



Estudio de casos



# Explorando internet de las cosas basado en ESP8266: herramientas y caso de estudio

## *Exploring the internet of things based on ESP8622: tools and case study*

Elva Lilia Reynoso Jardón , Manuel Nandayapa , Quirino Estrada Barbosa , Raul Ñeco Caberta , Meilynn Jeyli Pineda Gugenhul , José Alfredo Ramirez Monares , José Francisco Arvizu Astorga 

Universidad Autónoma de Ciudad Juárez, Manuel Díaz H. No. 518-B Zona Pronaf. Condominio, 32315 Cd Juárez, Chihuahua, México

**Autor de correspondencia:** Elva Lilia Reynoso Jardón, Universidad Autónoma de Ciudad Juárez, Manuel Díaz H. No. 518-B Zona Pronaf Condominio, 32315 Cd Juárez, Chihuahua, México. E-mail: [elva.reynoso@uacj.mx](mailto:elva.reynoso@uacj.mx). ORCID: 0000-0002-0729-2822.

**Enviado:** 28 de Mayo del 2023

**Aceptado:** 14 de Agosto del 2023

**Publicado:** 20 de Septiembre del 2023

**Resumen.** - *El presente proyecto se centra en la exploración del internet de las cosas basado en ESP8266 con el desarrollo de una aplicación para dispositivos móviles que integra el 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 de 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.*

**Palabras clave:** Aplicación móvil; Alimentador de mascotas; MQTT.

**Abstract.** - *This project focuses on exploring internet of things based on ESP8266 by 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. A series of tests of the response times of the feeder after entering the application, pressing the button, and using the same network to 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.*

**Keywords:** Mobile app; Pet feeder; MQTT.

## 1. Introducción

Desde que el Internet empezó a ser importante para el mundo se realizaron intentos para conectar cosas con el Internet. De acuerdo con [1], se pudo conectar por primera vez una tostadora que podía encenderse y apagarse a través de internet. Por otro lado, [2] y [3] dieron la primera descripción del funcionamiento de los sensores y dispositivos similares conectados a la Internet. El internet de las cosas (IoT por sus siglas en inglés) se entiende como la conexión entre humanos, computadoras y cosas [4]. La tecnología IoT utiliza la identificación de elementos, sensores, sistemas de posicionamiento global, escáner, láser, arduino, entre otros, para conectar cualquier artículo bajo protocolo de Internet (IP) [5]. Con una dirección IP se logra posicionamiento, seguimiento, monitoreo y administración inteligente de artículos. La administración de estos artículos puede hacerse en tiempo real [6]. Por ejemplo, controlar los interruptores eléctricos en una casa inteligentes desde un dispositivo móvil de manera remota por medio de programas de aplicación (apps) [7].

Las apps en los teléfonos inteligentes han brindado una gran cantidad de utilidades [8]. Las aplicaciones móviles pueden llegar a ser una gran herramienta como señaló [9], 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 usar aplicaciones móviles para optimizar algún proceso como señala [10] 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, los procesaba y enviaba a la oficina donde se trabajaba con la información. De igual manera como lo hacen notar [11] y [12] donde desarrollaron 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 el 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 para todos los teléfonos conectados con ubicaciones en todo el mundo [13, 14]. En este trabajo 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. Desde el punto de vista humanista, la alimentación de las mascotas siempre ha sido una actividad importante, sin embargo, en algunas ocasiones los dueños se les olvida alimentarlos. Por lo tanto, la solución propuesta contribuye como una herramienta de interconexión que implica actuadores, dispositivos inteligentes y conectividad por la nube, entre otros.

## 2. Herramientas en internet de las cosas

### 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, máquina-máquina y mecanismos de enrutamiento [15]. El puerto MQTT con el que trabaja por defecto es TCP/IP 1883. MQTT tiene diferentes tipos de servidores como mosquito, hivemq, y paho MQTT [16]. 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-máquina dentro de la tecnología IoT [17]. Este

protocolo utiliza el patrón publicación/suscripción para proporcionar una implementación más sencilla y una transición flexible [18]. En la Figura 1 se muestra cómo funciona el patrón mencionado de MQTT.

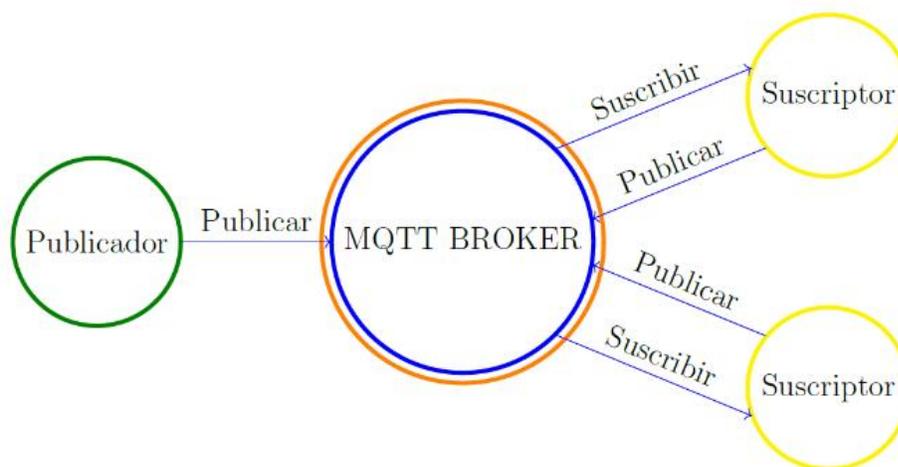


Figura 1. Proceso de publicación/suscripción utilizada por MQTT.

## 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 [19].

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, estos 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 40 mA de corriente [20].

### 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 [21].

La estructura del sistema operativo de android está compuesta de aplicaciones que son ejecutadas desde un *framework* Java de aplicaciones [22]. 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 gráfica OpneGL ES 2.0 3D, un motor de renderizado *Webkit*, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic [10].

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 [23, 24].

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 [25, 26]. 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 [27, 28].

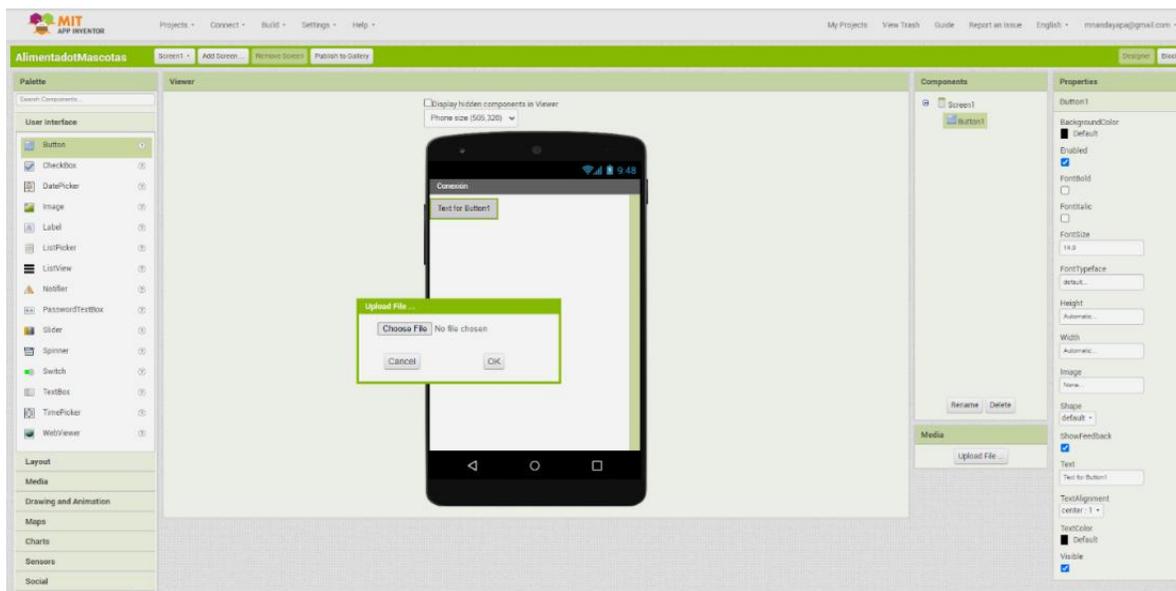


Figura 2. Editor de Bloques.

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 [29].

### 3. Caso de estudio: alimentador de mascotas

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.

#### 3.1 Diseño de la estructura física

Se diseñó en el software de SolidWorks una estructura que pudiera 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.

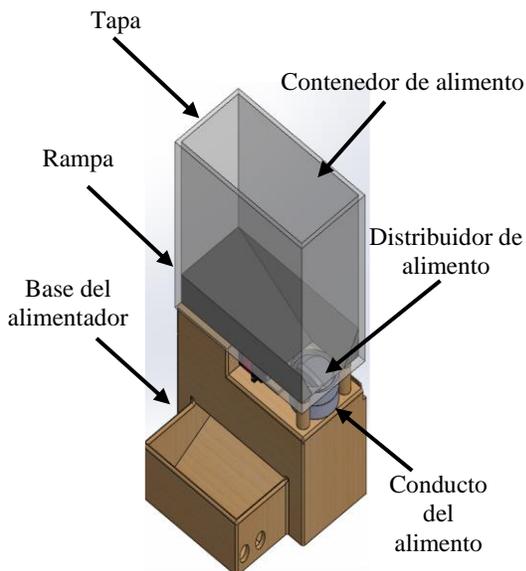


Figura 4. Alimentador de mascotas.

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 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.

#### 3.2 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 [30], una tarjeta de desarrollo basada en 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, el arduino 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 analógicas y digitales [31].

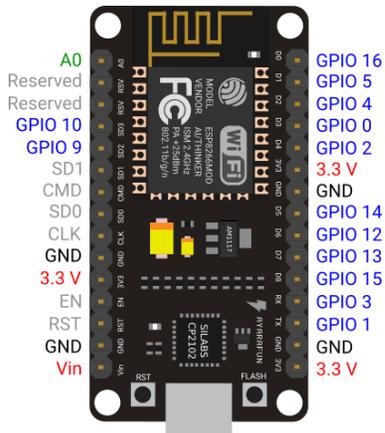


Figura 5. Módulo de WIFI NodeMCU ESP8266.

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 mandarían la señal al chip NodeMCU ESP8266. La Figura 7 muestra la apariencia física que tiene el sensor HC-SR04

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.

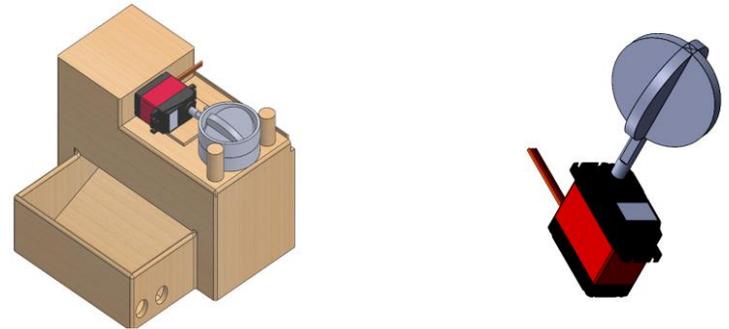


Figura 6 . Acoplamiento de servomotor y distribuidor de alimento [32].



Figura 7. Sensor ultrasónico HC-SR04.

En la Figura 8 se muestra la topología de todo el circuito eléctrico, de 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.

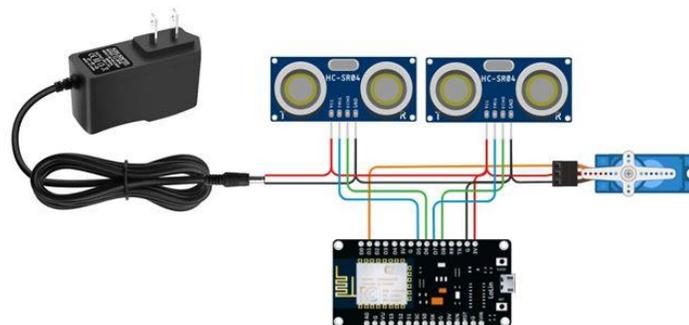


Figura 8. Circuito electrónico del control.

### 2.3 Control y monitoreo del servidor

En la etapa de programación del protocolo de comunicación en el NodeMCU ESP8266, primero se utilizó el software MQTT explore como monitor del servidor, donde se configura el servidor test.mosquitto.org en el puerto 1883. El software de monitoreo tiene diferentes funciones

como mandar mensajes al servidor, escoger el nombre del tópic, 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 de monitoreo.

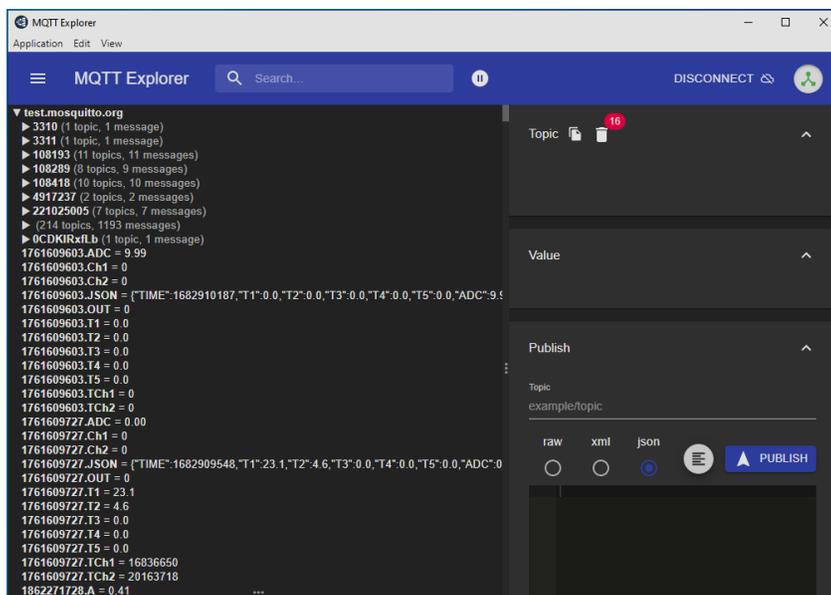


Figura 9. Interfaz del software MQTT explorer.

Para la programación del ESP8266 se utilizó el software de arduino. El programa inicia incluyendo las librerías necesarias para realizar la programación completa utilizando el lenguaje de programación en C. También se definen las variables que se utilizaron a lo largo del programa para realizar las diferentes operaciones.

## 4. Resultados

### 4.1 Aplicación móvil

La aplicación móvil fue realizada 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 [33]. 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 lanzará una notificación indicando que no se ha logrado establecer la conexión [34, 35]. En la Figura 10 se muestra la interfaz fallando en el proceso de conexión.

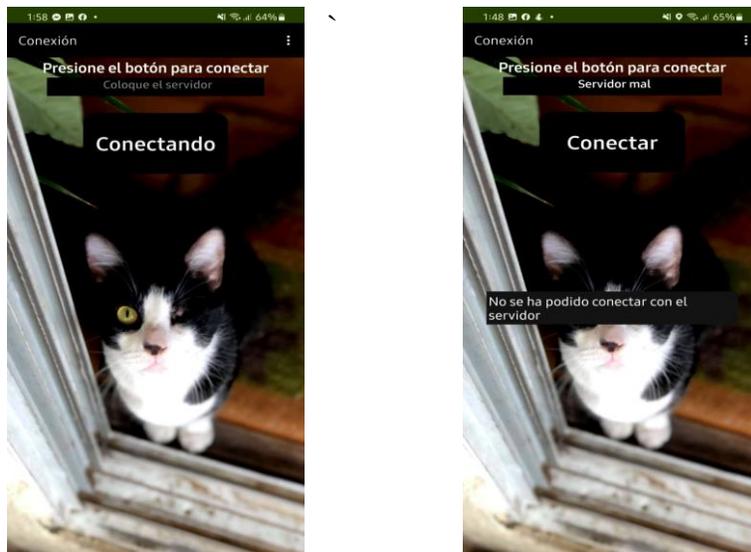


Figura 10. Interfaz de conexión de la aplicación indicando falla en la conexión.

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, deshabilitará el botón de alimentar para evitar servir dos veces la porción de comida. La cantidad de alimento correspondiente es a una porción para la mascota.

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 11. Interfaz principal de la aplicación.

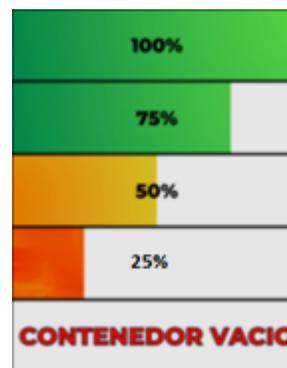


Figura 12. Barras de porcentaje del contenedor.

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.

#### 4.2 Resultados del alimentador

Se realizaron 5 pruebas donde se midió el tiempo de respuesta del servomotor con un cronómetro 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.

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

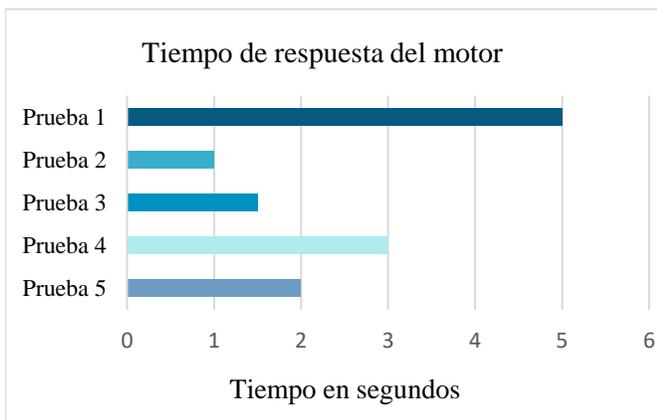


Figura 15. Tiempo de respuesta del servomotor día 1.

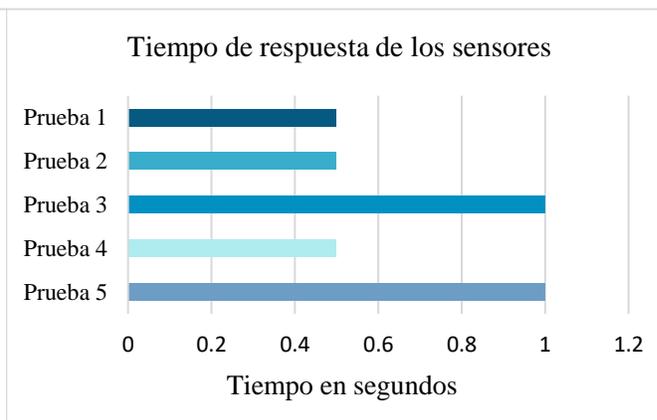


Figura 16. Tiempo de respuesta de los sensores.

Se realizaron las mismas pruebas para medir el tiempo de respuesta de los sensores y el servomotor a las 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 segundo como se observa en las Figuras 17 y 18

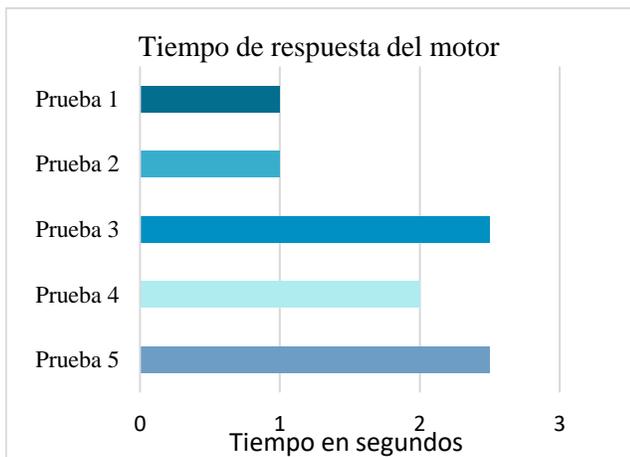


Figura 17. Tiempo de respuesta del servomotor.

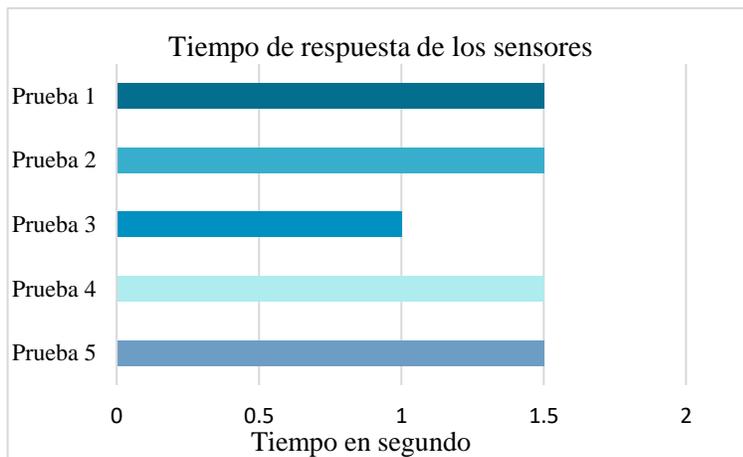


Figura 18. Tiempo de respuesta de los sensores para la prueba de sujeto 1.

Finalmente, en la Figura 19 puede apreciar el dispositivo final en funcionamiento.



**Figura 19.** Alimentador de mascotas.

## 5. Conclusiones

El propósito de esta investigación fue desarrollar una app para dispositivos móviles para el control y monitores de un alimentador de mascotas. El dispositivo con la app apoyará a las personas que sufren de manera temporal o permanente alguna dificultad física, convirtiendo una tarea fácil para estas personas. Con base en los resultados obtenidos por medio del análisis de datos, se puede concluir lo siguiente:

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

El usuario 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 en el servidor.

## 6. Reconocimiento de autoría

*Elva Lilia Reynoso Jardón y Manuel de Jesús Nandayapa Alfaro:* Conceptualización; Recursos; Ideas; Metodología; Análisis formal; Investigación; Recursos; Análisis de datos; Borrador original. *Quirino Estrada Barbosa:* Conceptualización; Ideas; Investigación; Análisis de datos; Escritura: revisión y edición. *Raúl Neco Caberta:* Borrador original; Revisión y edición; Administración de proyecto. *Meilynn Jeyli Pineda Gugenbuhul, José Alfredo Ramírez Monares:* Conceptualización; Ideas; Metodología; Análisis formal; Investigación; Visualización; Escritura. *José Francisco Arvizu Astorga:* Investigación, análisis formal, análisis formal, borrador, validación.

## Referencias

- [1] R. S. Rosli, M. H. Habaebi and Md. R. Islam, "Characteristic Analysis of Received Signal Strength Indicator from ESP8266 WiFi Transceiver Module," 7th International Conference on Computer and Communication Engineering (ICCCE), 2018. <https://doi.org/10.1109/ICCCE.2018.8539338>
- [2] X. Villamil, T. Guarda, and G. Ninahualpa Quina, "Agile software development for mobile applications and wireless interaction with hardware development board (Arduino)," 13th Iberian Conference on Information Systems and Technologies (CISTI) in IEEE, pp. 13-16, 2018. <https://doi.org/10.23919/CISTI.2018.8399328>
- [3] A. Aguilar-Hernandez, J. Dias Rodrigues, and A. Tukker, "Macroeconomic, social and environmental impacts of a circular economy up to 2050: A meta-analysis of prospective studies," J. Clean. Prod., vol. 278, no.1,2021.

<https://doi.org/10.1016/j.jclepro.2020.123421>

[4] S. Ding, A. Tukker, and H. Ward, "Opportunities and risks of internet of things (IoT) technologies for circular business models: A literature review," *J. Environ. Manage.*, vol. 336, pp. 117662, 2023.

<https://doi.org/10.1016/j.jenvman.2023.117662>

[5] P. Suresh, J.V. Daniel, V. Parthasarathy, and R.H. Aswathy, "A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment," *Int. Conf. Sci. Eng. Manag. Res.*, pp. 1-8, 2014.

<https://doi.org/10.1109/ICSEMR.2014.7043637>

[6] M.Q. Chau, X.P. Nguyen, T.T. Huynh, V.D. Chu, T.H. Le, T.P. Nguyen, and D.T. Nguyen, "Prospects of application of IoT-based advanced technologies in remanufacturing process towards sustainable development and energy-efficient use," *Energy Sources A: Recovery Util. Environ. Eff.*, 2021.

<https://doi.org/10.1080/15567036.2021.1994057>

[7] S. H. Lee, and K. B. Aspirantes, "Using multimodal educational apps to increase the vocabulary of children with and without reading difficulties," *Int. J. Hum. Comput. Stud.*, vol. 36, pp.100579,2023.

<https://doi.org/10.1016/j.ijcci.2023.100579>

[8] Almaraz-Menéndez, F. Maz-Machado, and A. López, "Tecnología móvil y enseñanza de las matemáticas: una experiencia de aplicación de app Inventor," *Revista Épsilon*, pp. 77-86, 2020.

<https://dialnet.unirioja.es/servlet/articulo?codigo=5589286>

[9] X. Villamil, and 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, 2018.

<http://repositorio.espe.edu.ec/handle/21000/18705>

[10] N. Paucara, "Sistema de Control Domótico Basado en Arduino," *Aplicación Móvil y Voz*, in *Universidad Mayor de San Andrés*, pp.10-13, 2016.

<http://repositorio.umsa.bo/xmlui/handle/123456789/10674>

[11] 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.

[https://upcommons.upc.edu/bitstream/handle/2117/124326/Planos\\_3.pdf?sequence=4&isAllowed=y](https://upcommons.upc.edu/bitstream/handle/2117/124326/Planos_3.pdf?sequence=4&isAllowed=y)

[12] A. Prieto, "Diseño de la app y el control de una cámara video con Arduino," in *Universidad Politécnica de Valencia*, 2016.

<http://hdl.handle.net/10251/76000>

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

<https://doi.org/10.1109/ACCESS.2020.2974911>

[14] B. Pourghebleh, and N.J. Navimipour, "Data aggregation mechanisms in the internet of things: a systematic review of the literature and recommendations for future research," *J. Netw. Comput. Appl.*, vol. 97, pp. 23 - 34,2017.

<https://doi.org/10.1016/j.jnca.2017.08.006>

[15] M. B. Yassein, M.Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, "Internet of Things: Survey and open issues of MQTT protocol", in *2017 International Conference on Engineering MIS (ICEMIS)*, pp 2-5, 2017

<https://doi.org/10.1109/ICEMIS.2017.8273112>

[16] F. D. Rango, G. Potrino, M. Tropea, and P. Fazio, "Energy-aware dynamic Internet of Things security system based on Elliptic Curve Cryptography and Message Queue Telemetry Transport protocol for mitigating Replay attacks," *Pervasive Mob Comput*, vol.16, pp.101105,2020.

<https://doi.org/10.1016/j.pmcj.2019.101105>

[17] I. Chatzigiannakis, A. Vitaletti, A. Pyrgelis, "A privacy-preserving smart parking system using an IoT elliptic curve-based security platform," *Comput. Commun.*, vol.89-90, pp. 165-177,2016.

<https://doi.org/10.1016/j.comcom.2016.03.014>

[18] M.B. Yassein, M.Q. Shatnawi, S. Aljwarneh, R. Al-Hatmi, "Internet of Things: Survey and open issues of MQTT protocol", 2017 International Conference on Engineering & MIS (ICEMIS).

<https://doi.org/10.1109/ICEMIS.2017.8273112>

[19] Arduino, 2013c, Arduino. (2013c).

<http://arduino.cc/es/Tutorial/HomePage>.

[20] F.A. Candelas, G.J. García, S. Puente, J. Pomares, C.A. Jara, J. Pérez, D. Mira and, F. Torres, "Experiences on using Arduino for laboratory experiments of Automatic Control and Robotics" *IFAC-Papers OnLine*, vol.48, no.29, pp. 105-110,2015.

<https://doi.org/10.1016/j.ifacol.2015.11.221>

[21] P. Liu, Q. Xia, K. Liu, J. Guo, X. Wang, J. Liu, J. Grundy, and L. Li, "Towards automated Android app internationalisation: An exploratory study," *J. Stat. Softw.*, vol.197, pp. 111559,2023.

<https://doi.org/10.1016/j.jss.2022.111559>

[22] X. Zhang, and Y. Zhang, "Achieving resource-centric access control for web-app

interactions on android", *High-Confidence Computing*, vol. 2, no 3, pp. 100073,2022.

<https://doi.org/10.1016/j.hcc.2022.100073>

[23] J. Garrido Codo, "TFC Desarrollo de aplicaciones móviles," in open access, 2013

<https://www.lawebdelprogramador.com/pdf/295-5-TFC-Desarrollo-de-Aplicaciones-Moviles.html>

[24] K. Siu-Cheung "Computational Thinking Education", in Harold Abelson Editors, pp. 31-37, 2017.

<https://link.springer.com/book/10.1007/978-981-13-6528-7>

[25] T. Mikolajczyk, H. Fuwen, L. Moldovan, A. Bustillo, M. Matuszewski, and K. Nowicki, "Selection of machining parameters with Android application made using MIT App Inventor bookmarks," *Procedia Manuf.*, vol. 22, pp. 172-179,2018.

<https://doi.org/10.1016/j.promfg.2018.03.027>

[26] Neoteo, "MIT App Inventor: Cómo crear apps para Android sin escribir código", 2020.

<https://www.neoteo.com/mit-app-inventor-como-crear-apps-para-android-sin-escribir-codigo/>

[27] J. Pujol Pérez, "Programamos videojuegos y apps, Crea tu propio videojuego Arkanoid para dispositivos móviles Android con App Inventor," 2017.

<https://programamos.es/crea-tu-propio-videojuego-arkanoid-para-dispositivos-moviles-android-con-appinventor/>

[28] H. Yip, "MIT App Inventor, Connect your Phone or Tablet over WiFi", [https://fog.ccsf.edu/~hyip/mit\\_app\\_inventor/mit\\_app\\_inventor.pptx](https://fog.ccsf.edu/~hyip/mit_app_inventor/mit_app_inventor.pptx)

[29] H. Abelson," Making Mobile Apps with App Inventor", Julio 2017. <http://appinventor.mit.edu/explore/resources/building-mobile-applications-6s063-spring-2013-hal-abelson-mit>

[30] "Website stats for Naylampmechatronics - naylampmechatronics.com," NodeMCU v2 ESP8266 WiFi", 2021. <https://naylampmechatronics.com.usitestat.com/>

[31] Robótica Fácil,"NodeMCUV2ESP8266WiFi", 2019. <https://roboticafacil.es/prod/nodemcu-v2-esp8266-wifi/>

[32] Direct Industry," Products Axor", 2022. <https://www.directindustry.com/prod/axor-industries-29798.html>

[33] R. Herrero, "RTP transport in IoT MQTT topologies", Internet of Things and Cyber-Physical Systems,vol. 3, pp. 37- 44,2023. <https://doi.org/10.1016/j.iotcps.2023.02.001>

[34] A. Shahraki, M. Abbasi, A. Taherkordi, and A. D. Jurcut," A comparative study on online machine learning techniques for network traffic streams analysis," Computer Networks Vol.207, pp.108836,2022. <https://doi.org/10.1016/j.comnet.2022.108836>

[35] A. Shahraki, M. Abbasi, M. Piran, and M. Chen M., Cui S," A comprehensive survey on 6G networks: Applications, core services, enabling technologies, and future challenges," 2021. <https://doi.org/10.48550/arXiv.2101.12475>

Derechos de Autor (c) 2023 Elva Lilia Reynoso Jardón, Manuel de Jesús Nandayapa Alfaro, Quirino Estrada Barbosa, Raúl Neco Caberta, Meilynn Jeyli Pineda Gugenbuhul, José Alfredo Ramírez Monares, José Francisco Arvizu Astorga



Este texto está protegido por una licencia [Creative Commons 4.0](https://creativecommons.org/licenses/by/4.0/).

Usted es libre para compartir —copiar y redistribuir el material en cualquier medio o formato — y adaptar el documento — remezclar, transformar y crear a partir del material— para cualquier propósito, incluso para fines comerciales, siempre que cumpla la condición de:

Atribución: Usted debe dar crédito a la obra original de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace de la obra.

[Resumen de licencia](#) - [Texto completo de la licencia](#)