



Monitoreo y control remoto de un dispensador de alimento para mascotas basado en IoT

Pet feeder monitoring and remote control based on IoT

Sandra Aguilar Alvarez

Escuela Politécnica Nacional, Quito, Ecuador
sandra.aguilar.alvarez@outlook.com

Ricardo Hinojosa Altamirano

Escuela Politécnica Nacional, Quito, Ecuador
ricardo.10marco@gmail.com

Pablo Hidalgo Lascano

Escuela Politécnica Nacional, Quito, Ecuador
pablo.hidalgo@epn.edu.ec

Patricio Cruz Dávalos

Escuela Politécnica Nacional, Quito, Ecuador
patricio.cruz@epn.edu.ec

doi: <https://doi.org/10.36825/RITI.09.17.008>

Recibido: Octubre 21, 2020

Aceptado: Enero 18, 2021

Resumen: La búsqueda de alternativas que faciliten el cuidado y alimentación de una mascota ha creado la necesidad de aplicar en este campo el desarrollo tecnológico actual, ya que los dueños de mascotas ven restringido el tiempo disponible para pasar junto a ellas debido a sus ocupaciones diarias, estudios, trabajo o descuido. En la región, adquirir un dispositivo automático que dispense alimento y agua en un mismo equipo e incluya monitoreo remoto mediante una app en un *smartphone* es difícil y puede resultar costoso. Por tal razón, este trabajo presenta el desarrollo de un dispositivo inteligente basado en MQTT (*Message Queuing Telemetry Transport*) para dispensar alimento y agua para mascotas, capaz de ser monitoreado y controlado a través de un dispositivo móvil. Mediante la integración de múltiples tecnologías y herramientas como: lenguajes de programación, manejo de sensores y actuadores, Raspberry Pi, Arduino, diseño CAD (*Computer-Aided Design*) e impresiones 3D, esta solución IoT (*Internet of Things*) permite implementar un prototipo innovador que contribuye al bienestar de las mascotas no solo en nuestro país sino en toda la región. Las pruebas de funcionalidad validan que el dispositivo desarrollado tiene un tiempo de respuesta corto y permite a los dueños de mascotas mayor independencia y tranquilidad.

Palabras clave: MQTT, Android, IoT, Raspberry Pi, Arduino.

Abstract: The search for alternatives that facilitate the care and feeding of a pet has created the need to apply current technological development in this type of application, since pet owners have restricted time to stay with them due to their daily duties, studies, work or carelessness. In the region, acquiring an automatic device that dispenses food and water in the same equipment and includes remote monitoring through an app on a smartphone is challenging and can be expensive. Therefore, this work presents the development of an automatic device based on MQTT (Message Queuing Telemetry Transport) to dispense food and water for pets, capable of being monitored and controlled through a mobile device. By integrating multiple technologies and tools such as: programming languages, sensor and actuator management, Raspberry Pi, Arduino, CAD (Computer-Aided Design) design and 3D printing, this IoT (Internet of Things) solution allows to implement an innovative prototype that contributes to improve this social problem not only in our country but throughout the region. The functionality tests validate that the developed device has a short response time and that it allows pet owners greater independence and peace of mind.

Keywords: MQTT, Android, IoT, Raspberry Pi, Arduino.

1. Introducción

Las mascotas son parte fundamental de muchos hogares en nuestro entorno y se las considera como parte de la familia; algunos dueños celebran sus cumpleaños, abren cuentas en redes sociales dedicadas a ellas o incluso las envían a guarderías. En el Ecuador, cada vez las mascotas adquieren más derechos y protección mediante leyes [1]. Pero a pesar de que las mascotas tienen un lugar importante en el hogar y en la sociedad, las personas a menudo no tienen el tiempo necesario para cuidarlas, alimentarlas e hidratarlas regularmente, ya sea por motivo de trabajo, viaje o incluso olvido. Por lo cual, el horario de alimentación e hidratación de las mascotas debe adaptarse al tiempo y criterio del dueño, permitiendo inclusive pasar toda una jornada sin los cuidados necesarios. Debido a esto, las personas buscan alternativas tecnológicas que faciliten el cuidado de sus mascotas, creándose así una demanda de soluciones innovadoras.

En páginas de *e-commerce* como Amazon existe una gran variedad de dispensadores de alimento o agua para mascotas, tales como dispensadores mecánicos, dispensadores automáticos no controlados remotamente, entre otros [2]. En general estos dispensadores, a pesar de brindar ayuda, tienen características limitadas, ya que se los debe programar localmente, no tienen una forma de administración remota a través de Internet, o son soluciones que proporcionan alimento en porciones fijas, sin considerar los requerimientos de cada mascota.

Es por tal razón, que el propósito del presente trabajo es presentar el desarrollo de una solución basada en IoT, que brinde una ayuda en la alimentación e hidratación de mascotas y principalmente pueda ser de gran ayuda a sus dueños. El desarrollo del proyecto ha permitido fusionar varias disciplinas, conceptos y herramientas en un mismo dispositivo, como lo son la automatización, la transmisión y gestión de datos a través de Internet, la integración de diferentes sistemas embebidos y el prototipado mediante impresión 3D.

Por medio del desarrollo de este dispositivo se ha podido dar soluciones a algunas limitaciones de los dispensadores comerciales. El equipo planteado es un dispositivo automático, administrable mediante una aplicación Android, y que puede ser monitoreado y controlado a través de Internet. Permite personalizar el tamaño y frecuencia de las porciones de alimento. Además, cuenta con dos reservorios, uno para almacenar alimento y otro para agua; cuenta con los respectivos sensores y actuadores, una tarjeta Arduino, una tarjeta Raspberry Pi e incluye una cámara de video. La app en Android se comunica mediante el protocolo MQTT con la Raspberry Pi del dispensador, lo que facilita el monitoreo de los niveles de alimento y agua en los reservorios, así como notificaciones de actividad y alertas de niveles bajos o atasco de alimento. Por esto, el prototipo implementado es un equipo completo y ofrece la versatilidad a los dueños de mascotas, de que por más lejos que se encuentren, puedan dar una alimentación e hidratación adecuada a su mascota con la ayuda de su dispositivo móvil.

2. Estado del arte

A continuación, se describen las principales características de algunos dispensadores para mascotas que se pueden encontrar comercialmente y algunos que se basan en Raspberry Pi y/o Arduino.

Existen comercialmente en páginas de *e-commerce* una gran cantidad de dispositivos con características y funcionalidades muy similares, algunos bastante completos e interesantes, como lo son el SmartFeeder de Petnet

[3] o el alimentador automático para mascotas de Feed'em [4]. El SmartFeeder de Petnet es un dispensador automático de alimento que puede ser controlado y monitoreado por una aplicación móvil; permite porciones de alimento personalizadas y envío de notificaciones cuando las mascotas han sido alimentadas. Por otra parte, se encuentra el alimentador automático de Feed'em que permite programar y personalizar las porciones a ser administradas; con un control mediante una app, monitoreo mediante una cámara e interacción de voz.

Respecto de las soluciones basadas en Raspberry Pi y Arduino, Vania, Karyono y Hargyo [5] diseñaron un dispensador de alimento con Arduino Uno basado en el protocolo MQTT, con identificación de mascotas empleando etiquetas RFID (*radio-frequency identification*) que permite autenticar al perro que tiene acceso al dispositivo. Permite programar las porciones a través de una aplicación Android y está compuesto de una celda de carga que permite medir las porciones que el usuario configura.

El Smart Pet Care System de Seungcheon Kim [6] es un dispensador automático que cuenta con un Raspberry Pi, un Arduino, una cámara, una aplicación para Android y una almohadilla para excrementos. Permite dispensar alimento de manera remota, cuenta con un sensor de peso y *timer* para porciones personalizadas.

Luego de haber observado algunas de las alternativas existentes, el dispensador desarrollado y que se presenta en este documento complementa las características anteriormente descritas. Los cuatro equipos descritos, integran en sus capacidades únicamente la dispensación de alimento, lo cual, significa que el usuario debe adquirir otro equipo o buscar algún mecanismo adicional para dotar de agua fresca a su mascota. En cambio, el proyecto desarrollado se enfoca en el diseño y construcción de un dispensador de agua y alimento para mascotas, controlado remotamente por una app en Android a través de Internet mediante MQTT, y que adicionalmente cuenta con respaldo de energía. El usuario por medio de una app puede enviar comandos al dispensador para alimentar a su mascota, además tiene la opción de visualizarla mediante una cámara. MQTT permite que se cuente también con el monitoreo de los niveles de agua y alimento, y la recepción tanto de alertas de niveles críticos en los reservorios como notificaciones de la dispensación.

En los desarrollos tecnológicos de los últimos años, es evidente que MQTT es el protocolo convencional para IoT [7], ha permitido la implementación de varios sistemas y dispositivos inteligentes como fitotrones [8], detectores de terremotos [9], monitores de signos vitales [10], entre otros. Y es debido al paradigma de comunicación de MQTT que el proyecto desarrollado puede ser escalado para soportar múltiples usuarios y dispensadores sin cambios en su diseño, pudiendo ser llevado a un ámbito comercial siendo la única preocupación la fabricación mecánica del dispositivo.

3. Metodología

En la Figura 1 se presenta un esquema general del sistema implementado [11]. Está compuesto por tres partes: el Sistema de Comunicación, el Sistema de Control y el Sistema de Alimentación Eléctrica y Respaldo de Energía.

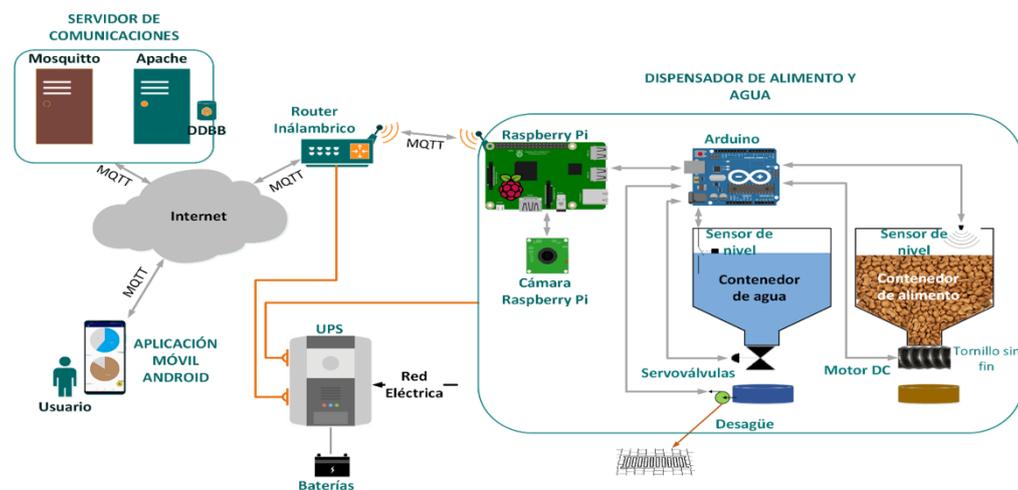


Figura 1. Esquema general del sistema implementado.

El Sistema de Control se basa en un Arduino Mega 2560 que es el dispositivo controlador que administra todos los sensores y ejecuta las acciones de dispensación de alimento y agua a través de distintos actuadores. Todos los datos de nivel de agua y alimento en los reservorios, así como las notificaciones de actividades realizadas y las alertas son enviadas por comunicación serial a la tarjeta Raspberry Pi.

El Sistema de Comunicación está conformado por tres componentes: una aplicación móvil en Android, un servidor de comunicaciones y una tarjeta Raspberry Pi 3B con su cámara. Su diseño está basado en IoT, para lo cual, se emplea el protocolo MQTT, el mismo que trabaja bajo el modelo de comunicación publicador/suscriptor [12]. En donde los nodos, en este caso la Raspberry Pi, se comunican con la app mediante *topics* o colas de mensajes a través de un Broker MQTT. Esto permite al usuario el envío de comandos a su dispensador y a la vez la recepción de datos y notificaciones enviados por el sistema de control.

A través de estos sistemas se ha implementado la arquitectura IoT de la Figura 2, en la cual, los datos generados en el dispensador (niveles de alimento y agua, notificaciones de actividad y alertas) son transportados por el protocolo MQTT a través de Internet, para luego ser procesados y presentados como información al usuario por medio de la app en Android. De la misma manera, los datos de comandos accionados por medio de la app son enviados al dispensador.



Figura 2. Arquitectura IoT del proyecto.

Adicionalmente, el Sistema de Alimentación Eléctrica y Respaldo de Energía proporciona alta disponibilidad de la energía necesaria para el funcionamiento de todos los dispositivos electrónicos que conforman el dispensador. Para ello cuenta con una UPS (*Uninterruptible Power System*) y baterías que brindan un respaldo de energía de aproximadamente 200 minutos en caso de ausencia de energía eléctrica comercial.

3.1. Sistema de Comunicación

El Sistema de Comunicación es un sistema de software distribuido, el cual está definido por un conjunto de funcionalidades de software organizadas en tres capas: la capa de Presentación, la capa de Lógica de Negocio y la capa de Datos como se indica en la Figura 3. Estas capas lógicas del sistema software se encuentran distribuidas en los tres sistemas de cómputo del sistema: la Aplicación Móvil en Android, el Servidor de Comunicaciones y la Raspberry Pi. El desarrollo de este sistema se basó en la metodología ágil Kanban [13].

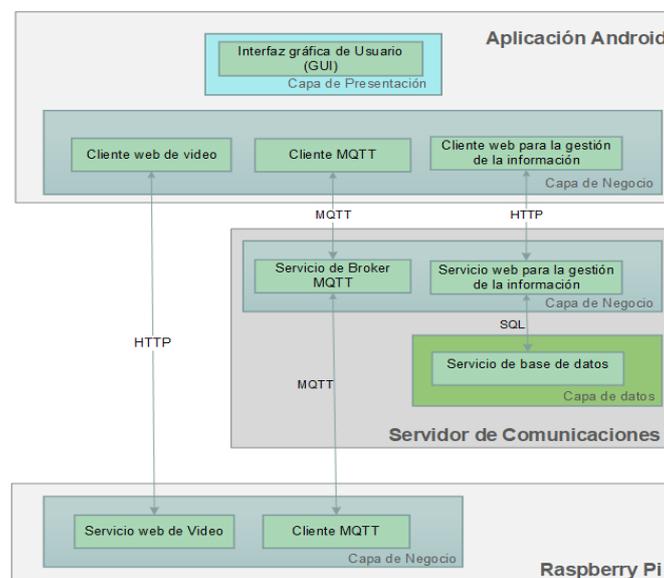


Figura 3. Arquitectura del software del Sistema de Comunicación.

La app permite al usuario enlazar su dispensador mediante un código QR, registrar una cuenta e ingresar los datos de nombre, raza, peso y tamaño de la mascota. Con base en esta información, se le sugiere al dueño de la mascota una ración diaria adecuada, conforme a las sugerencias de Royal Canin [14], o se permite crear un perfil personalizado de alimentación; es decir, puede ingresar la cantidad de gramos a dispensar, así como el número de porciones al día. También la app permite visualizar los niveles de alimento y agua en los reservorios mediante diagramas circulares dinámicos, activar el recambio de agua fresca, recibir notificaciones, alertas y visualizar el video de la mascota. Por esto la app cuenta con:

- Una interfaz gráfica de usuario diseñada en Balsamiq mediante *wireframes*.
- Un módulo de software para la comunicación MQTT con la Raspberry Pi aplicando el servicio Paho para Android [15], lo que permite al usuario el envío de comandos al dispensador y a la vez la recepción de datos enviados por el sistema de control.
- Un cliente web mediante WebView de Android para la visualización del video que es proporcionado por el servicio web que se ejecuta en la Raspberry Pi.
- Un cliente web adicional, mediante el uso de Volley [16], para la interacción con la base de datos del servidor de comunicaciones, que permite al usuario registrar una cuenta, autenticarse y enlazar el dispensador mediante el escaneo de un código QR empleando la librería ZXing [17].

El servidor de comunicaciones provee el servicio de Broker MQTT mediante Mosquitto [18] y un Servicio Web mediante Apache; además, alberga un motor de base de datos MySQL para almacenar información de cuentas de usuarios y dispensadores. La app se comunicará con el Servicio Web para registrar la cuenta del usuario en la base de datos y enlazar con su respectivo dispensador. Una vez que se haya enlazado, la app consultará al Servicio Web los *topics* MQTT asociados al dispensador, los que servirán para una comunicación directa con el dispensador mediante el Broker MQTT.

La tarjeta Raspberry Pi es la que permite al dispensador la comunicación con la app a través de Internet; además, permite la visualización del video mediante su cámara. Es el punto de conexión con el Sistema de Control mediante comunicación serial; recibe las configuraciones del modo de dispensación y ejecuta una rutina de temporización que permite indicar al Sistema de Control que dispense la cantidad de gramos configurada en el momento correcto. El programa que se ejecuta en la Raspberry Pi incluye:

- Un módulo de software para la comunicación MQTT con la app mediante la librería Eclipse Paho MQTT para Python [19]. Permite recibir comandos como las configuraciones de dispensación y recambio de agua, además envía los niveles de alimento y agua, notificaciones de actividad y alertas del Sistema de Control.
- Un servicio web mediante *MJPEG-streamer* [20] para transmitir el video que captura la cámara de la Raspberry Pi.

Los *topics* que se emplean en la comunicación MQTT se resumen en la Tabla 1.

Tabla 1. Resumen de *topics* MQTT.

Topic MQTT	Uso	Suscriptor	Publicador
idDispensador/comida	Para indicar el modo de dispensación y la ración diaria de alimento	Raspberry Pi	App
idDispensador/niveles	Para enviar los niveles de alimento y agua en los reservorios	App	Raspberry Pi
idDispensador/notificaciones	Notifica la dispensación de alimento y agua, y cambio de agua fresca	App	Raspberry Pi
idDispensador/alertas	Alerta sobre atascos o niveles bajos de alimento y agua	App	Raspberry Pi
idDispensador/camara	Para activar o desactivar la cámara	Raspberry Pi	App
idDispensador/desague	Para activar el cambio de agua del plato de la mascota	Raspberry Pi	App

Fuente: Elaboración propia.

3.2. Sistema de Control

Este sistema es el encargado de ejecutar las acciones de dispensación de alimento y agua, así como de enviar datos de niveles, actividad y alertas. Está conformado por un Arduino Mega 2560, sensores de nivel y peso, y actuadores, como servoválvulas y un motor DC. Los drivers se seleccionaron para que el Arduino pueda administrar los sensores y controlar los actuadores del prototipo.

Entre los sensores y actuadores se tienen: un sensor de peso con una celda de carga de 5 Kg, sensores de nivel continuo (ultrasónico) y discreto (infrarrojo), un sensor de corriente de efecto Hall, un motor DC, servomotores y válvulas. El sensor de corriente cumple la función de proteger el motor en caso de que se produzca un atasco durante la dispensación; además, el motor será controlado a través de un driver L293D.

La estructura del dispensador está constituida por dos reservorios (ver Figura 4), uno para agua y el otro para el alimento seco. Estos reservorios y demás piezas fueron diseñadas empleando Autodesk Inventor y posteriormente implementadas en acrílico (reservorios) e impresiones 3D. El mecanismo dispensador de agua está conformado por sensores de nivel en el reservorio y en el plato de la mascota, y tres servoválvulas. La primera para el ingreso automático de agua al reservorio, la segunda para dispensar agua al plato y la tercera para evacuar el agua que la mascota no ha consumido por algún tiempo. El mecanismo dispensador de alimento se basa en un tornillo sin fin; este tornillo es accionado por un motor DC que, en conjunto con un sensor de peso permiten dosificar el alimento de manera adecuada. El nivel de alimento en el reservorio es determinado a través de cinco sensores infrarrojos de presencia dispuestos en la respectiva tolva.

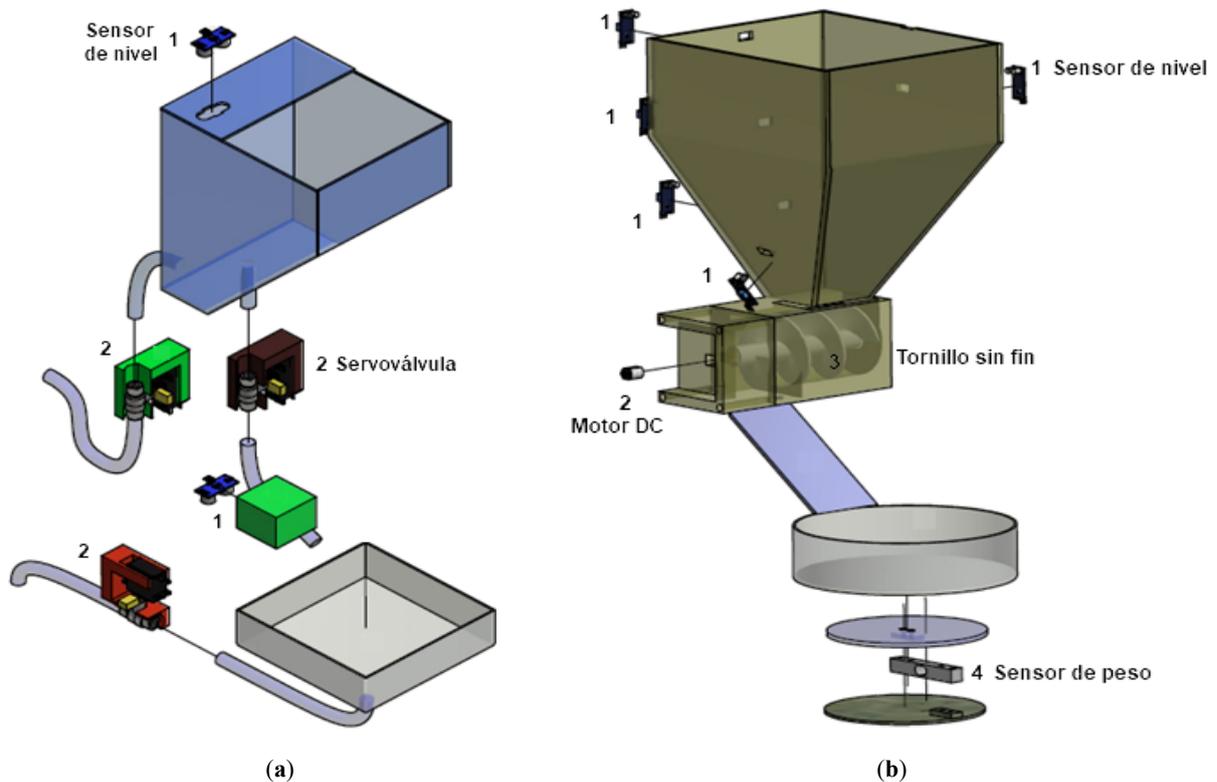


Figura 4. Estructura del prototipo: (a) Mecanismo dispensador de agua; (b) Mecanismo dispensador de alimento.

El Arduino ejecuta un programa de control, cuyo proceso de automatización funciona de acuerdo con el diagrama de flujo de la Figura 5; este programa se encarga de la coordinación de los elementos de control y de la adquisición de datos, así como el procesamiento e interpretación tanto de datos como de comandos que envía la aplicación móvil, para su posterior ejecución en el dispensador.

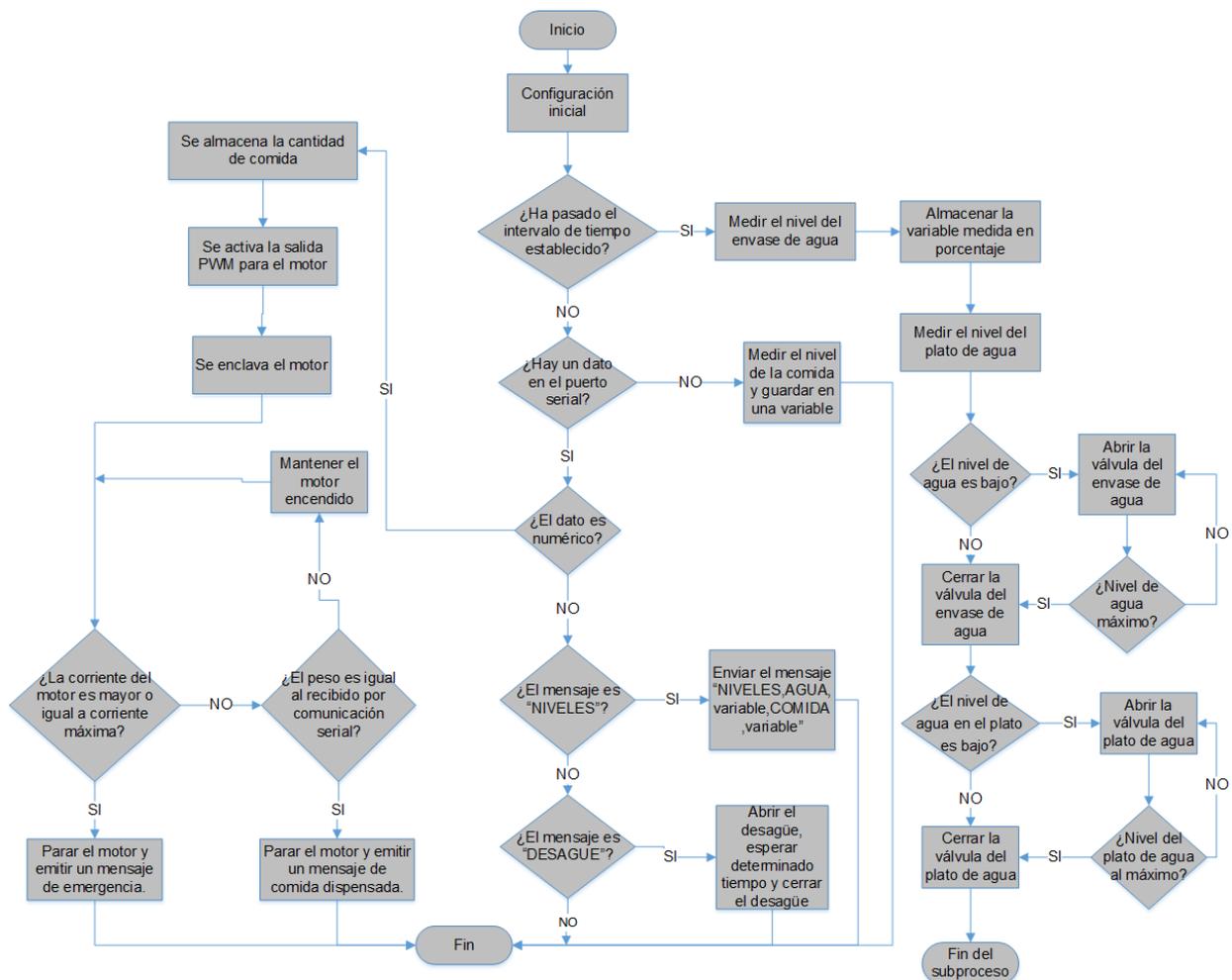


Figura 5. Diagrama de flujo del programa de control.

3.3. Sistema de Alimentación Eléctrica y Respaldo de Energía

Este sistema se diseñó con el objetivo de proporcionar la energía necesaria a todos los dispositivos de control que trabajan con voltajes de 12 y 5 voltios DC. Todos estos elementos se encuentran alimentados por una fuente de poder para garantizar su correcta polarización y protección a los pines del sistema de control.

El prototipo cuenta con respaldo de energía de tal modo que pueda seguir operando en caso de ausencia de energía eléctrica. Con esto, se busca que el dispensador se mantenga en funcionamiento hasta que se recupere la energía eléctrica. Se realizaron mediciones del prototipo y en funcionamiento normal el consumo es de 24 watts. Para el prototipo desarrollado se empleó el UPS APC Power Saving Back-UPS Pro 1000, que tiene una potencia de 1000 VA, modo de transferencia on-line y cuenta con dos baterías internas de 24 V. Este UPS, para el consumo del prototipo, brinda una independencia aproximada de 206 minutos.

4. Resultados

Se realizaron pruebas de dispensación de alimento con las que se buscó evaluar la comunicación y la automatización electrónica del dispensador. Para lo cual, se configuró en la app características de perros de raza pequeña, mediana y grande. Para cada tamaño de perro se realizaron cinco pruebas de dispensación de una, dos y tres porciones diarias. En cada una se verificaron los siguientes parámetros:

- Cantidad de gramos dispensados mediante una balanza comercial.
- Cumplimiento de la hora programada.
- Recepción de la notificación de dispensación en la app.
- Atasco de alimento.

Con respecto a la cantidad de gramos dispensados, en la Figura 6 se puede observar una gráfica de dispersión, a la que se aplicó una línea de tendencia lineal para poder determinar el comportamiento del error. Con esta línea de tendencia se obtuvo que R-cuadrado es igual a 0.43, es decir, que los datos de errores obtenidos son muy dispersos y se deberían tomar un mayor número de muestras. Aunque, se puede establecer que el prototipo presenta un margen de error aceptable, menor al 5%, cuando dispensa raciones a partir de los 200 gramos, lo cual, hace que el dispensador sea adecuado para mascotas de tamaño mediano y grande.

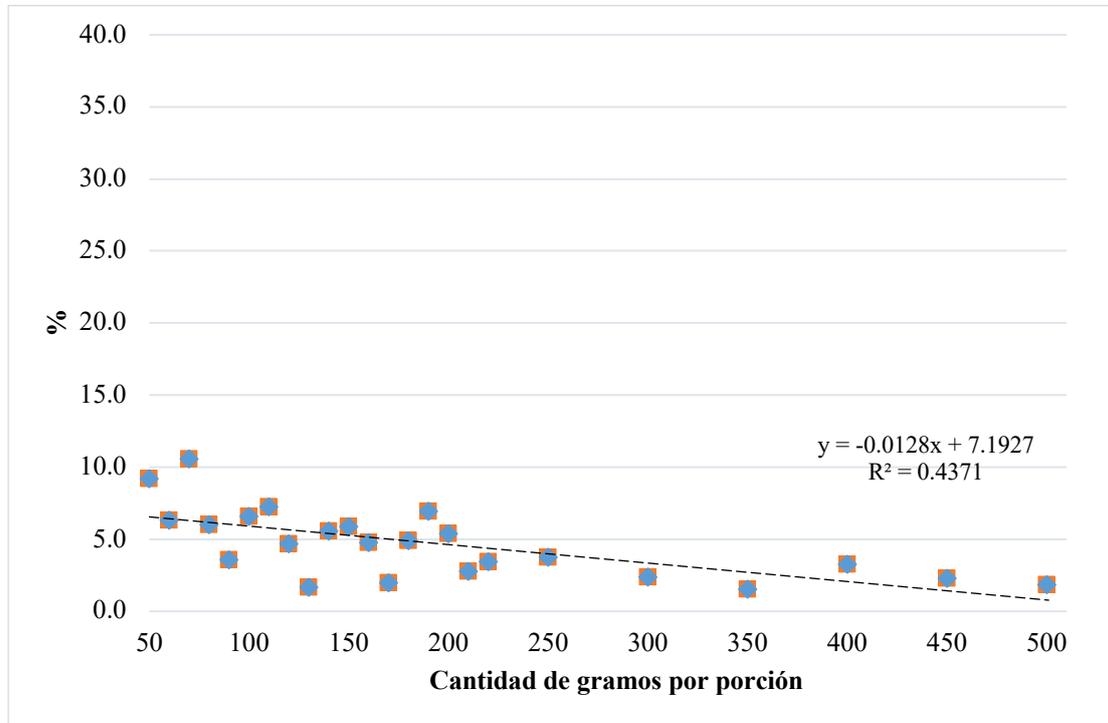


Figura 6. Porcentaje de error en la dispensación de alimento.

Respecto al cumplimiento de la hora programada de dispensación, como se indica en la Figura 7, en el 100% de las pruebas, se dispensó alimento en el horario indicado. De igual forma, todas las notificaciones de dispensación fueron recibidas en la app a través de Internet mediante MQTT de manera instantánea. Y en cuanto al último parámetro, el dispensador se atascó un 7% de las pruebas como se observa en la Figura 8, pero cabe recalcar que este porcentaje se lo obtiene después de realizar alrededor de 300 pruebas.

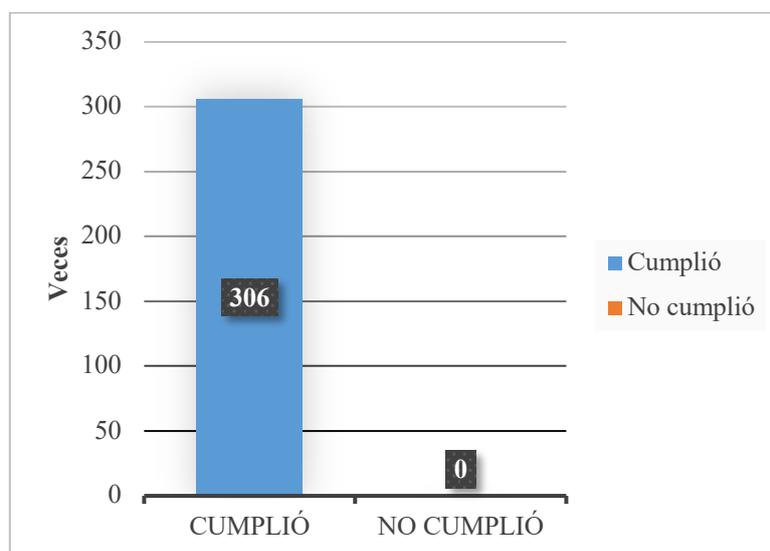


Figura 7. Porciones dispensadas en el horario indicado.

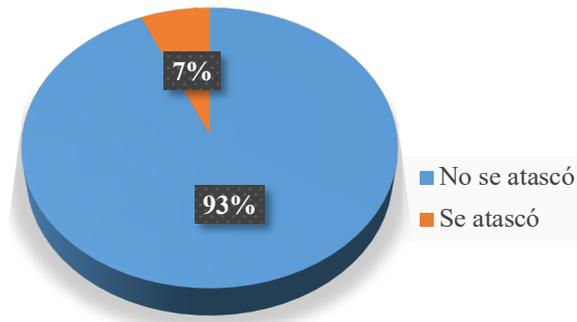


Figura 8. Porcentaje de atascamiento de alimento.

Simultáneamente, con la verificación de dispensación de agua, se realizaron pruebas de recambio de agua del plato de la mascota. Se realizan 20 pruebas y en el 100% de los casos, el dispensador cumplió con el desagüe del agua, el relleno del plato y se recibieron instantáneamente las notificaciones de recambio y dispensación de agua, respectivamente.

Luego, por medio de la app se verificó que el usuario puede visualizar el video que transmite el dispensador. En una primera prueba, en la cual, el *smartphone* se conectó a Internet mediante un acceso ADSL (*Asymmetric Digital Subscriber Line*) de 5 Mbps existieron retardos significativos en la recepción de video sin disminución de la calidad. Estos retardos alcanzaron los 30 y 40 segundos. Se procedió a cambiar el acceso a Internet y se empleó la red celular. En este segundo caso, se observó un retardo máximo de 5 segundos en las horas de la mañana y de hasta 10 segundos en la tarde.

Estos retardos pueden estar ocasionados porque el tráfico de datos en las redes de comunicación es mayor en las horas de la tarde y noche. Pero, de manera general, la visualización del video fue exitosa; el retardo no influye en la comprensión de lo que sucede en el dispensador, las imágenes son claras y se ha podido observar la dispensación de alimento y como el agua fluye cuando se dispensa o desagua. En la Figura 9 se indica una captura de pantalla de una imagen transmitida desde el servidor MJPG del dispensador.



Figura 9. Prueba de visualización del video en la app.

Para verificar el respaldo de energía, se conectó el prototipo y el *router* inalámbrico que le provee acceso a Internet al UPS y una vez que las baterías internas se cargaron al 100% se procedió a desconectar la red pública de energía eléctrica. Se comprobó que el UPS brindó una autonomía de 3 horas y 30 minutos aproximadamente y que el dispensador realizó todas sus actividades programadas en ese tiempo. Finalmente, se realizaron pruebas en un ambiente real por el lapso de 7 días, con un perro de tamaño mediano. En la aplicación móvil se configuró una ración diaria de 300 gramos repartidos en tres porciones al día en el horario de 7:00, 13:00 y 19:00.

Se observó que la mascota tuvo agua fresca a lo largo del día, pudo tener alimento en las horas adecuadas, comió sus porciones y no tuvo que esperar toda una jornada para ser alimentado nuevamente. Adicionalmente, el dueño de la mascota tuvo mayor independencia y tranquilidad ya que por más lejos que se encontró o por el limitado tiempo que tenía, su mascota estaba siendo alimentada e hidratada mediante el dispensador.

En la Figura 10 se puede visualizar al dispensador terminado y completamente funcional. El reservorio de alimento puede almacenar hasta 3 kilogramos de alimento seco y el reservorio de agua tiene una capacidad de 7 litros máximo, suficiente para alimentar por una semana a una mascota que requiera una ración diaria de 300 gramos aproximadamente.



Figura 10. Dispensador implementado. (a) Vista frontal; (b) Vista posterior.

5. Conclusiones

El dispensador desarrollado es un equipo fácil de configurar y controlar mediante la aplicación móvil, ha permitido al usuario una mayor independencia en el cuidado de su mascota. Además, se ha comprobado que IoT puede ser aplicado en áreas comunes de un hogar permitiendo mejorar la calidad de vida de las mascotas y sus dueños mediante la conexión de dispositivos inteligentes.

Es desafiante integrar varios conceptos y disciplinas en un mismo sistema; sin embargo, esto se ha cumplido exitosamente en la implementación del prototipo descrito en este trabajo. Se ha realizado el diseño y desarrollo de un sistema software distribuido basado en IoT; se automatizó el proceso de dispensación de alimento y agua; se realizaron diseños de estructuras y piezas mediante un software CAD y se implementaron mediante cortes láser de acrílico e impresiones 3D en PLA (*Polylactic Acid*).

Por medio de este proyecto, se han mostrado soluciones a algunos problemas de los dispensadores comerciales. El equipo puede ser controlado remotamente por el usuario mediante una app en Android y en el mismo equipo se cuenta tanto con dispensación de alimento como de agua. Adicionalmente, las porciones van acorde al tamaño, peso y raza de las mascotas. En base a esto, el equipo pesa la cantidad de gramos necesarios al dispensar y permite mantener el número de raciones recomendadas o preestablecidas por el usuario. También, se recambia el agua del plato de la mascota para proporcionarle agua siempre fresca. El usuario por medio de la aplicación móvil puede visualizar los niveles de alimento y agua en los reservorios, de la misma manera puede observar por medio de una cámara si su mascota toma agua o come su porción. También cuenta con un sistema de respaldo de energía para que siga funcionando cuando haya ausencia de energía eléctrica.

6. Agradecimientos

Agradecemos a la PhD. Soraya Sinche por sus valiosos comentarios y sugerencias respecto a la realización y presentación de este trabajo. Expresamos nuestro reconocimiento al Departamento de Electrónica, Telecomunicaciones y Redes de Información (DETRI) y al Departamento de Automatización y Control Industrial (DACI) de la Escuela Politécnica Nacional por las facilidades prestadas para el desarrollo de este proyecto.

7. Referencias

- [1] Protección Animal Ecuador. (2020). *Legislación vigente en Ecuador*. Recuperado de: <https://www.pae.ec/derecho-animal/legislacion-vigente/>
- [2] Amazon. (2020). *Dispensadores automáticos para mascotas*. Recuperado de: https://www.amazon.com/s?k=dispensadores+automaticos+para+mascotas&__mk_es_US=%C3%85M%C3%85%C5%BD%C3%95%C3%91&ref=nb_sb_noss
- [3] PetNet SmartFeeder. (2018). *Automatic pet feeder for cats and dogs*. Recuperado de: <https://www.petnet.io/smartfeeder>
- [4] Feed'em. (2020). *Smart Pet Feeder with HD Camera*. Recuperado de: <https://smartpetfeeder.ca/product/automatic-cat-feeder/>
- [5] Vania, Karyono, K., Nugroho, H. (2016). Smart dog feeder design using wireless communication, MQTT and Android client. Trabajo presentado en *International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, Tangerang, Indonesia. doi: <https://doi.org/10.1109/IC3INA.2016.7863048>
- [6] Kim, S. (2016). Smart Pet Care System using Internet of Things. *International Journal of Smart Home*, 10, 211-218. doi: <https://doi.org/10.14257/ijsh.2016.10.3.21>
- [7] Shi., H., Niu, L., Sun, J. (2020). Construction of Industrial Internet of Things Based on MQTT and OPC UA Protocols. Trabajo presentado en *IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Dalian, China. doi: <https://doi.org/10.1109/ICAICA50127.2020.9182598>
- [8] Ait Abdelouahid, R., Debauche, O., Mahmoudi, S., Marzak, A., Manneback, P., Lebeau, F. (2020). *Open Phytotron: A New IoT Device for Home Gardening*. Recuperado de: <http://hdl.handle.net/2268/248995>
- [9] Boccadoro, P., Montaruli, B., Grieco, L. A. (2019). QuakeSense, a LoRa-compliant Earthquake Monitoring Open System. Trabajo presentado en *IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, Cosenza, Italy. doi: <https://doi.org/10.1109/DS-RT47707.2019.8958675>
- [10] Sarierao, B. S., Prakasarao, A. (2018). Smart Healthcare Monitoring System Using MQTT Protocol. Trabajo presentado en *3rd International Conference for Convergence in Technology (I2CT)*, Pune, India. doi: <https://doi.org/10.1109/I2CT.2018.8529764>
- [11] Aguilar Alvarez, S. N., Hinojosa Altamirano, R. M. (2019). *Desarrollo de un prototipo para dispensar alimento y agua para mascotas con tecnología IoT monitoreado y controlado a través de un dispositivo móvil* (Tesis de Grado). Escuela Politécnica Nacional, Ecuador. Recuperado de: <https://bibdigital.epn.edu.ec/handle/15000/20590>
- [12] mqtt.org. (2019). *MQTT version 5.0*. Recuperado de: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf>

- [13] Nakazawa, S., Tanaka, T. (2016). Development and Application of Kanban Tool Visualizing the Work in Progress. Trabajo presentado en *5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, Kumamoto, Japan. doi: <https://doi.org/10.1109/IIAI-AAI.2016.156>
- [14] Royal Canin. (2020). *Nutrición saludable para perros*. Recuperado de: <https://www.royalcanin.com/co/dogs/products>
- [15] Eclipse. (2020). *Eclipse Paho Android Service*. Recuperado de: <https://github.com/eclipse/paho.mqtt.android>
- [16] Android Developers. (2020). *Descripción general de Volley*. Recuperado de: <https://developer.android.com/training/volley>
- [17] ZXing. (2020). *ZXing ("Zebra Crossing") barcode scanning library for Java, Android*. Recuperado de: <https://github.com/zxing/zxing>
- [18] Eclipse. (2020). *Eclipse Mosquitto*. Recuperado de: <https://mosquitto.org/>
- [19] Eclipse. (2020). *Eclipse Paho MQTT Python client library*. Recuperado de: <https://pypi.org/project/paho-mqtt/>
- [20] jacksonliam. (2017). *mjpg-streamer*. Recuperado de: <https://github.com/jacksonliam/mjpg-streamer>