 10 se muestra la interfaz fallando en el proceso de conexión.

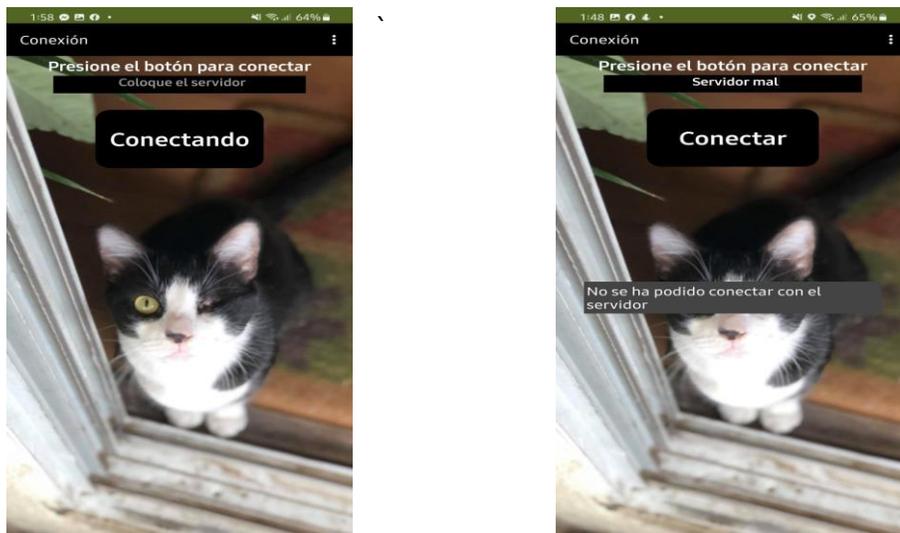


Figura 10. Interfaz de conexión de la aplicación indicando falla en la conexión.

Una vez conectado al servidor, cambia la interfaz de la aplicación, como se muestra en la Figura 11. Ahora la interfaz tiene un nuevo botón con el texto de “Alimentar” que activa el motor para servir la comida de la mascota en el plato. Además, contiene el estatus del plato indicando si se encuentra lleno o vacío. Si detecta que el plato de la mascota está lleno, deshabilitara el botón de alimentar para evitar servir dos veces la comida.



Figura 11. Interfaz principal de la aplicación

Mas abajo cuenta con un texto que dice “Contenedor” y una barra verde que marca el 100 por ciento. Esta barra mide el nivel de comida que le queda en el contenedor. Señala cuanto alimento queda por medio de porcentajes, cuando esté lleno señalará 100 % y de ahí bajará a 75, 50, 25 y 0%. Cada uno con su respectiva barra como se muestra en la Figura 12.

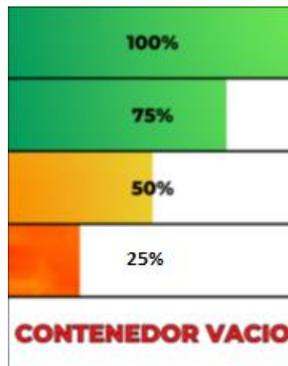


Figura 12. Barras de porcentaje del contenedor

El alimentador está condicionado, si detecta que el plato aún está lleno, arrojará una notificación que va a indicar que el plato aún tiene comida y no servirá más. También, cuenta con la condición de notificar que el contenedor de comida está vacío y no permite servir comida. Pero teniendo el plato vacío y el contenedor mínimo con el 25 % de su

capacidad servirá el alimento sin ningún problema. En la Figura 13 se muestra ambos casos de las condiciones y como se muestra la notificación.

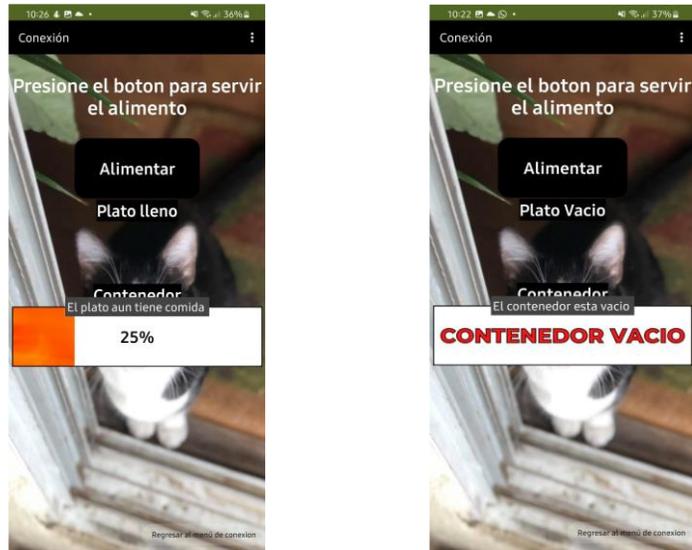


Figura 13. Aviso del estado del dispensador de comida.

Cuando el contenedor está lleno y el plato de la mascota vacío, se mostrará una notificación indicando que el alimento se está sirviendo como se muestra en la Figura 14.



Figura 14. Indicación del llenado de plato de comida para la mascota.

- 6. Instituciones, organismos o empresas de los sectores social, público o productivos participantes (Si aplica). No aplica
- 7. Resultados

Se realizaron 5 pruebas donde se midió el tiempo de respuesta del servomotor con un cronometro después de presionar el botón de alimentar en la aplicación. Las primeras 5 pruebas se hicieron a las 10 am tomando un máximo de tiempo de respuesta para el motor de 5 segundos. Los resultados se pueden apreciar en la Figura 15.

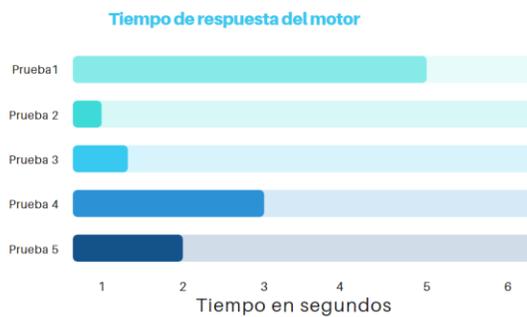


Figura 15. Tiempo de respuesta del servomotor día 1.

También se consideraron los tiempos de respuesta de los sensores de ultrasonido utilizados en el alimentador, para estas pruebas el tiempo fue de alrededor de 1segundo, como se puede apreciar en la Figura 16.



Figura 16. Tiempo de respuesta de los sensores día 1.

Se realizaron las mismas pruebas para medir el tiempo de respuesta de los sensores y el servomotor a la 10 am del siguiente día. Los resultados del funcionamiento del motor y sensores tomaron un máximo de tiempo de 2.5 segundos y un mínimo de 1 segundos como se observa en las Figuras 17 y 18.

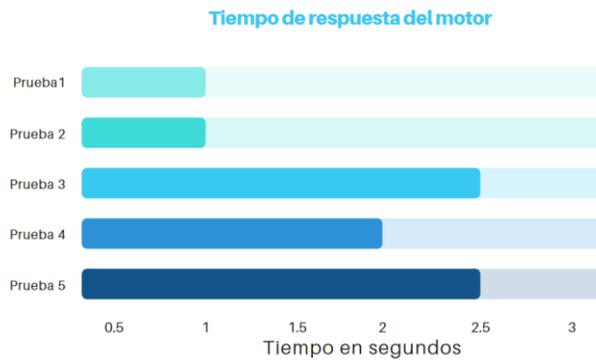


Figura 17. Tiempo de respuesta del servomotor día 2.

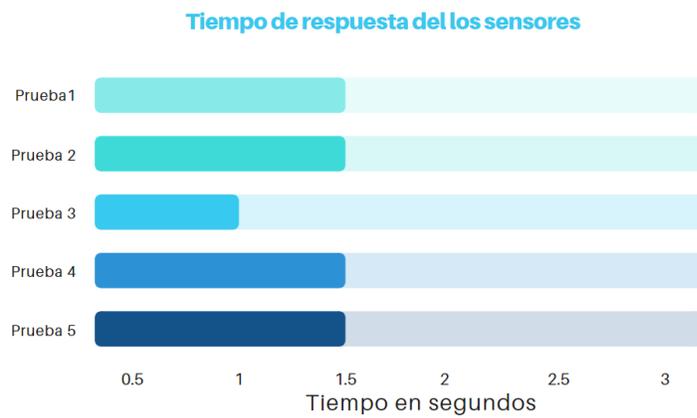


Figura 18. Tiempo de respuesta de los sensores día 2.

Finalmente, en la Figura 19 puede apreciar el dispositivo final en funcionamiento bajo el control de la aplicación indicando la cantidad de alimento disponible para la mascota.



Figura 19. Alimentador de mascotas

8. Productos generados

Reporte técnico

Tesis

Artículo (publicable con modificaciones). El artículo se envió a la revista RECIT aceptado con modificaciones, fue corregido y enviado nuevamente. Se espera la aceptación total y publicación del artículo en los próximos días.

9. Conclusiones

El propósito de esta investigación fue desarrollar una aplicación para dispositivos móviles para el control y monitores de un alimentador de mascotas con el objetivo de optimizar el tiempo de los dueños facilitando el proceso de alimentar a su mascota, además de poder hacerlo cuando están fuera de la casa o de la ciudad. El dispositivo con la app apoyar a las personas que sufren de manera temporal o permanente alguna dificultad física, convirtiendo una tarea fácil para estas personas. Con base a los resultados obtenidos por medio del análisis de datos, se puede concluir lo siguiente:

Le da al usuario apoyo visual para conocer el estatus del alimentador y da la oportunidad de que pueda servir la comida sin ningún problema desde cualquier parte del mundo. El motor de 20kg tiene el torque necesario para hacer girar el distribuidor de alimento implementado por el peso de la comida dentro del contenedor.

Puede recibir y enviar datos desde cualquier parte del mundo con la única condición de que exista conexión de internet en el dispositivo móvil, esto para permitir la comunicación con el servidor. El tiempo de respuesta del servomotor y sensores depende de la calidad de conexión que tenga el usuario, así como el tráfico de datos que presente el servidor. De acuerdo a las pruebas realizadas el servomotor responde con tiempos mínimos de 0.5 segundo y 5 segundos como máximo, por otro lado, los sensores presentaron tiempos mínimos de 1 segundos y como máximos de 1.5.

10. Mecanismos de transferencia. (Si aplica). Reporte técnico, manual de uso del equipo.

11. Contribución e impacto del proyecto

El proyecto ayudará a mostrar los proyectos desarrollados en la Universidad Autónoma de Ciudad Juárez como creadora del proyecto. La máquina dispensadora de alimento servirá para alimentar las mascotas sin la necesidad de estar presente o a personas incapaces de realizar esta actividad.

12. Impacto económico, social y/o ambiental en la región

El impacto dentro de la sociedad se verá reflejado en la disposición de tiempo, ya que las personas no se preocuparán por ciertas actividades como lo es alimentar a las mascotas.

13. Referencias (bibliografía)

[1] M. Hatamian, Engineering Privacy in Smartphone Apps: A Technical Guideline Catalog for App Developer, IEEE Access, Volume:8, pp 429 - 445, 2020.

[2] A. A. Galadima, Arduino as a learning tool, in 2014 11th International Conference on Electronics, Computer, and Computation (ICECCO), 2014.

[3] M. Tovey, Drawing and CAD in industrial design in Design Studies, Volume 10, Issue 1, pp 24-39, 1989.

- [4] R. S. Rosli, M. H. Habaebi y Md. R. Islam, Characteristic Analysis of Received Signal Strength Indicator from ESP8266 WiFi Transceiver Module in ,2018 7th International Conference on Computer and Communication Engineering (ICCCCE), 2018.
- [5] X. Villamil, T. Guarda, G. Ninahualpa Quina, Agile software development for mobile applications and wireless interaction with hardware development board (Arduino), in IEEE, pp 13-16, 2018.
- [6] A. Aguilar-Hernandez, J. Dias Rodrigues, A. Tukker, Macroeconomic, social and environmental impacts of a circular economy up to 2050: A meta-analysis of prospective studies, *Journal of Cleaner Production*, Volume 278, 1,2021.
- [7] S. Ding,A. Tukker,D. Hauke, Oportunidades y riesgos de las tecnologías de Internet de las cosas (IoT) para modelos comerciales circulares: una revisión de la literatura, *Diario de Gestión Ambiental*, Volumen 336, 15 , 2023.
- [8] M.Q. Chau, X.P. Nguyen,T.T. Huynh,V.D. Chu,T.H. Le,T.P. Nguyen,D.T. Nguyen, Perspectivas de la aplicación de tecnologías avanzadas basadas en IoT en el proceso de refabricación hacia el desarrollo sostenible y el uso eficiente de la energía, Fuentes de energía, Parte A: recuperación, Utilización y Efectos ambientales, *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*,2021.
- [8] Almaraz-Menéndez, F Maz-Machado ,A. López, Tecnología móvil y enseñanza de las matemáticas: una experiencia de aplicación de app Inventor”, in *Revista Épsilon*, pp 77-86, Septiembre 2020
- [10] X. Villamil y T. Guarda, App Móvil Desarrollada con Metodología Ágil para IoT Controlada desde una Red LAN/WAN con Placa de Desarrollo de Hardware Libre (Arduino), in *Revista Ibérica de Sistemas e Tecnologías de Información*, pp 379-392, noviembre 2018.
- [11] N. Paucara, Sistema de Control Domótico Basado en Arduino, Aplicación Móvil y Voz, in *Universidad Mayor de San Andrés*, pp 10-13, 2016
- [12] R. Hernández, Estudio para el diseño de un prototipo de sistema de entrenamiento Físico basado en Arduino y App móvil, in *Global Access to UPC knowledge*, pp 1-6, 2017.
- [13] A. Prieto, Diseño de la app y el control de una cámara video con Arduino, in *Universidad Politécnica de Valencia*, 2016.

- [14] Internet of Things: Survey and open issues of MQTT protocol, Mayo 2017.
- [15] J. Garrido Codo, TFC Desarrollo de aplicaciones móviles, in open access, 2013
- [16] Making Mobile Apps with App Inventor, Julio 2017.
- [17] 330 ohms, Diferencias entre un microprocesador y un microcontrolador, 2020.

14. Anexos

editorrecit@gmail.com



Para: Elva Lilia Reynoso Jardon; Manuel de Jesus Nandayapa Alfaro **y 5 más**

Tomamos una decisión sobre su envío a REVISTA DE CIENCIAS TECNOLÓGICAS (RECIT), "Exploring the internet of things: tools and case study : Explorando internet de las cosas: herramientas y caso de estudio ".

Nuestra decisión es: PUBLICABLE CON MODIFICACIONES

PORFAVOR ATENDER LAS CORRECCIONES USANDO EL FORMATO ADJUNTO

Revisor/a A:
Recomendación:

Instituto de Ingeniería y Tecnología

UACJ | UNIVERSIDAD AUTÓNOMA
DE CIUDAD JUÁREZ

DEPARTAMENTO:
INGENIERÍA INDUSTRIAL Y MANUFACTURA
No. OFICIO:
DIIM-179-1-22
ASUNTO:
PROYECTO DE INGENIERÍA

20 de mayo de 2022.

ARVIZU ASTORGA JOSÉ FRANCISCO
Presente.-

Por este conducto, se le notifica a Usted, que el trabajo titulado: "**Aplicación Móvil para Monitoreo de Llenado de Alimento**".

Ha sido revisado y aprobado por sus asesores, el comité de la clase de Proyectos de Ingeniería, la Academia de Ingeniería en Mecatrónica y el departamento correspondiente, por lo que se autoriza la impresión de ejemplares. Asimismo, se le comunica que la fecha de su acto recepcional, ha sido programada para el **viernes 27 de mayo de 2022**; se reúnen en la sala audiovisual del edificio H1 del Instituto de Ingeniería y Tecnología, los integrantes del jurado **a las 20:03 horas**.

Agradeceremos, envíe su tesis en formato electrónico a su asesor de proyecto, maestro de asignatura y realice su presentación, con el fin de efectuar y cumplir con sus trámites correspondientes.



DR. ERWIN ADÁN MARTÍNEZ GÓMEZ
Jefe del Departamento de Ingeniería
Industrial y Manufactura



M.C. MIGUEL ÁNGEL GARCÍA TERÁN
Coordinador de la Academia de
Ingeniería en Mecatrónica



DR. ÁNGEL FLORES ABAD
Maestro de la Asignatura

DEPARTAMENTO DE
INGENIERÍA INDUSTRIAL
Y MANUFACTURA



DR. MANUEL DE JESÚS NANDAYAPA ALFARO
Asesor del Proyecto



DRA. ELVA LILIA REYNOSO JARDÓN
Co-asesor

a. Taxonomía de los Roles de Colaborador (con las actividades logradas)

Roles	Definición de los roles	Nombre de él(la) investigador(a)	Figura	Grado de contribución	Actividades logradas durante el proyecto	Tiempo promedio semanal (en horas) dedicado al proyecto
1. Responsabilidad de la dirección del proyecto	Coordinar la planificación y ejecución de la actividad de investigación. Organiza los roles de cada colaborador, tiene la habilidad de identificar potenciales de cada individuo para generar una sinergia de equipo colaborativo.	Dra. Elva Lilia Reynoso	Director(a) del proyecto	- Principal	Desarrollaron el proyecto mediante la coordinación y asignación de tareas a los colaboradores.	10
2. Responsabilidad de supervisión	Elaborar la planificación de las actividades de la investigación (cronogramas y controles de seguimiento), describe los roles identificados por el director del proyecto y facilita el apoyo constante a todos los roles para conseguir un trabajo integral, coherente y que llegue a buen término.	Dra. Elva Lilia Reynoso Jardón	Supervisor del proyecto	- Principal	Encargado de la supervisión de documentos y de que cada una de las actividades se desarrollaran en tiempo y forma.	6
3. Realización y redacción de la propuesta	Preparación, creación y redacción de la propuesta de investigación, específicamente la redacción, revisión de coherencia del texto, presentación de los datos y la normatividad aplicable para garantizar el cumplimiento de los requisitos.	Dr. Osslan Ossiris Vergara Villegas	Redactor de la propuesta	- De apoyo	En base a la experiencia del doctor ayudo al diseño del equipo	10
4. Desarrollo o diseño de la metodología	Contribuir con el diseño de la metodología, modelos a implementar y el sustento teórico, empírico y científico para la aplicabilidad de los instrumentos en la ejecución del proyecto.	M. Francesco José García Luna	Diseñador de la metodología	- De apoyo	Propuesta de la metodología a seguir del proyecto.	10
5. Recopilación/recolección de datos e información	Ejecuta las estrategias propuestas en acciones encaminadas a obtener la información, haciendo la recopilación de datos y la inclusión de la evidencia en el proceso.	Dr. Oscar Tenango Pirin	Recopilador de datos	- De apoyo	En base al análisis del comportamiento de la máquina obtener los datos necesarios	6

					para realizar ajustes y mejoras.	
6. Elaboración del análisis formal de la investigación	Aplicar métodos estadísticos, matemáticos, computacionales, teóricos u otras técnicas formales para analizar o sintetizar los datos del estudio. Verifica los resultados preliminares de cada etapa del análisis, los experimentos implementados y otros productos comprometidos en el proyecto.	Dr. Quirino Estrada Barbosa	Analista de datos	- De apoyo	Durante la obtención de resultados se analizaron para presentarlos y reportarlos para la validación del equipo	10
7. Preparación, creación y/o presentación de los productos o entregables	Preparar la redacción del reporté técnico de avance parcial y el reporte técnico final. Se hace la revisión crítica, la recopilación de las observaciones y comentarios del grupo de investigación. Y finalmente se procede a la edición del documento a entregar.	Dr. Manuel de Jesús Nandayapa Alfaro	Editor de reportes técnicos Y documentos	-De apoyo	Los doctores se dedicarán tiempo a la redacción de los documentos, así como complementar la investigación.	10

i. Estudiantes participantes en el proyecto

Nombre de estudiante(s)	Matrícula	Tiempo promedio semanal (en horas) dedicado al proyecto	Actividades logradas en la ejecución del proyecto
José Francisco Arvizu Astorga	167903	8	Desarrollo de la app para control del alimento y desarrollo de planos en software CAD.