



INSTITUTO TECNOLÓGICO SUPERIOR DE MISANTLA

“DESARROLLO DE PROTOTIPO DISPENSADOR DE ALIMENTO PARA MASCOTAS PET-HOME ALBA BASADO EN EL IOT”

TESIS

QUE PARA OBTENER EL TÍTULO DE

**INGENIERO EN TECNOLOGÍAS DE LA
INFORMACIÓN Y COMUNICACIONES**

P R E S E N T A:

ARELY MONTSERRAT GARCÍA MÉNDEZ

ASESOR

LIC. DANIELA LIZBETH HERNÁNDEZ SÁNCHEZ

Co-ASESORES

MSC. ARNULFO GAMALIEL HERNÁNDEZ GONZÁLEZ
ING. FRANCISCO JAVIER FERNÁNDEZ DOMÍNGUEZ

MISANTLA, VERACRUZ

MARZO, 2020



INSTITUTO TECNOLÓGICO SUPERIOR DE MISANTLA
DIVISIÓN DE ESTUDIOS PROFESIONALES
AUTORIZACIÓN DE IMPRESIÓN DE TRABAJO DE TITULACIÓN

FECHA: 03 de Marzo de 2020

ASUNTO: AUTORIZACIÓN DE IMPRESIÓN
DE TESIS PROFESIONAL.

A QUIEN CORRESPONDA:

Por medio de la presente hago constar que el (la) C:

ARELY MONTSERRAT GARCÍA MÉNDEZ

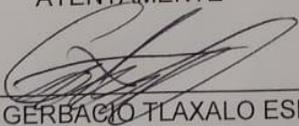
pasante de la carrera de INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES con No. de Control 152T0079 ha cumplido satisfactoriamente con lo estipulado por el Manual de Procedimientos para la Obtención del Título Profesional de Licenciatura bajo la opción Titulación Integral (Tesis Profesional)

Por tal motivo se Autoriza la impresión del Tema titulado:

“DESARROLLO DE PROTOTIPO DISPENSADOR DE ALIMENTO PARA MASCOTAS PET-HOME ALBA BASADO EN EL IOT”

Dándose un plazo no mayor de un mes de la expedición de la presente a la solicitud del Acto de Recepción para la obtención del Título Profesional.

ATENTAMENTE


ING. GERBACIO TLAXALO ESPINOZA
DIVISIÓN DE ESTUDIOS PROFESIONALES



Archivo.

*“Si nunca lo has intentado, ¿Cómo vas a saber si
existe alguna posibilidad?”.*

Jack Ma

Este proyecto está dedicado a...

Principalmente a Dios por brindarme las habilidades personales y profesionales para cumplir una meta más en la vida.

A mi familia por ser el pilar fundamental, que con paciencia y mucho amor supieron guiarme.

A mis amigos que compartieron este objetivo conmigo y juntos pudimos alcanzar esta meta.

A mis maestros en estos años de constantes sacrificios y valiosas enseñanzas que fueron los responsables de mi formación y crecimiento como profesionista.

Agradecimientos

A mi abuelita, mi madre Esmeralda, mi hermano Bryan, mis tíos Héctor, Jorge y Sergio por estar siempre presentes, por confiar, por los consejos, por los valores y principios que me han inculcado, por sobre todo el amor brindado cada día. Los amo y estoy eternamente agradecida.

A mi novio Jesús por todo su amor y apoyo incondicional durante este tiempo juntos, gracias por impulsarme a terminar este proyecto, la ayuda que me has brindado ha sido sumamente importante. Muchas gracias, te quiero tanto.

A mis amigos sobre todo Cinthia, Dalia y David que son mis personas especiales a las que me gustaría agradecer su amistad quienes siempre estuvieron conmigo para brindarme su apoyo, quiero darles las gracias por formar parte de mí ¡Los quiero!

Igualmente me siento agradecida con todos mis maestros durante toda la carrera sobre todo a mi asesora Lic. Daniela y mis co-asesores Mtro. Gamaliel y al Ing. Francisco, a quienes les debo mi gratitud en la elaboración de este trabajo.

A la empresa ALBA-DTI, por darme todas las facilidades necesarias para completar este proyecto de residencias y compartir sus enseñanzas conmigo.

Resumen

El trabajo que se presenta a continuación se realizó en la empresa ALBA Desarrollo Tecnológico e Innovación (ALBA-DTI), la cual en busca de explorar en el mundo del Internet de las cosas (IoT) decide el desarrollo de un prototipo dispensador de alimento dosificable, es decir, suministrar la cantidad de alimento de acuerdo a la información de las mascotas que va dirigido especialmente para perros y gatos. El funcionamiento de este prototipo consiste en un nodo central siendo el servidor, un nodo lector y una aplicación móvil para su control remoto. Para el servidor se ha instalado Mosquitto como Broker MQTT al ser Open-source beneficia en la disminución de los costos y es adecuado para el uso de producción; permite la comunicación por el protocolo Message Queue Telemetry Transport (MQTT) con la aplicación móvil y el nodo lector, siendo que este Broker se encuentra alojado en una Raspberry Pi que trabaja en un entorno Ubuntu. El nodo lector consiste en un módulo NodeMcu ESP8266 al que se conecta un circuito electrónico integrado por sensores que obtienen el nivel de alimento y actuadores que ejecutan la tarea de dispensar el alimento. Por último, la aplicación móvil se realiza para controlar de forma remota en tiempo real o automática el prototipo, de asimismo monitorear el nivel de alimento en el contenedor del dispensador; la información ingresada de la mascota y los horarios para el dispenso del alimento por la aplicación se almacena en la base de datos que se encuentra en el servidor, cabe mencionar que al hacer uso de Flutter permite que el diseño de la aplicación sea atractiva e intuitiva para el usuario.

Tabla de contenido

Introducción	1
Capítulo I. Generalidades del proyecto	2
1.1. Planteamiento del problema	2
1.2. Antecedentes	3
1.2.1. Dispensadores comunes	3
1.2.2. Dispensadores automáticos o inteligentes	4
1.2.2.1. Dispensador Petnet	4
1.2.2.2. Dispensador Bistro	4
1.2.2.3. Dispensador Fresh Element	5
1.2.2.4. Dispensador CleverPet	5
1.2.2.5. Dispensador Catspad	6
1.2.2.6. Dispensador Petsafe	7
1.2.2.7. Dispensador Petmate	7
1.3. Justificación.....	8
1.4. Objetivos	10
1.4.1. Objetivo general.....	10
1.4.2. Objetivo específicos.....	10
1.5. Hipótesis.....	10
Es posible construir un dispensador remoto de alimento para mascotas basado en el Internet de las cosas para mejorar la alimentación de las mascotas a partir de herramientas de bajo costo.....	10
1.6. Alcances y limitaciones	12
1.6.1. Alcances	12
1.6.2. Limitaciones	12
1.7. Propuesta de solución	13
1.8. Metodología	14
Capítulo II. Análisis de los fundamentos.....	16
2.1. Marco histórico.....	16
2.1.1. El internet de las cosas	16
2.1.2. El dispensador.....	17
2.2. Marco teórico	18
2.2.1. Base de datos	18

2.2.2. Sistema gestor de base de datos.....	18
2.2.2.1. Oracle	18
2.2.2.2. PostgreSQL.....	18
2.2.2.3. MySQL	19
2.2.3. Servidor web.....	19
2.2.3.1. Wamp	19
2.2.3.2. Laragon.....	19
2.2.3.3. Lamp.....	20
2.2.4. Lenguaje de PHP	20
2.2.5. Protocolos para el internet de las cosas.....	21
2.2.5.1. CoAP	21
2.2.5.2. Rest API.....	21
2.2.5.3. MQTT.....	22
2.2.6. Broker MQTT.....	23
2.2.6.1. HiveMQ	23
2.2.6.2. VerneMQ	23
2.2.6.3. Mosquitto.....	23
2.2.7. Python	24
2.2.8. Plataforma.....	24
2.2.8.1. PlatformIO	24
2.2.8.2. Arduino IDE	24
2.2.9. Tarjetas	25
2.2.9.1. Ethernet Shield.....	25
2.2.9.2. Raspberry Pi.....	26
2.2.10. Módulo de conexión.....	27
2.2.10.1. ESP32	27
2.2.10.2. ESP8266	27
2.2.11. Servomotor	28
2.2.12. Sensor de distancia.....	28
2.2.13. Sensor de peso	29
2.3. Estado del arte	30
2.3.1. Dispensadores automáticos	30

2.3.1.1. Dispensador automático de alimento para mascotas.....	30
2.3.1.2. Programmable Pet Feeder.....	30
2.3.1.3. Happy Pet Dispensador electrónico para mascotas de agua y comida.....	30
2.3.1.4. Automatic Pet Feeder	31
2.3.1.5. Diseño de dispensador automático de alimento para mascotas usando una celda.....	31
2.3.1.6. Dispensador automático de alimento seco para mascotas caninas	31
2.3.2. Dispensadores basados en el IOT	32
2.3.2.1. Smart dog Feeder design using Wireless communication, MQTT and Android Client	32
2.3.2.2. Desarrollo de un prototipo para dispensar alimento y agua para mascotas con tecnología IOT monitoreado y controlado a través de un dispositivo móvil	32
2.3.2.3. The study and Application of the IoT in pet System	32
2.3.2.4. Control domótico basado en el protocolo MQTT	33
2.3.2.5. Monitorización de sensores con Arduino utilizando el protocolo MQTT ..	33
2.3.3. Dispensadores a control remoto.....	34
2.3.3.1. Diseño e implementación de un sistema dispensador de alimento para mascota, controlado vía ethernet	34
2.3.3.2. Arduino-based Food and Water Dispenser for Pets with GSM Technology Control	34
2.3.3.3. Dispensador automático de comida para mascotas, programable y controlado remotamente	34
2.3.3.4. Low-Cost IoT Design and Implementation of a Remote Food and Water Control System for Pet Owners.....	35
2.3.3.5. Automatización y asistencia remota para cuidado de mascotas	35
Capítulo III. Desarrollo	36
3.1. Recolección de información e investigación.....	37
3.2.1. Diseño de la investigación y procedimiento.....	37
3.2.2. La población y la muestra	37
3.2.3. Criterios de inclusión y exclusión.....	39
3.2.4. Instrumentos	39
3.2. Análisis y requerimientos del proyecto.....	40
3.2.1. Análisis del broker y nodo lector.....	40
3.2.1.1. Broker MQTT	42

3.2.1.2. Nodo lector	42
3.2.1.3. Sistema electrónico	42
3.2.2. Requerimientos de los sistemas	43
3.2.3. Elaboración de las tablas de porciones de alimento.....	43
3.3. Configurar Raspberry Pi como Broker MQTT	45
3.3.1. Instalar sistema operativo	45
3.3.2. Configurar conexión y cliente SSH	45
3.2.3. Configuraciones iniciales en consola.....	47
3.2.4. Configurar IP estática.....	48
3.2.5. Implementación del broker MQTT y clientes	49
3.2.6. Arquitectura de un sistema MQTT	51
3.2.7. Filtrar mensajes con base al Topic.....	52
3.2.8. Publicar desde el broker con cliente Python MQTT.....	53
3.2.9. Automatización del dispensador.....	54
3.2.10. Instalación y configuración del servidor LAMP.....	55
3.4. Configuración de cliente y port forwarding para acceso remoto.....	57
3.5. Configurar NodeMCU ESP8266 como nodo lector	59
3.5.1. Preparación del entorno Arduino IDE.....	60
3.5.2. Configuración de red WiFi.....	60
3.5.3. Implementación de MQTT	63
3.5.4. Conexión de los componentes	64
Capítulo IV. Análisis de los datos y resultados	66
4.1. Datos Pet-Home ALBA	66
4.2. Pruebas de funcionamiento del nodo lector	69
4.3. Pruebas de funcionamiento del broker MQTT.....	71
Conclusiones	73
Bibliografía	75
Anexos	79

Tabla de Figuras

Figura 1. Dispensador común.	3
Figura 2. Dispensador Petnet.	4
Figura 3. Dispensador Bistro.	5
Figura 4. Dispensador Fresh Element.	5
Figura 5. Dispensador CleverPet.....	6
Figura 6. Dispensador Catspad.	6
Figura 7. Dispensador Petsafe.....	7
Figura 8. Dispensador Petmate.	7
Figura 9. Porcentaje de la población según condición de Internet, 2018.	8
Figura 10. Porcentaje de la población según uso de telefonía celular.....	9
Figura 11. Propuesta de solución: Pet-Home ALBA.	13
Figura 12. Metodología	15
Figura 13. Tarjeta Ethernet Shield.	25
Figura 14. Tarjeta Raspberry Pi Model B.....	26
Figura 15. Módulo ESP32.	27
Figura 16. Módulo ESP8266 V3.....	27
Figura 17. Servomotor.....	28
Figura 18. Sensor ultrasónico HC-SR04.....	28
Figura 19. Módulo HX711.	29
Figura 20. Cronograma de actividades.	36
Figura 21. Ecuación para obtener del tamaño de la muestra.	37
Figura 22. Determinar el tamaño de muestra.....	39
Figura 23. Diagrama de bloques esquema general.....	40
Figura 24. Diagrama de secuencia general.	41
Figura 25. Sistema electrónico del prototipo	42
Figura 26. Porciones de alimento para gatos.	44
Figura 27. Porciones de alimento para perros.....	44
Figura 28. SO utilizado: Raspbian Buster.	45
Figura 29. Cliente SSH: PuTTY.	46
Figura 30. TightVNC: Escritorio Raspberry.....	47
Figura 31. Configuraciones con raspi-config.	47
Figura 32. Configuración archivo /etc/dhcpd.conf.....	49
Figura 33. Topología MQTT.	51
Figura 34. Publicación/Suscripción MQTT.....	52
Figura 35. Ejemplo de estructura de un Topic.	52
Figura 36. Wildcards nivel único.	53
Figura 37. Wildcards nivel múltiple.....	53
Figura 38. Diagrama de secuencia modo automático.....	54
Figura 39. Configuración NO-IP.....	57
Figura 40. Configuración cliente NO-IP en servidor.	58
Figura 41. port forwarding.....	58
Figura 42. Especificaciones del módulo ESP8266 V3.....	59

Figura 43. Configuraciones para la placa NodeMCU ESP8266.....	60
Figura 44. Diagrama de bloques conexión WiFi-Manager.	61
Figura 45. Diagrama de flujo WiFiManager.	62
Figura 46. Diagrama de implementación MQTT en ESP8266.....	64
Figura 47. Esquemático del circuito elaborada en Fritzing.	65
Figura 48. Resultado de la población que cuenta con Internet y un Smartphone.	66
Figura 49. Resultado de la población que cuentan con una mascota y que tipo es.....	67
Figura 50. . Resultado de la población que están fuera de casa y el motivo.	67
Figura 51. Resultado de la población que ha adquirido un dispensador.	68
Figura 52. Resultado de la población que está dispuesto a comprar uno y cuánto pagaría..	68
Figura 53. Circuito electrónico para el prototipo.....	69
Figura 54. Conexión del nodo lector a wifi por WiFiManager.....	70
Figura 55. Balanza digital prueba sin alimento	71
Figura 56. Hostname del cliente noip.....	71
Figura 57. Cliente Python MQTT: controlar.py	72
Figura 58. Base de datos Mascotas	72
Figura 59. Base de datos tabla configuración del dispensador.	72

Introducción

En la actualidad las personas cuentan con un ritmo de vida acelerado a lo que ha llevado a “*la sociedad de la información y del conocimiento el acelerado desarrollo de productos intensivos en tecnología y con un gran valor agregado que se inserta en la sociedad, a nivel mundial*” [1], lo cual ha inundado nuestro entorno con televisores, teléfonos, módems, redes y etc., por medios de sistemas tecnológicos integrados donde las *Tecnologías de la información y comunicación* es uno de los principales responsables y que además aborda uno de los fenómenos que en la actualidad está ganando cada vez más reconocimientos, el *Internet de las cosas* y el otro la domótica que se refiere a controlar diferentes tareas del hogar de forma remota e incluso automatizarlas mejorando la calidad de vida.

México es un país con un gran porcentaje de su población que cuenta con alguna mascota, siendo una de estas razones para aprovechar el innovar productos en este campo. por esto, se lleva a cabo el desarrollo de un prototipo dispensador para perros y gatos que brinde el suministro de alimento para la mascota ofreciendo al usuario el control de porción del alimento a ingerir con base a la información de la mascota. Este prototipo, utiliza Raspberry pi y Arduino aplicando el IoT de tal manera que la integración de estas herramientas permite al usuario monitorear y controlar de forma remota y automática mediante un dispositivo móvil. Este proyecto de titulación se encuentra organizado de la siguiente forma, donde:

Capítulo I. Generalidades muestra el porqué, los objetivos, alcances y limitaciones que presenta el desarrollo del proyecto *Pet-Home ALBA*.

Capítulo II. Análisis de los fundamentos subdividido en un marco histórico que son los conceptos y áreas de aplicación, marco teórico que se muestran las herramientas para solucionar el desarrollo del proyecto y estado del arte que son antecedentes del proyecto.

Capítulo III. Consiste en el desarrollo que provee en detalle la metodología utilizada para la elaboración del prototipo que se encuentra dividida en secciones la parte hardware que es el nodo lector y el servidor, y la parte software que consiste en una aplicación móvil para cumplir los requerimientos y permitir el funcionamiento del sistema.

Capítulo IV. Se muestra el análisis de los datos y resultados del funcionamiento del prototipo y la aplicación móvil y las Conclusiones donde se presentan una compilación de los conocimientos adquiridos en el desarrollo de este proyecto.

Capítulo I. Generalidades del proyecto

1.1. Planteamiento del problema

El ser humano ha recibido grandes beneficios de los animales uno de los más importantes, su compañía. Actualmente, se ha incrementado el número de animales domésticos en los hogares, debido a diferentes factores sobre todo porque representan una fuente importante de afecto. En México de cada 100 personas 57 tienen una mascota en su hogar, según datos obtenidos del INEGI; los perros representan aproximadamente el 85% y los gatos un 15% del total [2]. El tener un nuevo integrante en la familia implica una responsabilidad sobre satisfacer sus necesidades básicas como la adecuada alimentación, un espacio físico amplio, cuidado sanitario por mencionar algunos. Por lo tanto, la mayoría de las personas enfrentan cambios en su estilo de vida que se deben a diferentes compromisos personales que conllevan a la irresponsabilidad por parte del dueño para satisfacer las necesidades de su mascota afectando a una de las más importantes, la alimentación.

Para evitar este problema los dueños de las mascotas adquieren un dispensador de alimento para abastecer de alimento a sus mascotas en situaciones que no pueden hacerlo. Actualmente se encuentran dos tipos de dispensadores existentes en el mercado: Dispensadores comunes los cuales no proporcionan la cantidad recomendada de alimento para las mascotas además de que no hay forma de comunicar la necesidad de abastecerse de más alimento, lo que provoca una mala alimentación. También, se encuentran los dispensadores automáticos los cuales se disponen de dos tipos los programables que permiten ajustar las porciones de comida y determinar la hora; y aquellos que funcionan con WiFi a través de una aplicación móvil en los que ajustan las porciones de alimento a dispensar y visualizan la cantidad de alimento restante en el dispensador.

Teniendo en cuenta lo anterior, el presente proyecto propone la construcción de un dispensador de alimento y bebida para mascotas dosificable con la finalidad de suministrar la cantidad de alimento de acuerdo a la información de la mascota, mientras que el desarrollo de una aplicación móvil ayudará a controlar de manera remota el dispensador, asimismo la opción de programar las horas para dispensar el alimento, registrar a la mascota y ver la cantidad de alimento almacenado en el dispensador.

Por lo anterior, se plantea la siguiente pregunta ¿La construcción de un prototipo dispensador de alimento de bajo costo utilizando el Internet de las cosas, controlado remotamente mediante una interfaz, logrará alimentar las mascotas remotamente?

1.2. Antecedentes

En la actualidad se han desarrollado varios productos de esta temática, los cuales se clasifican en dos tipos de dispensadores: comunes y automáticos que estos a su vez tienen distintos modos de trabajo por baterías o corriente eléctrica. Los trabajos mencionados posteriormente harán énfasis en las características de cada módulo que compone la estructura física del dispensador y su funcionalidad. Aunque en el mercado comercial existe una gran variedad de dispensadores de alimento para cada tipo de mascotas se deben tener algunos puntos generales en cuenta, ya sea por la raza, tamaño, peso, dosificación adecuada, tipo de alimento o condiciones especiales que requiera la mascota y el gusto de los usuarios.

1.2.1. Dispensadores comunes

La *Figura 1* muestra los dispensadores comunes que consisten en un depósito en donde se almacena el alimento el cual se va dispensando en cantidades que tal vez no sean recomendables para la mascota sobre su base a un cuenco sea de plástico o cerámica, sin embargo, depende de las mascotas la decisión de alimentarse, además que no poseen una funcionalidad que informe cuando el alimento se esté acabando o sea necesario rellenar el contenedor más que con el contacto visual.



Figura 1. Dispensador común.

1.2.2. Dispensadores automáticos o inteligentes

Con el auge de la tecnología actualmente se están cubriendo más áreas e innovando. A tal punto que un dispensador común va más allá de solo ser un simple comedero, haciendo la vida más fácil a aquellos que por ciertos motivos no pueden estar en todo momento a lado de su mascota. A continuación, se describirán algunos dispensadores automáticos e inteligentes cada uno con ciertas particularidades.

1.2.2.1. Dispensador Petnet

Petnet es un dispensador que se controla con una aplicación móvil como se muestra en la *Figura 2* el cual se puede ajustar el horario de alimentación, recibir alertas de cuando se está produciendo la ingesta de comida y saber el inventario de comida restante en el dispositivo. Posee una batería que se activa en caso de que se corte corriente eléctrica, de manera que se asegura el suministro de la alimentación en dichos casos, el costo del dispensador es de \$8,500.00 aproximadamente.



Figura 2. Dispensador Petnet.

1.2.2.2. Dispensador Bistro

Bistro es un dispensador que incorpora un sistema de reconocimiento facial como se muestra en la *Figura 3* y que es controlado por una plataforma que pesa al animal cuando acude a comer, que aportan los datos necesarios para llevar un registro de los hábitos de alimentación de cada animal, y así regular la porción en función de su peso. El dispensador envía notificaciones al móvil del propietario si se producen incidencias o cambios en la alimentación habitual, su costo oscila entre los \$4625.29.



Figura 3. Dispensador Bistro.

1.2.2.3. Dispensador Fresh Element

Fresh Element es un dispensador para perros y gatos como se muestra en la *Figura 4*, que permite programar desde la aplicación de Peket configurar horarios y cantidades de comida, cambiar la configuración para cuando se requiera bloquear y activar el alimento en cualquier momento. Tiene una capacidad de 2,7kg de alimento seco, activa un led y una notificación en la app cuando baja la cantidad de alimento e incorpora un botón de alimentación, permitiendo dar le un premio extra a la mascota cuando es necesario, el costo del dispensador \$4591.48 aproximadamente.



Figura 4. Dispensador Fresh Element.

1.2.2.4. Dispensador CleverPet

CleverPet es un dispensador para perros como se muestra en la *Figura 5* que obliga a resolver retos visuales y sonoros para entregarle el alimento. El dueño coloca cierta cantidad de premios, si el perro obedece a la instrucción, el sistema girará y entregará el premio; en caso contrario, seguirá recordándole hasta que el perro entienda el proceso para ganar la recompensa. El dispensador se controla por una app móvil que permite a los dueños llevar un registro de cuántos premios ha conseguido la mascota en un día, los ejercicios que logró

y el tiempo que tardó en resolverlos, e incluso podrá elevar la dificultad de los retos, el costo del dispensador estimado es de \$5,179.85.



Figura 5. Dispensador CleverPet.

1.2.2.5. Dispensador Catspad

Catspad es un dispensador de comida y agua como se muestra en la *Figura 6* consiste en un compartimiento de alimento con una capacidad de 1.7kg y bebida con una capacidad de 8L. Controlado por una app móvil, que permite suministrar alimento cada cierto tiempo manual o programarlo para que el dispensador funcione de forma autónoma. El dispositivo posee sensores que detectan el collar de la mascota para saber de qué gato se trata en caso de que se tenga más de un gato en casa, el costo de venta estimado es de \$4,622.88.



Figura 6. Dispensador Catspad.

1.2.2.6. Dispensador Petsafe

Petsafe es un dispensador para gatos y perros de raza pequeño y mediano, funciona con una aplicación móvil en iPhone, iPod, IOS 9 o posterior y Android 5.0 o posterior como se muestra en la *Figura 7* el cual permite dispensar alimento hasta doce veces al día con una capacidad de 5kg de alimento seco, el costo aproximado es de \$4,407.00.



Figura 7. Dispensador Petsafe.

1.2.2.7. Dispensador Petmate

Petmate dispensador que repone automáticamente el depósito de alimentos cuando los niveles son bajos como se muestra en la *Figura 8*. Cuenta con dos compartimientos uno para alimento con una capacidad de 1.4 kg y otro para agua con una capacidad de 2.84 litros, el costo aproximado del dispensador es de \$399.00.



Figura 8. Dispensador Petmate.

1.3. Justificación

Las Tecnologías de la Información y Comunicación (TIC) han dado grandes avances en los últimos años, que son un elemento clave en la innovación y la creación de nuevos productos y negocios, el cual tiene un amplio campo de aplicación que se encuentran enfocados tanto a personas como a animales. La mayoría de las tendencias tecnológicas de las TIC van dirigidas a temáticas relacionadas con: El internet, el desarrollo de software, desarrollo de aplicaciones móviles y el Internet de las Cosas (IoT). La internet actualmente es una herramienta que se utiliza a diario y ayuda a que las actividades sean más fáciles contribuyendo a la comodidad de los usuarios y a las empresas.

El uso del internet, estudiado por el Instituto Nacional de Estadística y Geografía (INEGI) con base a “Estadísticas a propósito del día mundial del Internet (17 de mayo) Datos nacionales” que se muestra en la *Figura 9*, de las personas encuestadas de seis años o más en el país en 2018, 74.3 millones declararon sí usar Internet (65.8% de la población). Lo cual indica un avance de 1.9% respecto de los datos de 2017 (63.9%) [3].



Figura 9. Porcentaje de la población según condición de Internet, 2018.

Fuente: INEGI, ENDUTH 2018.

Sobre la telefonía celular, se mantiene como un servicio de gran importancia para la población por su rápida conectividad, fácil traslado y acceso, así como su multifuncionalidad constantemente actualizada. En la *Figura 10* se aprecia que el 73.5% de la población indicó tener a su disposición un teléfono celular de lo cual, un 83.8% cuenta con al menos un celular común y un celular llamado “Smartphone”; un 0.8% de la población cuenta con al menos un celular común y un celular inteligente, y un 15.4% de la población únicamente cuenta con al menos un celular común.

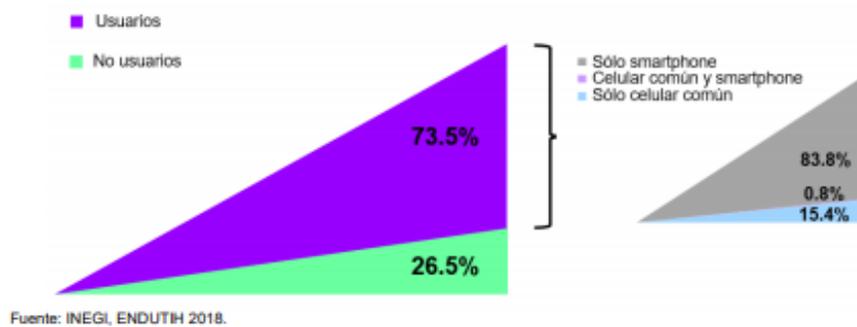


Figura 10. Porcentaje de la población según uso de telefonía celular

Fuente: INEGI, ENDUTH 2018

Teniendo en cuenta la información expuesta anteriormente, es viable la construcción de un prototipo aplicando la domótica en el contexto del IoT, que es un control inalámbrico o automático sin necesidad de conocer la distancia y en la cual, emplean tecnología sofisticada de internet al conectar objetos de la vida diaria con sensores incrustados en ellos y, además de ser controlados y proporcionar información al usuario [4] por algún medio como, el Smartphone. Gracias a esto el mercado de productos dirigido a las mascotas va en crecimiento y representan un nicho por lo cual invertir. En la actualidad existen diversos tipos de dispensadores los cuales tienen muchas funcionalidades (como cámaras, sensores, audio, grabación, música, etc.), pero la mayoría de ellos con un precio muy elevado en el mercado.

Por lo tanto, el dispensador “Pet-home ALBA” que se presenta como propuesta en este proyecto estará constituido por un servidor, un circuito electrónico y una aplicación móvil haciendo que este prototipo sea de bajo costo y accesible a todas aquellas personas que tienen un perro o un gato facilitándole el suministro alimenticio según la información de la mascota del usuario, además de mostrar la cantidad de alimento en el dispensador.

1.4. Objetivos

1.4.1. Objetivo general

Diseñar y construir un prototipo de dispensador remoto de alimento para mascotas basado en la Internet de las cosas, con la finalidad brindar nuevas herramientas que permitan mejorar la alimentación de las mascotas que no pueden ser asistidas físicamente.

1.4.2. Objetivo específicos

- Investigar porción de alimento para consumo de cada mascota en libros, revistas veterinaria y en empaques de alimento para una dosificación adecuada.
- Diseñar y crear la base de datos, para una dosificación de alimento adecuada para las mascotas, información del dispensador y clientes.
- Realizar un análisis de los diferentes componentes electrónicos tomando en cuenta las características y funcionalidades, para el desarrollo del dispensador.
- Determinar los componentes electrónicos y diseñar el circuito, para ser implementado al dispensador en la dosificación de alimento.
- Almacenar la información proveniente del dispensador para su posterior visualización.
- Desarrollar la aplicación móvil, para controlar de forma remota el dispensador.
- Realizar las pruebas por parte del servidor, la aplicación móvil y el dispensador en un ambiente real con la finalidad de comprobar una correcta funcionalidad.

1.5. Hipótesis

Es posible construir un dispensador remoto de alimento para mascotas basado en el Internet de las cosas para mejorar la alimentación de las mascotas a partir de herramientas de bajo costo.

1.6. Alcances y limitaciones

1.6.1. Alcances

El prototipo será probado en mascotas perros y gatos en la región de Misantla, Veracruz, se realizan las pruebas de funcionalidad a través de una aplicación móvil para controlar el dispensador dosificable de alimento.

El usuario de la aplicación:

- Iniciar sesión con nombre de usuario y contraseña.
- Programar las horas en las que desea dispensar el alimento, en caso de que se encuentre en un lugar sin internet o datos móviles.
- Registrar a su mascota e ingresar los datos que se les solicita.
- Dispensar el alimento cuando el usuario quiera.
- Ver el nivel de alimento restante en el dispensador.

Dispensador de alimento y bebida:

- Protocolo de comunicación IoT.
- Contenedor de alimento con una capacidad para almacenar hasta 4-5 kg.
- Fuente de energía: Siendo el principal por corriente eléctrica.
- Contará con dos modos de conexión: Conectado a WiFi; el cual enviará cierta información al usuario y modo pre-programado; donde se establecen las horas o la cantidad de alimento a libre disposición.
- Guardar datos de manera local.

1.6.2. Limitaciones

- Dispensador orientado únicamente a mascotas como perros y gatos.
- Apto sólo para alimento de tipo seco.
- Dispensador que abastece a una sola mascota.
- Dispensador que solo funciona conectado a luz eléctrica.
- Depende del material el dispensador se adapta a interior o exterior.

1.7. Propuesta de solución

En la *Figura 11*, se muestra que el proyecto consiste en el diseño de un sistema versátil, utilizando plataformas y herramientas de desarrollo como un servidor que se encuentre comunicándose mediante un protocolo de internet a un controlador que proporcione conectividad a los diferentes sensores y actuadores empleados que ejecutarán diferentes tareas. La programación de las tareas por los diferentes componentes conectados al controlador se realizará en un entorno de desarrollo integrado siendo que los sensores encargados de registrar los datos del nivel de alimento y enviarlo al servidor y los actuadores que suministrarán el alimento a la mascota. Por otra parte, el usuario mediante una aplicación móvil en que accede por un nombre de usuario podrá registrar el número de dispensador, configurar los datos de la mascota y con base a esa información indicar la cantidad adecuada de alimento por porción, y configurar los horarios para suministrar el alimento a su mascota; de tal forma que pueda monitorear y controlar el prototipo que se encuentra en su casa o departamento para alimentar a su mascota.

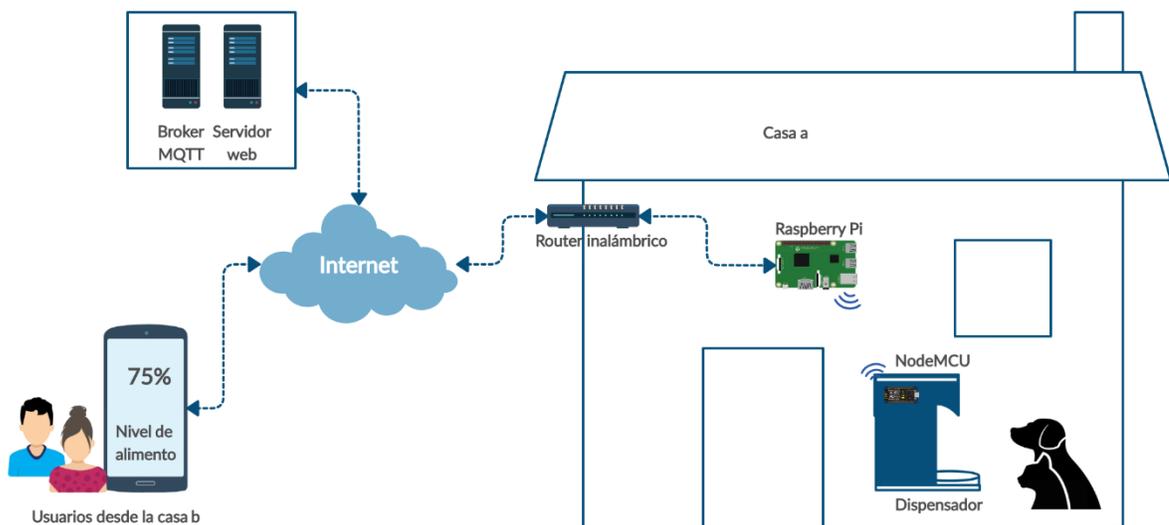


Figura 11. Propuesta de solución: Pet-Home ALBA.

Fuente: Elaboración propia

1.8. Metodología

Como se mencionó en el *apartado 1.7* se presenta una solución a una de las necesidades existentes de la población en cuanto a la alimentación de su mascota. A continuación, en la *Figura 12* se muestra la metodología empleada en el proyecto que consiste en una secuencia de procesos los cuales buscan dar cumplimiento a los objetivos planteados en el *apartado 1.4*.

El primer proceso indica la *Investigación preliminar* que abarca el diseño de la investigación la cual mostrará la factibilidad del desarrollo del prototipo ante los usuarios, análisis corresponde la interacción que habrá entre el servidor / prototipo y las porciones de alimento por el tipo de mascota, las especificaciones y requerimientos son las características y funcionalidades de las herramientas empleadas en el desarrollo.

El segundo proceso *Desarrollo del prototipo* consiste en la realización del esquema de la conexión de los componentes al módulo NodeMCU ESP8266 y la instalación de las librerías correspondientes a dicho módulo, el diseño de las pantallas de la aplicación para el control del prototipo, las configuraciones del protocolo IoT para el servidor al establecer comunicación entre el prototipo al servidor y la aplicación al servidor.

El tercer proceso *Implementación del prototipo* se realizan pruebas pilotos en ambiente real, consiste en una evaluación realizada por un usuario con la finalidad de conocer qué objetivos se han cumplido al mismo tiempo se evalúa la funcionalidad del sistema completo la aplicación móvil, el broker y el prototipo.

El cuarto proceso *Análisis del resultado* consiste en una descripción sobre qué objetivos planteados se lograron y las mejoras que se puede llevar cabo para la viabilidad del sistema.

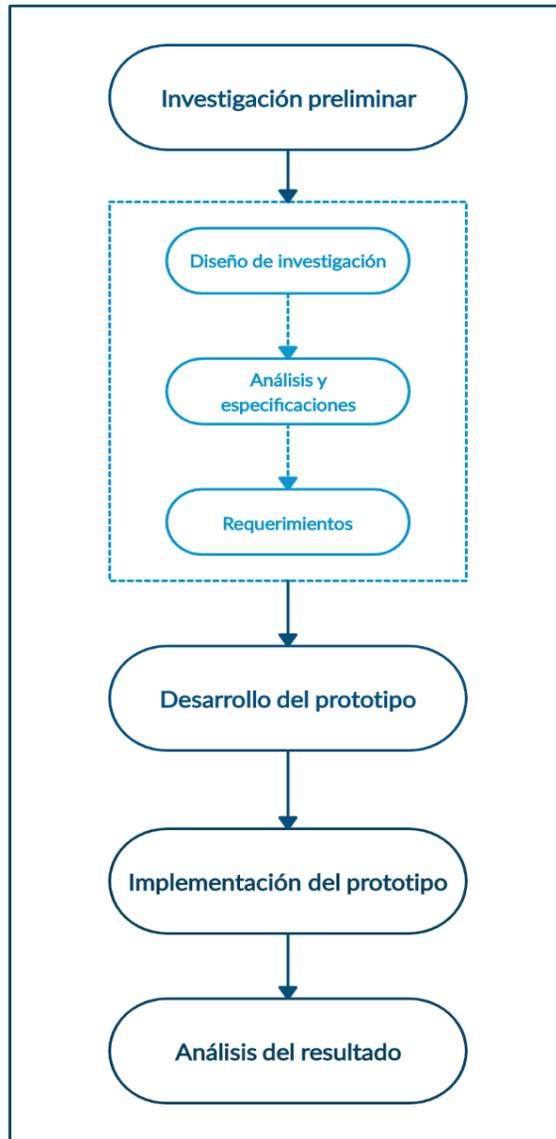


Figura 12. Metodología

Fuente: Elaboración propia

Capítulo II. Análisis de los fundamentos

2.1. Marco histórico

2.1.1. El internet de las cosas

Es una base de datos objeto-relacional se pueden usar los lenguajes SQL y PL/SQL para manipular datos de objetos y datos relacionales. Diseñada para almacenar y recuperar datos de objetos igual que sucede con los datos relacionales, utilizando el SQL como método estándar de comunicación con la base de datos [5].

El *Internet de las cosas*, aunque es tendencia y de importancia técnica, social y económica es un concepto que a pesar de muchos años no cuenta con una definición concisa, sin embargo, fue en 2009, Kevin Ashton, profesor del MIT por aquel entonces que acuñó este término en una presentación realizada en el Procter & Gamble (P&G) en 1999.

“Si tuviéramos ordenadores que supieran todo lo que hay que saber sobre las cosas - utilizando los datos que recogen sin nuestra ayuda- podríamos rastrear y contar todo, y reducir enormemente el desperdicio, la pérdida y el coste. Sabríamos cuándo hay que reemplazar, reparar o retirar las cosas, y si están frescas o han pasado su mejor momento.” [6].

Lo anterior ha originado que se realicen muchas investigaciones y proyectos en este ámbito por lo cual, se han ido obtenido a lo largo del tiempo diferentes definiciones sobre lo que significa el IoT, a lo cual podríamos decir que son: *“Dispositivos incrustados en objetos utilizados en la vida cotidiana desde una playera, una cafetera hasta un refrigerador, etc que tienen la capacidad de almacenar y comunicarse entre sí por la red Internet para así poder procesar y distribuir la información para una toma de decisiones además de poder controlarlos remotamente”*. Siendo el término del IoT cada vez más popular para descubrir escenarios en los que la conexión del internet y la capacidad de cómputo se extiendan a una variedad de objetos y dispositivos que pretendan facilitar la vida cotidiana y en el cual se

puede observar que existen numerosos ámbitos de aplicación de esta tecnología y que sirven para mejorar diferentes aspectos de nuestras vidas como lo son en la domótica que se define como un sistema capaz de automatizar todos los aparatos de una vivienda o edificación, electrodomésticos que se comunican con el usuario los cuales pueden avisar cierta información a través de un dispositivo y tendencias de uso de los consumidores como poder detectar si un corredor tiene o no los zapatos adecuados para los trayectos que realiza.

2.1.2. El dispensador

Los dispensadores de alimento surgen de la idea inicial de las máquinas expendedoras, una máquina expendedora es aquella que proporciona bebidas, aperitivos o golosinas, entre otros productos a los consumidores, su objetivo principal es vender sin la presencia de un dependiente para cobrar y despachar los artículos, de forma periódica un usuario o empleado repone los productos y recoge el dinero en forma de monedas o billetes, aunque esta última es poco frecuente, con el avance de la tecnología ya se puede pagar en algunas dispensadoras con tarjeta monedero, tarjeta de crédito o teléfono móvil. Las posibilidades de las máquinas expendedoras son amplias, suelen adaptarse de acuerdo a las necesidades de los usuarios para vender libros, cigarros, bebidas e incluso para suministrar alimento para mascotas.

Los dispensadores de alimento para mascotas fueron diseñados para facilitar el cuidado de la mascota al momento de suministrar el alimento, ahorra tiempo y así como las máquinas expendedoras solo necesitan que sean rellenadas una sola vez de forma periódica, estas funcionan de forma mecánica cuyo funcionamiento es sin intervención de ningún mecanismo eléctrico, son sencillas, y presentan algunas limitaciones, también existen las electrónicas las cuales implementan las nuevas tecnologías y hacen uso de componentes eléctricos para su funcionamiento, también necesitan de la corriente eléctrica o baterías para poder funcionar.

2.2. Marco teórico

2.2.1. Base de datos

Las bases de datos son colecciones de información (datos) dispuestos con el objetivo de proporcionar información a los usuarios y permitir transacciones como inserción, eliminación y actualización de datos, que se relacionan para crear un sentido y dar más eficiencia a una encuesta, un estudio organizado o la estructura de datos de una empresa, son de vital importancia, y en las últimas décadas se han convertido en la parte principal de los sistemas de información [7].

2.2.2. Sistema gestor de base de datos

Un sistema de gestión de bases de datos (SGBD) es el nombre dado a un conjunto de programas informáticos que gestionan una base de datos. Su objetivo principal es evitar la manipulación directa por un usuario de una base de datos y establecer un marco estándar para que los datos sean organizados y manipulados y tengan una interfaz estándar para que otros programas puedan acceder a la base de datos. En el mercado hay muchos SGBD, en las bases de datos relacionales los sistemas más populares son: Oracle, SQL Server, MySQL y PostgreSQL [7].

2.2.2.1. Oracle

Es una base de datos objeto-relacional se pueden usar los lenguajes SQL y PL/SQL para manipular datos de objetos y datos relacionales. Diseñada para almacenar y recuperar datos de objetos igual que sucede con los datos relacionales, utilizando el SQL como método estándar de comunicación con la base de datos [5].

2.2.2.2. PostgreSQL

Es un sistema de gestión de base de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es uno de los sistemas de gestión de base de datos junto con MySQL de código abierto más potente del mercado [8].

2.2.2.3. MySQL

MySQL es un sistema de base de datos libre, de código abierto, rico en funciones para los usuarios novatos, y más sencillo que otros sistemas con características similares como PostgreSQL. Es importante destacar que, en la actualidad en el mercado, MySQL está más valorado en el segmento de desarrollo web [7].

2.2.3. Servidor web

2.2.3.1. Wamp

WAMP es un acrónimo que significa Windows, Apache, MySQL y PHP. Es una pila de software que significa que se instala Apache, MySQL y PHP en su sistema operativo (Windows en el caso de WAMP). Aunque puede instalarse por separado, generalmente están agrupados y también por una buena razón. Lo que es bueno saber es que WAMP deriva de LAMP, para sistemas operativos basados en LINUX. El acrónimo de WAMP significa:

- **W:** Significa Windows.
- **A:** Significa Apache. Apache es el software del servidor responsable de servir las páginas web cuando solicita que vea una página. Apache le otorga su solicitud a través de HTTP y le muestra el sitio.
- **M:** Significa MySQL, el trabajo de MySQL es ser el sistema de gestión de bases de datos para su servidor. Almacena toda la información relevante, como el contenido de sus sitio, los perfiles de usuario, etc.
- **P:** Significa PHP. Es el lenguaje de programación que se utilizó para escribir WordPress. Actúa como pegamento para toda esta pila de software. PHP se ejecuta junto con Apache y se comunica con MySQL [9].

2.2.3.2. Laragon

Laragon es una suite de desarrollo para PHP que funciona sobre Windows diseñado especialmente para trabajar con Laravel. Similar a otras herramientas como Xampp o Wampp, Es un entorno de desarrollo universal portátil, rápido y potente para PHP, Node.js, Python, java, Go, Ruby. Es rápido, liviano, fácil de usar y fácil de extender. Laragon es ideal para crear y administrar aplicaciones web modernas, se centra en el rendimiento: fue desarrollado

en torno a la estabilidad, la simplicidad, la flexibilidad y la libertad. Es muy ligero y su núcleo binario en sí es inferior a los 2 MB y utiliza menos de 4 MB de RAM cuando se está ejecutando. Laragon no utiliza los servicios de Windows, contiene el propio *servicie orchestration* el cual gestiona los servicios de forma asíncrona y son bloqueo, por cual las cosas se ejecutan de manera rápida y sin problemas [10].

2.2.3.3. Lamp

LAMP, que significa Linux, Apache, MySQL y PHP es una robusta combinación de aplicaciones impulsadas por el poderoso sistema operativo Linux. La arquitectura abierta de cada uno de estos elementos permite una integración fluida y sin problemas entre sí y resulta en una combinación de tecnologías con un historial probado de ser eficiente, segura y siempre a la vanguardia de la siempre popular Internet [11].

2.2.4. Lenguaje de PHP

PHP es un Lenguaje de programación libre bastante popular, especialmente adecuado para crear sitios web e incrustarse en HTML. Este lenguaje se utiliza para generar páginas dinámicas, es decir, aquellas cuyo contenido cambia según ciertas circunstancias, por ejemplo, dependiendo de la información contenida en una base de datos, de lo que ingrese el usuario o de una búsqueda realizada. A continuación, se mencionan algunas ventajas de las más importantes:

- Puede conectarse con muchos motores de bases de datos, y se destaca su compatibilidad con algunos de los más utilizados, por ejemplo, MySQL y PostgreSQL.
- El código que creamos en PHP será invisible al navegador, ya que es ejecutado por el servidor, esto lo convierte en una opción bastante segura y confiable.
- Es posible aumentar el potencial de PHP mediante módulos o extensiones; además, podemos acceder a una completa documentación en su web oficial.
- Es un lenguaje libre, por lo que resulta accesible para todos [12].

2.2.5. Protocolos para el internet de las cosas

Los protocolos de comunicación poseen un papel vital dentro del desarrollo de proyectos que requieren comunicación constante con el servidor o con los dispositivos, es por eso que se desarrollaron protocolos diseñados específicamente para facilitar la transmisión de información de manera rápida y ligera.

2.2.5.1. CoAP

CoAP (Constrained Application Protocol) es un protocolo de transferencia especializado en la red para el uso de nodos y redes compactadas en el IoT. Este protocolo está diseñado para transferencias máquina a máquina (M2M) tales como Smart Energy o Smart Building. Las características de CoAP son:

- Modelo de reposo para los dispositivos pequeños: Los dispositivos solo envían mensajes en caso de ser necesario mediante métodos como GET, PUT, POST o DELETE.
- Hecho para millones de nodos de todo tipo: CoAP permite a microcontroladores desde 10 kb de RAM y 100 kb de espacio direccionarle conectarse de manera adecuada.
- Transferencia de habilidades existentes (HTTP COAP): CoAP un sistema muy similar a HTTP por lo que un programador que posea experiencia HTTP puede fácilmente adaptarse a desarrollo COAP.
- Seguridad: CoAP por defecto implementa parámetros de seguridad equivalen a llaves RSA de 3072-bit.
- Fácil integración: CoAP comparte características con HTTP por lo que pueden ser conectados de manera sencilla.
- Descubrimiento integrado: Cada nodo con CoAP tendrá un modelo de identificación para permitir a los demás nodos descubrirlos de manera sencilla.

2.2.5.2. Rest API

REST (Representational State Transfer) es un estilo de arquitectura para sistemas hipermedia distribuidos. REST es un estilo híbrido derivado de varias arquitecturas orientadas a la red. Las restricciones establecidas son:

- Interfaz uniforme: Esta restricción define que la interfaz existente entre clientes y servidores debe mantenerse igual en todos los nodos, lo cual permite que pese al desenvolvimiento individual de cada nodo la información se transfiera de igual manera permitiendo conectividad todo el tiempo.
- Sin estado: Para minimizar el ancho de banda ocupado por una sesión abierta convencional, el sistema REST implementa un envío único de información en el que se dan todas las indicaciones necesarias al nodo receptor para el adecuado tratamiento de la información sin necesidad que este último tenga que comunicar el estado del proceso de la información.
- Cacheable: Las comunicaciones pueden o no almacenar un caché de las mismas para ser utilizadas en futuras transacciones minimizando el tiempo de procesamiento.
- Cliente servidor: El cliente y el servidor poseen estructuras internas distintas, centradas a la tarea de cada uno de estos.
- Sistema en capas: Rest permite la implementación de servidores intermediarios para mejorar la escalabilidad del sistema activando el balance de carga [13].

2.2.5.3. MQTT

MQTT es un protocolo completamente creado en 1999 por el Dr. Andy Stanford-Clark de IBM y Arrlen Nipper de Arcom. Es un protocolo abierto, de fácil implementación, además sencillo y ligero. Es ideal para responder a las necesidades del IoT dado que está adaptado para utilizar un mínimo de espacio en el ancho de banda, lo cual lo hace ideal para ser usado en redes inalámbricas, MQTT es ampliamente utilizado en una gran variedad de artefactos tanto industriales como particulares. Entre las principales características de MQTT se pueden notar:

- Bajo consumo energético.
- Rápido y ligero para el envío y recepción de información.
- Requiere pocos recursos para su adecuado funcionamiento.

MQTT está basado en una estructura suscriptor/publicador donde el servidor (suscriptor) llamado broker recibe los datos de los publicadores y en caso de ser necesario, enviará datos a los publicadores únicamente bajo petición del broker. Ningún mensaje puede jamás poseer un tamaño superior a 256 Mb [13].

2.2.6. Broker MQTT

2.2.6.1. HiveMQ

Facilita el traslado de datos desde y hacia los dispositivos conectados de manera eficiente, rápida y confiable, el cual está diseñado para implementaciones nativas de la nube para hacer un uso óptimo de los recursos de la nube; MQTT reduce el ancho de banda de red requerido para mover datos. HiveMQ conecta cualquier dispositivo y sistemas de back-end de manera confiable y segura a través del protocolo estándar IoT MQTT [14].

2.2.6.2. VerneMQ

Agrupación de MQTT para alta disponibilidad y escalabilidad. VerneMQ es un software de código abierto, ampliable y dispone de soporte empresarial. Se escala horizontal y verticalmente en hardware básico para admitir una gran cantidad de editores y consumidores concurrentes, manteniendo baja latencia y tolerancia a fallas. VerneMQ es el centro de mensajes confiable para su plataforma IoT o productos inteligentes [15].

2.2.6.3. Mosquitto

MQTT es solo un estándar de protocolo, y aún requiere herramientas de software para que pueda implementarse en aplicaciones reales. Mosquitto es una implementación de código abierto del agente de mensajes, que admite la última versión del protocolo MQTT standard. El agente de Mosquitto habilita el paradigma pub / sub implementado por el protocolo MQTT, al tiempo que proporciona un mecanismo liviano para permitir la mensajería entre máquinas. El desarrollo de Mosquitto se apoya a través de los esfuerzos de la comunidad. Mosquitto es una de las implementaciones MQTT más populares, disponible gratuitamente y ampliamente compatible en Internet [16].

2.2.7. Python

Python es un lenguaje de programación interpretado, multiparadigma, de alto nivel con un tipado dinámico fuerte, dotado de una gestión automática de los recursos, de un alto grado de introspección y de un sistema de gestión de excepciones. Es libre y gratuito, funciona sobre todas las plataformas, apareció en 1990 y posee varias implementaciones, entre ellas CPython, Jython, IronPython y PyPy. Su licencia es la Python Software Foundation License. Es relativamente cercana a la licencia BSD y compatible con la licencia GPL. Su sintaxis es minimalista, explícita, clara, sencilla y lo suficientemente cercana al lenguaje natural como para permitir que un algoritmo se comprenda tras su primera lectura, al menos por un lector que conozca el inglés [17].

2.2.8. Plataforma

2.2.8.1. PlatformIO

PlatformIO inicialmente se desarrolló con la filosofía de la línea de comandos. Se ha utilizado con éxito con otros IDE como Eclipse o Visual Studio. Recientemente han lanzado una versión con IDE integrado basado en el editor de texto Atom.

- Desarrollo rápido: Finalización inteligente de código y Smart Code Linter para un rápido desarrollo profesional. Flujo de trabajo de proyectos múltiples con paneles múltiples.
- Desarrollo embebido: Sistema de compilación multiplataforma sin dependencias externas del sistema operativo más de 750 placas integradas, más de 30 plataformas de desarrollo, más de 20 marcos. Compatible con Arduino y ARM mbed.
- Depurador unificado: PIO Unified Debugger con soporte para múltiples arquitecturas y plataformas de desarrollo que le permite depurar más de 250 placas integradas con configuración cero [18].

2.2.8.2. Arduino IDE

Arduino nació en el año 2005 en el instituto de Diseño interactivo de Ivrea (Italia), apareció por la necesidad de contar con un dispositivo para utilizar en las aulas que fuera de bajo coste, que funcionase bajo cualquier sistema operativo y que contase con documentación adaptada a gente que quisiera empezar de cero. La idea original fue, pues, fabricar la placa para uso

interno de la escuela. No obstante, el Instituto se vio obligado a cerrar sus puertas precisamente en 2005. Ante la perspectiva de perder en el olvido todo el desarrollo del proyecto Arduino que se había ido llevando a cabo durante aquel tiempo, se decidió liberarlo y abrirlo a “la comunidad” para que todo el mundo tuviera la posibilidad de participar en la evolución del proyecto, proponer mejoras y sugerencias y mantenerlo “vivo”. Y así ha sido: la colaboración de muchísima gente ha hecho que Arduino poco a poco haya llegado a ser lo que es actualmente: un proyecto de hardware y software libre de ámbito mundial. Arduino es en realidad tres cosas una placa hardware libre, un software gratis, libre y multiplataforma, y un lenguaje de programación libre [19].

2.2.9. Tarjetas

2.2.9.1. Ethernet Shield

En la *Figura 13* se muestra la tarjeta ethernet shield que proporciona acceso a la red local, por lo tanto, puede ser utilizada en diversos proyectos. Para conectar la shield a la red, es necesario contar con un cable de red RJ45, cuyos extremos deben conectarse al puerto a un punto de red, respectivamente. Una vez que la conexión se ha realizado, las posibilidades de Arduino aumentan, por ejemplo, es posible conectarse internet o crear un servidor web con Arduino. El shield ethernet es una de las más utilizadas, nos otorga acceso a la red local por lo que abre interesantes posibilidades [20].



Figura 13. Tarjeta Ethernet Shield.

2.2.9.2. Raspberry Pi

En la *Figura 14* se muestra la tarjeta Raspberry pi que es una computadora económica construida en una sola placa de circuito impreso. Fue desarrollada en el Reino Unido por la Fundación Raspberry Pi para alentar la enseñanza de la informática básica en las escuelas. Para mantener el precio bajo, el Raspberry Pi 3 Modelo B tiene algunas características de diseño innovadoras:

- En su corazón hay un procesador ARM que tiene una arquitectura System-on-Chip (SoC) para integrar varios componentes tradicionalmente separados en un solo chip.
- A diferencia del diseño de computadora, la Raspberry no tiene un disco duro, sino que emplea una tarjeta Micro-sd para contener el sistema operativo y almacenar los archivos.
- El Raspberry Pi Modelo B tiene una memoria total de solo 1 GB, que es pequeña en comparación con la de las computadoras tradicionales de hoy. Incluso con esta limitación, se logra un rendimiento sorprendentemente bueno, ya que ni el procesador ni el sistema operativo están "hambrientos de memoria". Esto a su vez permite que los programas que se ejecutan en Raspberry Pi utilicen cantidades muy bajas de memoria.
- Lo más notable es que cada Raspberry Pi se suministra sin una carcasa, por lo que puede integrarse fácilmente en otro dispositivo, como un monitor, y sus componentes pueden identificarse fácilmente [21].



Figura 14. Tarjeta Raspberry Pi Modelo B

2.2.10. Módulo de conexión

2.2.10.1. ESP32

En la *Figura 15* se muestra el módulo ESP32 que es un SoC (o combo chip) que ofrece conectividad WiFi y Bluetooth, con frecuencia 2.4 GHz, potencia computacional (CPU + memorias), I/OS, RTC, soporte para la operación de diversas comunicaciones (SPI, I2C, I2S, etc.), soporte operación de baja potencia y bloques de hardware dedicados a la seguridad en un solo chip. Según el datasheet oficial, utilizando el chip junto con aproximadamente 20 componentes externos, es posible disfrutar plenamente de las funcionalidades del ESP32, con el beneficio de ocupar un área muy pequeña en una placa circuito impreso [22].



Figura 15. Módulo ESP32.

2.2.10.2. ESP8266

En la *Figura 16* se muestra el Nodemcu que es un kit de desarrollo y firmware de código abierto. Esta placa tiene un chip de USB a serie, pero carece de un circuito de carga de batería y un conector Li-Po viene con una compilación de firmware LUA de NodeMCU que le permite programarse utilizando el lenguaje de programación LUA, aunque puede programarse utilizando el IDE de Arduino y PlatformIO. El kit de desarrollo NodeMCU viene con 4MB de memoria flash, por lo que puede soportar MicroPython, además de LUA y el núcleo de Arduino [23].



Figura 16. Módulo ESP8266 V3.

2.2.11. Servomotor

En la *Figura 17* se muestra un servomotor –también llamados servos- siendo motores que incorporan además un potenciómetro y cierta circuitería de control para poder establecer la posición del eje del motor de forma precisa. Es decir, su eje no gira libremente, sino que, rota un determinado ángulo, indicado a través de una señal de control. Lo que lo hace especial es que podemos ordenarle que gire una cantidad de grados concreta, cantidad que dependerá de la señal de control enviada en un momento dado por (por ejemplo) un microcontrolador programado por nosotros [19].



Figura 17. Servomotor.

2.2.12. Sensor de distancia

En la *Figura 18* se muestra un sensor ultrasónico que es un dispositivo que puede medir la distancia a un objeto mediante el uso de ondas de sonido. Mide la distancia enviando una onda de sonido a una frecuencia específica y escuchando esa onda de sonido para recuperarse. Al registrar el tiempo transcurrido entre la onda de sonido que se está generando y la onda de sonido que rebota, es posible calcular la distancia entre el sensor de sonda y el objeto [24].



Figura 18. Sensor ultrasónico HC-SR04.

2.2.13. Sensor de peso

En la *Figura 19* se muestra el módulo HX711 que está basado en la tecnología patentada de Avia Semiconductor, el HX711 es un convertidor analógico-digital (ADC) de 24 bits de precisión diseñado para básculas y aplicaciones de control industrial para interactuar directamente con un sensor de puente. El multiplexor de entrada selecciona la entrada diferencial del canal A o B al amplificador de ganancia programable de bajo ruido (PGA). El canal A se puede programar con una ganancia de 128 o 64, correspondiente a un voltaje de entrada diferencial a escala completa de $\pm 20\text{mV}$ o $\pm 40\text{mV}$ respectivamente, cuando se conecta un suministro de 5V al pin de la fuente de alimentación analógica AVDD. El canal B tiene una ganancia fija de 32. El regulador de fuente de alimentación Onchip elimina la necesidad de un regulador de fuente externo para proporcionar energía analógica para el ADC y el sensor. La entrada del reloj es flexible. Los circuitos de encendido y reinicio del chip simplifican la inicialización de la interfaz digital [25].

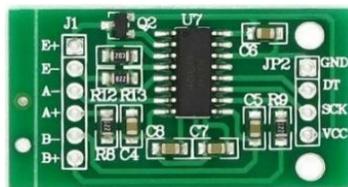


Figura 19. Módulo HX711.

2.3. Estado del arte

Actualmente existen distintos trabajos que han brindado a la sociedad una alternativa para el cuidado alimenticio de las mascotas, estos proyectos utilizan distintas tecnologías, mecanismos y sensores. A continuación, se muestran en detalle los proyectos.

2.3.1. Dispensadores automáticos

2.3.1.1. Dispensador automático de alimento para mascotas

En Cuenca, Ecuador, Icaza Daniel *ét al* [26] dispensador que ofrece al usuario las opciones de configurar la cantidad de alimento dependiendo del tamaño de la mascota y escogiendo dos horarios al día para suministrar el alimento. La estructura consta de un tanque de almacenamiento para 5kg de alimento seco, la configuración y la cantidad de alimento con la que cuenta el tanque se visualiza a través de un display. Su modo de trabajo principal consiste estar conectado a energía eléctrica, pero, en caso de haber un corto cuenta con un banco de baterías que respalda la energía.

2.3.1.2. Programmable Pet Feeder

Tianjin, China, TG Berhan *ét al*. [27] Dispensador programable que es controlado por un microcontrolador, y al cual se le conecta una pantalla LCD para la visualización de datos, un zumbador para alertar a la mascota durante la hora de la comida o cuando el alimento está dispensado, una mesa giratoria que se divide en cuatro secciones diferentes para la cantidad de alimento que, con la ayuda de un motor paso a paso y la elección del usuario, se dispensarán en el momento deseado.

2.3.1.3. Happy Pet Dispensador electrónico para mascotas de agua y comida

Lima, Perú, A. G, Giggiola del Pilar *ét al*. [28] dispensador automatizado que cuenta con dos compartimientos uno que suministra alimento en determinado tiempo y otro para el agua que caerá de forma automática, cada compartimiento podrá ser extraíble individualmente. El dispensador cuenta con tres botones para programar las raciones de comida hasta en tres diferentes horarios diferentes lo cual se hace con un timer, un grabador de voz que se activa cuando se suministra la ración de comida y una pantalla LCD medirá la carga actual de la

batería y contar con un cargador para recargarlo en cualquier enchufe de la casa o departamento.

2.3.1.4. Automatic Pet Feeder

Mumbai, India, K. Aasavari y G. (Mohite), Vaishali [29] dispensador automático que implementa un circuito electrónico que consiste en un Arduino Uno al cual conectan componentes como: sensor de proximidad para detectar a cierta distancia a la mascota y dispensar el alimento, un sensor de peso y un servomotor SG90 con un grado angular de 0-180° que consiste como un sistema de bloqueo para controlar cuánta comida se libera cuando se abre o desbloquea el bloque.

2.3.1.5. Diseño de dispensador automático de alimento para mascotas usando una celda

Colombia, L. –A, Nelson Andrés *ét al.* [30] dispensador de alimento y agua, que dan prioridad al horario de suministro y la cantidad de alimento por ración a perros y gatos de razas pequeñas. El funcionamiento del dispensador integra un circuito electrónico un microcontrolador al que conectan una pantalla alfanumérica en la que se visualiza la configuración realizada por medio de un teclado, un reloj de tiempo real la cual se selecciona la fecha y hora a suministrar el alimento, un servomotor que dispensa el alimento a un dispositivo basado en Galgas extensiométricas que permite medir la cantidad de la porción de alimento a suministrar, además de que el usuario puede grabar su voz en el prototipo que avisará a la mascota que el alimento y el agua ha sido suministrado.

2.3.1.6. Dispensador automático de alimento seco para mascotas caninas

Irapuato, Guanajuato, M. R, Alberto *ét al.* [31] dispensador de alimento para perros con censado y alarma de nivel de alimento, permite al usuario ingresar la edad y tamaño de la mascota, para determinar la porción de alimento a suministrar. La estructura consiste en un contenedor de madera que almacena entre 7-8kg de alimento y un circuito electrónico que incorpora una pantalla LCD, un teclado numérico, un sistema de reloj de tiempo real para suministrar el alimento a la hora especificada; estas horas de alimentación están prefijos para cuatro veces y otro para dos veces al día, un indicador luminoso para indicar bajo nivel de alimento los cuáles son controlados por un microcontrolador PIC18F4550.

2.3.2. Dispensadores basados en el IOT

2.3.2.1. Smart dog Feeder design using Wireless communication, MQTT and Android Client

Tangerang, Indonesia, Vania *ét al.* [32] El dispensador es controlado por una aplicación móvil Android en el cual se establece el tiempo de alimentación, la porción a dispensar, controlar procesos, actualizar el stock de alimento, contar con un historial de la alimentación y el tiempo en el que el alimento fue dispensado, y registrar o anular el registro del dispositivo. El circuito electrónico consiste en un Arduino Uno al que están conectados un lector RFID-RC522 para autenticar, una celda de carga para medir la porción, un servomotor para controlar la abertura de la dispensación, un RTC para establecer el horario de alimentación y un interruptor para detectar la tapa sobre el recipiente; el Arduino Uno espera recibir órdenes del módulo NodeMCU que a su vez recibe de la aplicación móvil tal comunicación es llevada a cabo por el protocolo MQTT.

2.3.2.2. Desarrollo de un prototipo para dispensar alimento y agua para mascotas con tecnología IOT monitoreado y controlado a través de un dispositivo móvil

Quito, Ecuador, A. A., Sandra Nathaly [33] dispensador para mascotas su estructura consta de dos tanques para almacenar alimento y agua se encuentra basado en el IoT su circuito consta de una Raspberry Pi y un Arduino conectando una cámara de video permitiendo al usuario monitorearlo y controlarlo remotamente mediante un dispositivo móvil a través de Internet, los datos que pueden ser monitoreados es el nivel de alimento y agua en el dispensador. El protocolo IoT utilizado para la comunicación del circuito y la aplicación es MQTT como broker Mosquitto.

2.3.2.3. The study and Application of the IoT in pet System

Taipéi, China, C.-M Own *ét al.* [34] proyecto que comprende en un recipiente de alimentación con subdivisiones que almacena solo alimento seco e implementa un módulo WSN que consta de un contador de tiempo proporcionando los horarios de alimentación programada que abre o cierra la cubierta, un mecanismo de bloqueo para mantener la unidad en su lugar y un módulo de reproducción de sonido, el dispensador es accionado por un sensor

de proximidad infrarrojo que consiste en mandos de distancias IR integrado en el collar de la mascota que detecta su presencia y abre la tapa para el acceso al alimento.

2.3.2.4. Control domótico basado en el protocolo MQTT

Valladolid, de la C. C., Javier, [35] proyecto que consiste en una mini-PC GIGABYTE como dispositivo domótico que se conecta a Internet a través de WiFi y el cual se comunica con un módulo NodeMCU que se encarga de un sensor de temperatura y un actuador periférico encargado de accionar la calefacción de la casa mediante un relé que para indicar su estado es a través de unos leds para la interactuar es por una aplicación móvil Android que se encarga de controlar la temperatura y programar modo automático de la caldera. La comunicación e intercambio de información entre el dispositivo domótico y la aplicación se realiza por el protocolo MQTT utilizando el broker Mosquitto.

2.3.2.5. Monitorización de sensores con Arduino utilizando el protocolo MQTT

Catalunya, España, C. C. Mario y G. L. Joan [36] proyecto caracterizado por M2M que consisten en un sensor de temperatura/humedad y un led conectados a un Arduino Uno que se encarga de enviar la información obtenida del sensor y el led a una Ethernet Shield broker MQTT: Cloud MQTT al cual se accede desde la web o móvil desarrollada en IoT MQTT Dashboards que hará de cliente y se suscribirá a los Topics correspondientes de manera que pueda ser controlado el encendido y apagado del led así como obtener información del estado del led y el nivel de la temperatura y humedad.

2.3.3. Dispensadores a control remoto

2.3.3.1. Diseño e implementación de un sistema dispensador de alimento para mascota, controlado vía ethernet

D.F., México, P. T, C. Laura [37] dispensador de alimento para mascotas, como gatos y perros de raza pequeña, controlado vía Ethernet para el envío de información por la red PAN por medio de una página web dinámica. El funcionamiento del dispensador es por un circuito electrónico, que consiste en una ethernet shield para la manipulación de datos de forma remota está enlazada a un Arduino Uno y al cual se conecta un servomotor para abastecer el alimento es manipulado por una página web en donde se visualiza los parámetros que se desean controlar representados mediante indicadores de colores: 1) Morado, para alimentar una mascota chico, 2) Azul, alimento para un mascota mediano y 3) Rosa, alimento para un mascota grande, a lo cual, cada botón proporciona una ración de comida considerado por el tamaño de la mascota.

2.3.3.2. Arduino-based Food and Water Dispenser for Pets with GSM Technology Control

Manila, Filipinas, B. –Jr, Angelo A. *ét al.* [38] dispensador que implementa un circuito electrónico siendo una fuente de alimentación que suministra la energía al módulo GSM Shield que envía un mensaje al dueño para indicarle que el dispensador está inicializado y lista para funcionar. El dispensador consiste en un contenedor para comida y otro para agua, su funcionamiento comienza con el dispositivo móvil enviando un SMS al módulo GSM que activará al servomotor el cual permite suministrar el alimento a la mascota y una válvula solenoide para controlar el flujo de agua, el equipo notificará al propietario y enviará un SMS que la alimentación de la mascota se llevó a cabo.

2.3.3.3. Dispensador automático de comida para mascotas, programable y controlado remotamente

Santiago de Cali, Colombia, L. Q, John David y R. A, Daniel [39] dispensador de comida para mascotas, electrónico, automático y controlado remotamente. La estructura del dispensador es especial para alimentar perros y gatos de pequeño y mediano tamaño, cuenta

con un tanque donde almacena la comida durante un tiempo determinado. Se implementó un circuito siendo el módulo GPRS encargado de manejar la interfaz local los periféricos asociados son una pantalla LCD, un RTC y un teclado procesan la información obtenida de los sensores infrarrojos que indica el nivel de alimento y una celda de carga ubicada en el plato que monitoreaba la cantidad de alimento dispensada mientras para la visualización de los datos.

2.3.3.4. Low-Cost IoT Design and Implementation of a Remote Food and Water Control System for Pet Owners

Estambul, Turquía, Ugur Baran Asaner and Armagan Elibol [40] proyecto que se compone de tres partes: unidad de ejecución, servidor y aplicación móvil. El dispensador que proporciona control de agua y alimentos para mascotas y se controla mediante una aplicación móvil en Android que muestra la cantidad de comida y agua en las últimas 10hrs que se visualizan en gráficas. El servidor una Raspberry Pi tiene el rol de conexión entre otros componentes, ya que administra las solicitudes y respuestas entre la unidad ejecutora y la aplicación móvil, de igual forma proporciona autenticación para utilizar el sistema.

2.3.3.5. Automatización y asistencia remota para cuidado de mascotas

Valencia, España, C., Jeremie [41] proyecto donde el circuito electrónico consiste en un módulo HX711 que trabaja con una celda de carga para calcular el peso del alimento suministrado y un motor para generar movimiento en los engranes hasta dispensar el alimento indicado conectados a un Arduino Nano que es controlado por una Raspberry Pi Zero w implementado como servidor en NodeJS que se comunica por API REST al cliente móvil donde el usuario se registra, registra la mascota y las porciones de alimento para ofrecer un mejor control en su alimentación, y asimismo verificar el estado del dispensador.

3.1. Recolección de información e investigación

3.2.1. Diseño de la investigación y procedimiento

Esta es una investigación de tipo aplicada orientada a demostrar la validez de ciertos métodos, instrumentos y técnicas para dar respuesta a un problema con innovación tecnológica y producto. Según el desarrollo del tema estudiado es transversal.

Su procedimiento consiste en:

1. La primera etapa consistió en realizar la elaboración de la encuesta para su posterior aplicación.
2. Se procede a investigar en los datos del INEGI sobre la población del municipio de Misantla.
3. Por último, se aplicó la encuesta a personas que cumplen con los criterios de inclusión donde se realizará una visita a las personas y otra en línea, la información quedará en anonimato.

3.2.2. La población y la muestra

La población de este estudio es de tipo finita ya que está conformada por un número limitado de elemento la cual se aplica al municipio de Misantla, entidad Veracruz de Ignacio de la Llave que está conformada por un total de 62,919 habitantes [42] de las cuales se tomarán en cuenta a 30441 personas que se encuentran en un rango de edad entre los 20 y 59 años, de sexo tanto femenino como masculino.

La muestra será obtenida referencialmente a través de la fórmula de muestra finita:

$$n = \frac{N * Z_a^2 * p * q}{(e^2 * (N - 1)) + Z_a^2 * p * q}$$

Figura 21. Ecuación para obtener del tamaño de la muestra.

Fuente: Elaboración propia

En donde:

n = Tamaño de la muestra buscada (número de encuestas que vamos a hacer).

N = Es el tamaño de la población o universo (número total de posibles encuestados) siendo 30441 personas que representan los individuos que cumplen con los criterios de inclusión para este estudio.

Z = Es una constante que depende del nivel de confianza que asignemos. El nivel de confianza indica la probabilidad de que los resultados la investigación sean ciertos, para ello el investigador asigna el valor estimado de confianza a través del grado de certeza que quiera de la investigación. En la *Tabla 1* se muestra los valores de Z alfa, dado el nivel de confianza.

Nivel de confianza	99.7%	99%	98%	96%	95%	90%	80%	50%
Z alfa	3	2.58	2.33	2.05	1.96	1.645	1.28	0.674

Tabla 1. Nivel de confianza

Fuente: Elaboración propia

e = Error de estimación máximo aceptado se refiere a la cantidad de error de muestro aleatorio resultado de la elaboración de una investigación, el valor es asignado por el investigador.

p = Es la proporción de individuos que poseen en la población la característica de estudio.

q = Es la proporción de individuos que no poseen esa característica, es decir, $1-p$.

La proporción de individuos que poseen en la población las características de estudio (p) indica la probabilidad de que ocurra el evento con éxito mientras que la proporción de individuos que no poseen esa característica (q) indica la probabilidad de que el evento no ocurra. Cuando no se conoce el valor de p o q , el investigador deberá asignar el mismo peso de ambos eventos.

En lo cual asumimos que queremos un nivel de confianza del 1.645 que pertenece al 90% mientras que consideramos que la probabilidad de éxito y fracaso es del 50%, y deseamos un error de estimación del 10% para 30441 individuos que pertenece al tamaño de la población

para esta investigación. A continuación, en la *Figura 22* se muestra el cálculo para determinar el tamaño de muestra:

Parámetro	Valores
N	30.441
Z	1.645
P	50.00%
Q	50.00%
e	10.00%

Tamaño de muestra
"n" =

67.50

Figura 22. Determinar el tamaño de muestra.

Fuente: Elaboración propia

3.2.3. Criterios de inclusión y exclusión

Quedaron incluidos en este estudio las personas que viven en el Municipio de Misantla, de ambos sexos, con edades comprendidas entre los 20 a 59 años, tanto como cuenten o no con una mascota, pero han dado su consentimiento para la participación de la investigación y se excluyeron del estudio las personas que no presentaron las características contempladas. En la cual se utiliza un listado de preguntas que se presenta de forma física o electrónica a fin de que sean contestadas.

3.2.4. Instrumentos

Para la recolección de la información, como técnica principal se aplica la encuesta de forma personal y a través de Internet, el cual es elaborado por los investigadores y que se muestra en *anexos*, los cuales para una mejor obtención de la información cuenta entre dos a cuatro respuestas de elección.

La encuesta consta de nueve ítems con la finalidad de determinar:

- La cantidad de personas que cuentan con alguna mascota.
- La cantidad de personas que cuentan con internet en casa o un Smartphone.
- La cantidad de personas que estarían dispuestas a adquirir un dispensador inteligente.

3.2. Análisis y requerimientos del proyecto

3.2.1. Análisis del broker y nodo lector

Como se mostró anteriormente en el cronograma durante las primeras semanas se llevó a cabo un análisis del proyecto, a lo cual se hace uso principalmente de las dos siguientes tarjetas:

1. Raspberry pi 3 Model B.
2. NodeMCU ESP8266 v.3.

Además, cabe mencionar que el entorno de desarrollo es ARDUINO IDE ya que es liviano, cuenta con una gran comunidad y disponibilidad de librerías. A continuación, se muestra en la *Figura 23. Diagrama de bloques esquema general* de la solución de propuesta donde se compone de un sistema electrónico que está conformado por sensores y un actuador, un nodo lector que es el NodeMCU ESP8266 y un Broker MQTT siendo la Raspberry Pi y en la *Figura 24. Diagrama de secuencia general* se muestra el análisis realizado.

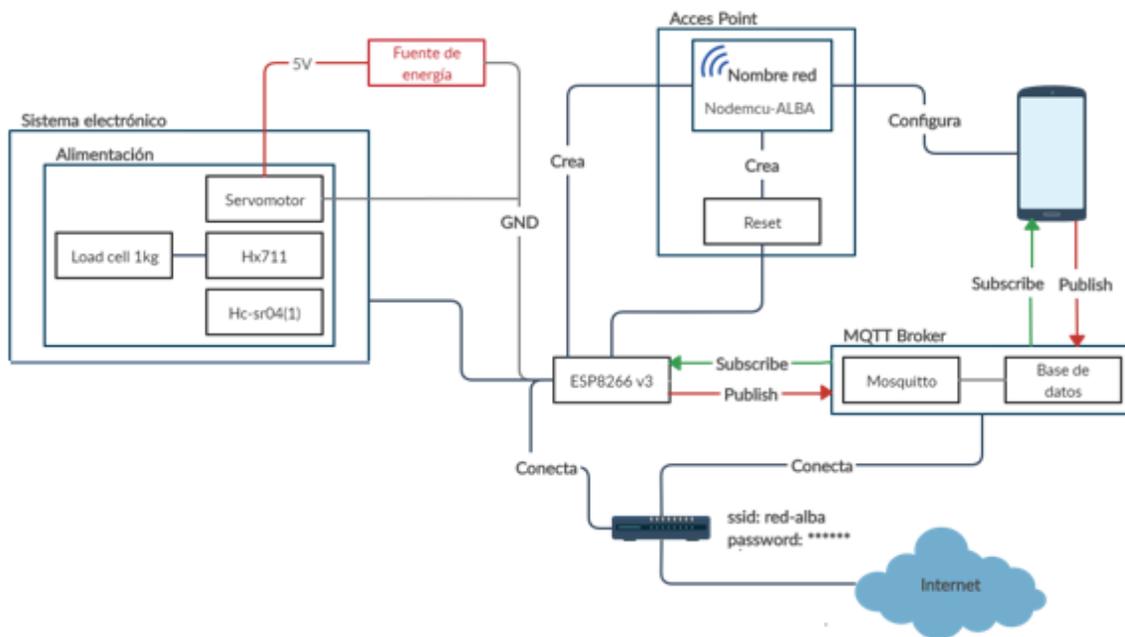


Figura 23. Diagrama de bloques esquema general.

Fuente: Elaboración propia

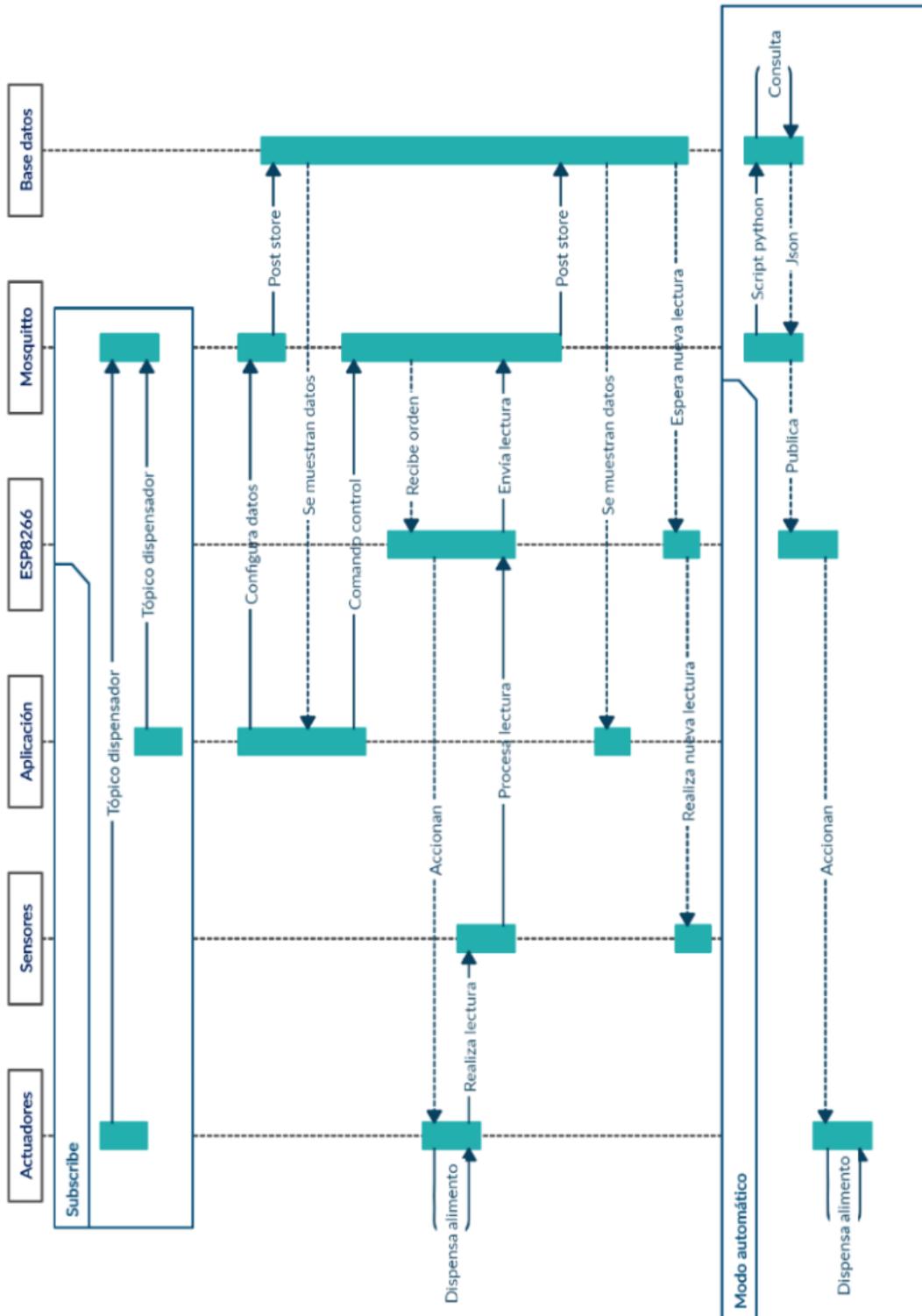


Figura 24. Diagrama de secuencia general.

Fuente: Elaboración propia

3.2.1.1. Broker MQTT

El broker Raspberry pi ejerce como el nodo central, es decir, enviando las órdenes necesarias al módulo NodeMCU ESP8266 de forma que los componentes conformados del sistema de circuito, efectúan las tareas correspondientes provenientes de la aplicación móvil.

3.2.1.2. Nodo lector

Este módulo realiza el procesamiento de la información recibida por el sensor ultrasónico, sin embargo, sólo la información obtenida del sensor Hc-sr04 se emitirá a través de la tecnología WiFi a una base de datos que se encuentra alojada en el servidor (Raspberry pi), esta información podrá ser visualizada desde un dispositivo móvil.

3.2.1.3. Sistema electrónico

El alimentador como se muestra en la *Figura 25* dispone de un servomotor, un Load Cell o una barra de carga de 1kg que se encuentra conectado a un módulo HX711 los cuales al recibir el mensaje desde el broker sobre la cantidad de alimento a dispensar el servomotor girará y abrirá la compuerta, el alimento caerá en la barra de carga y se proseguirá a verificar que la cantidad sea la misma que la recibida si no es, el servomotor girará de nuevo y si es la misma cantidad el sensor Hc-sr04 obtendrá el dato sobre la cantidad de alimento restante en el dispensador y lo enviará al módulo ESP8266 para ser transmitida al broker MQTT.

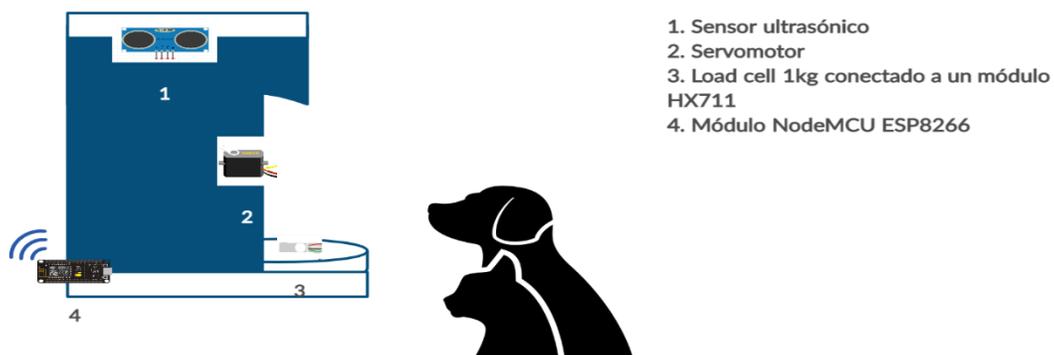


Figura 25. Sistema electrónico del prototipo

Fuente: Elaboración propia

3.2.2. Requerimientos de los sistemas

- Módem proporcionado por Telmex: Arcadyan VRV7006BW22
- HP Pavilion: Procesador AMD A8-7410 APU con AMD Radeon R5 Graphics, y memoria RAM de 8GB con sistema operativo Windows 10.
 - Programa acceso remoto en consola: PuTTY.
 - Programa acceso remoto entorno: TightVNC.
 - Programa para encontrar dirección ip: Advanced Ip Scanner.
- Raspberry Pi 3 Model B
 - Sistema operativo Raspbian Buster.
 - Lenguaje de programación: Python.
 - Editor de texto: GNU Nano.
 - Broker MQTT: Eclipse Mosquitto
 - Servidor web: Apache
 - Sistema gestor de base de datos: MariaDB (MySQL).
 - Lenguaje de programación: PHP.
- NodeMCU, dispone de un chip ESP8266 V3: Al que se conectan un sensor Hc-sr04, un servomotor, un módulo HX711 y una Load cell su instalación se realiza según la información proporcionada por el fabricante.
 - Se programa en el entorno de desarrollo integrado Arduino IDE 1.8.6 (o versión más reciente).
- Diferentes elementos para el circuito: Cables DuPont y protoboard.

3.2.3. Elaboración de las tablas de porciones de alimento

Para que el presente dispensador sea dosificable es importante saber la cantidad de ingesta de alimento para la mascota, esto es importante ya que las mascotas deben ser alimentados en cantidades proporcionales para que reciban la energía y nutrientes necesarios, pero, la cantidad de comida varía de acuerdo al tipo de mascota y sus características: la etapa, su tamaño y en algunos casos como su edad en meses y peso. A continuación, se muestran las porciones de alimento en gramos para cada tipo de mascota siendo en la *Figura 26* para gatos y en la *Figura 27* para los perros; según información obtenida de Purina Excellent Cat Chow [43] y Dog Chow [44].

CANTIDAD DE ALIMENTO

TABLA DE INFORMACIÓN DE PURINA EXCELLENT

De acuerdo a las características de tu mascota

Tabla de información para gatos



Etapa	Tamaño	Edad en meses	Peso	Cantidad (gramos)
Gatito		3 semanas – 2 meses		25 – 45 gramos
		2 meses – 6 meses		45-60 gramos
		6 meses – 1 año		60 – 75 gramos
Adulto	Mediano		2 – 3.5 Kg	35- 60 gramos
	Grande		3.5 – 6.5 Kg	60-110 gramos
Adulto 7+	Mediano		2 – 3.5 Kg	35- 60 gramos
	Grande		3.5 – 6.5 Kg	60-120 gramos

Figura 26. Porciones de alimento para gatos.

Fuente: Elaboración propia

CANTIDAD DE ALIMENTO

TABLA DE INFORMACIÓN DE PURINA EXCELLENT

De acuerdo a las características de tu mascota

Tabla de información para perros



Etapa	Tamaño	Edad (En meses)	Peso (Kg)	Cantidad
Cachorro	Minis	Destete – 3 meses	1- 5 Kg	50- 90 gramos
		3 – 5 meses	1- 5 Kg	60-80 gramos
		5- 8 meses	1- 5 Kg	60- 80 gramos
		8- 12 meses	1- 5 Kg	45- 120 gramos
	Pequeño	3 – 5 meses	1- 5 Kg	50- 70 gramos
		5-8 meses	5-10 Kg	90- 150 gramos
		8- 12 meses	5- 10 Kg	120- 200 gramos
	Mediano		10-25 Kg	200 – 360 gramos
			25 – 35 Kg	360- 450 gramos
	Grande		35 45 Kg	450 – 540 gramos
Gigante		45 Kg +	560 + 30g por cada 5kg adicional	
Adulto	Minis		1- 5 Kg	45 – 120 gramos
	Pequeño		5-10 Kg	120 – 200 gramos
	Mediano		10-25 Kg	200 – 360 gramos
			25 – 35 Kg	260 – 450 gramos
			35 – 45 Kg	450 – 540 gramos
Senior	Gigante		45 Kg +	560 + 30 por cada 5kg adicional
	Minis		1 – 5 Kg	35 – 110 gramos
	Pequeño		5 – 10 Kg	110 – 170 gramos
	Mediano		10 – 25 Kg	170 – 310 gramos
			25 – 35 Kg	310 – 390 gramos
			35 – 45 Kg	390 – 460 gramos
		45 Kg +	460 + 30 g por cada 5Kg adicional	

Figura 27. Porciones de alimento para perros.

Fuente: Elaboración propia

3.3. Configurar Raspberry Pi como Broker MQTT

3.3.1. Instalar sistema operativo

Para la implementación del proyecto se requiere de un sistema operativo, que es Raspbian y es nativo de Raspberry. Raspbian es un sistema operativo gratuito basado en Debian, optimizado para el hardware Raspberry Pi [45]. En la página oficial de Raspberry como se muestra en la *Figura 28* se descarga el archivo .zip *Raspbian Buster with Desktop* el cual permite la interacción con el usuario de manera fácil mediante la interfaz gráfica.

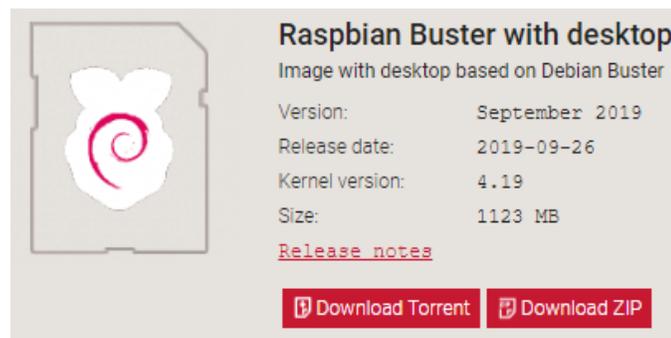


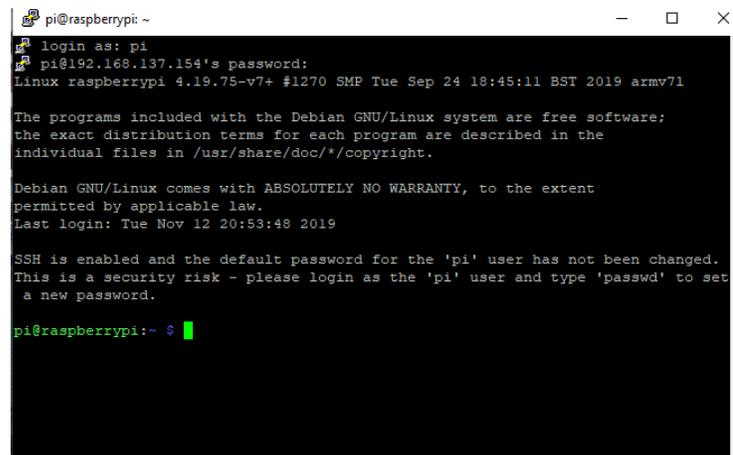
Figura 28. SO utilizado: Raspbian Buster.

Una vez descargado el archivo .zip se prosigue a descomprimir y escribir la imagen del SO en la tarjeta SD, utilizando el software Etcher que borra y sobrescribe en el dispositivo seleccionado (Tarjeta SD).

3.3.2. Configurar conexión y cliente SSH

SSH significa Shell seguro y es un protocolo de comunicación que sirve para acceder y controlar una computadora de manera remota a través de la red, los datos enviados a través de SSH se encuentran encriptados y es uno de los métodos más populares. Para dar continuación, se abre la tarjeta microSD y se habilita la comunicación *SSH* mediante un archivo en blanco con el nombre *ssh* sin extensión para establecer comunicación con la Raspberry, se retira la tarjeta microSD y se inserta en la ranura inferior de la Raspberry Pi. Se prosigue a encontrar la dirección ip del servidor en este caso la raspberry utilizando Advanced IP Scanner el cual puede ser ejecutado en Windows, Mac OS X o Linux.

Una vez obtenida la dirección IP de la Raspberry Pi se requiere instalar un cliente SSH en el ordenador, en este caso se hará uso de *PuTTY* que es un paquete de software gratuito para sistemas operativos Windows lanzado bajo la licencia *MIT*, similar a la licencia *GNU GPL* que usa la mayoría de las distribuciones de *Linux*. *PuTTY* no requiere ninguna configuración de instalación [46]. Se abre el programa *PuTTY* en el cual se ingresa la dirección IP obtenida (se deja las configuraciones por defecto) al realizar la primera conexión se muestra una advertencia de seguridad y se da clic en “Yes”, si la conexión es correcta, se muestra en pantalla indicando ingresar el usuario y la contraseña correspondiente, luego se muestra la pantalla de comandos de la Raspberry como se visualiza en la *Figura 29*.



```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.137.154's password:  
Linux raspberrypi 4.19.75-v7+ #1270 SMP Tue Sep 24 18:45:11 BST 2019 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Tue Nov 12 20:53:48 2019  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set  
a new password.  
  
pi@raspberrypi:~$
```

Figura 29. Cliente SSH: PuTTY.

Fuente: Elaboración propia

Una vez establecida la comunicación se prosigue a activar manualmente el puerto 22 para la comunicación SSH en la Raspberry, la activación se realiza en la terminal al introducir *raspi-config* la cual se encuentra en *Advanced options*. Al realizar la configuración se reinicia el sistema, quedando guardados los cambios efectuados.

Por último, se lleva a cabo la visualización y control del escritorio de la Raspberry en el ordenador. Para lograrlo se hace uso de una conexión *VNC*. *TightVNC* que es un paquete de software de control remoto gratuito con el cual se logrará ver el escritorio de una máquina remota y controlarlo con el mouse y teclado local, de tal modo como se haría sentado frente a una computadora [47]. Se configura el *VNC* en la Raspberry usando la *conexión SSH*

establecida, se realizan las configuraciones para el tamaño de la pantalla y al final obtener el escritorio de la Raspberry como se muestra en la *Figura 30*.

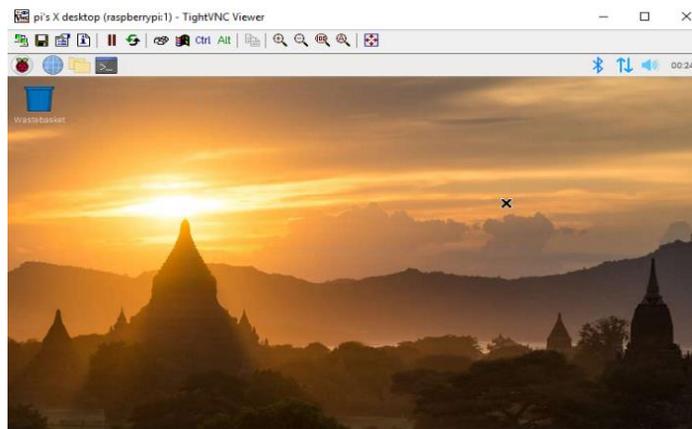


Figura 30. TightVNC: Escritorio Raspberry.

Fuente: Elaboración propia

3.2.3. Configuraciones iniciales en consola

Antes de comenzar con los cambios primero se debe actualizar el sistema con los cambios más reciente con los siguientes comandos: `sudo apt-get update` y `sudo apt-get upgrade`, este proceso puede durar algunos minutos además cabe mencionar que se debe ejecutar estos comandos de vez en cuando para mantener el sistema actualizado. Ahora se prosigue a ejecutar el comando `sudo raspi-config` para acceder a las herramientas de configuración del sistema donde se desplazarán por los distintos menús y submenús como se muestra en la *Figura 31*.

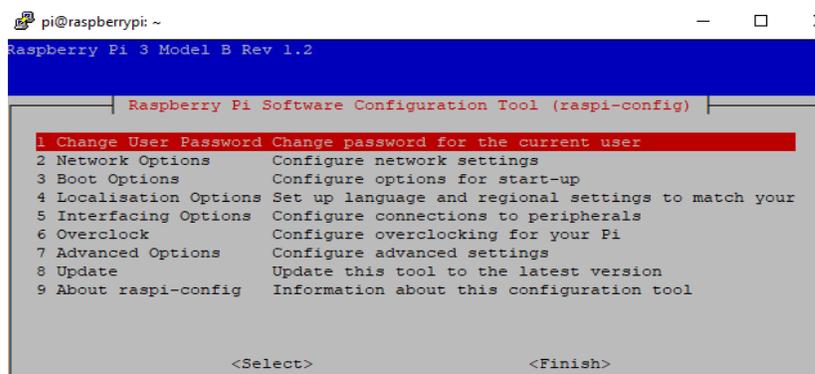


Figura 31. Configuraciones con raspi-config.

Fuente: Elaboración propia

- Se debe configurar el idioma y la zona horaria para el sistema operativo, lo cual además de cambiar la presentación de números, moneda, fecha, la ayuda y las aplicaciones conforme al estándar del idioma seleccionado según la ubicación geográfica para esto se selecciona *Localisation Options* y luego *change timezone* donde se selecciona el área geográfica *América* y, por último, se indica la ciudad o región correspondiente y en este caso se elige México-City.
- La siguiente configuración es la conexión Wi-Fi de la tarjeta donde se selecciona *Network Options* se especifica el país y se configura el *SSID* de la red y la *contraseña* correspondientes a la red que se desea conectar de esta forma el DHCP del router asignará una IP.

3.2.4. Configurar IP estática

La dirección IP obtenida de la Raspberry Pi es una dirección asignada dinámicamente dentro de un rango de direcciones proporcionada por el DHCP (Dynamic Host Configuration Protocol) siendo en este caso el router, presentando ser una desventaja para aplicaciones permanentes siendo la mejor opción disponer de una IP estática.

Para configurar una IP estática en la Raspberry Pi se debe acceder al fichero *dhcpcd.conf* que se muestra en la *Figura 32* y se accede con el comando *sudo nano /etc/dhcpcd.conf*. Al ser una aplicación para uso de red domótica de una vivienda pequeña se hace uso de la red de clase C la cual incluye los siguientes parámetros:

- Interface: Referencia al nombre de la interface de red ya sea *Ethernet*, *WLAN* y *WWAN*. Cabe mencionar que en ocasiones la interfaz de red Ethernet se puede mostrar como *enp3s0* o *eth0*, *eth1*, etc. Del mismo modo, la interfaz de red inalámbrica como *wlp1s0* o *wlan*, *wlan0*, etc, para este caso se hace uso de la interfaz de red inalámbrica.
- Static ip_address: Dirección fija a utilizar, la cual se recomienda que se encuentra fuera del segmento que otorga el DHCP.
- Static routers= Dirección de la puerta de enlace, *Gateway*, es donde se indica cual es la entrada al router. Por lo general son, 192.168.1.1 o 192.168.1.254

- Static domain_name_servers= Dirección del servidor *DNS* normalmente del router da la posibilidad de traducir las direcciones de dominios en sus direcciones IP a la hora de navegar por internet el *DNS* más utilizado es el *Google Public 8.8.8.8* y *8.8.4.4* que se encargar de optimizar el rendimiento y velocidad de navegación.

```

pi@raspberrypi: ~
GNU nano 3.2 /etc/dhcpd.conf Modified
#slaac hwaddr
# OR generate Stable Private IPv6 Addresses based from the DUID
slaac private

# Example static IP configuration:
#interface eth0
#static ip_address=192.168.0.10/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1

#EXAMPLE STATIC IP CONFIGURATION
interface wlan0
static ip_Address=192.168.1.70
static routers=192.168.1.1
static domain_name_servers=192.168.1.1

```

Figura 32. Configuración archivo /etc/dhcpd.conf.

Fuente: Elaboración propia

3.2.5. Implementación del broker MQTT y clientes

MQTT es un protocolo de conectividad de máquina a máquina (M2M) e IoT. Es de mensajería liviano que funciona con un mecanismo de publicación/suscripción basado en intermediario y se ejecuta sobre el protocolo de control de transmisión/protocolo de Internet (TCP/IP). Diseñado para admitir el transporte de mensajes desde dispositivos remotos que involucran poco código, bajo consumo de energía, bajo ancho de banda, alto costo de conexiones, alta latencia y disponibilidad variable; siendo muy eficientes y principalmente orientado en sensores y en una gran variedad de escenarios de automatización domésticas, por lo tanto, es la solución más apropiada para las aplicaciones de Internet de las cosas.

Para ejecutar el protocolo MQTT en la Raspberry Pi se necesita de un servidor que es el encargado de gestionar la red y redirigir los mensajes a los clientes correspondientes, para ejecutar el protocolo como si fuera un servidor, en este proyecto se hace uso de Mosquitto. Mosquitto es un broker Open-Source (código abierto) es un intermediario de mensajes que permite la conexión de dispositivos MQTT, es parte del proyecto IoT de Eclipse Foundation y se encuentra incluido en los repositorios de Raspbian, Mosquitto utiliza el puerto por defecto 1883.

- El primer paso para establecer la comunicación es descargar la signing key firma de conformidad mediante el comando `wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key` y la clave para autenticar el paquete `apt-key add mosquitto-repo.gpg.key`
- Se accede a la carpeta `sources.list.d` con el comando `cd` y se descarga la lista de repositorios de *Mosquitto* con `wget` y se instala la última versión del sistema operativo Raspbian dependiendo de la versión se digita el siguiente comando `wget http://repo.mosquitto.org/debian/mosquitto-buster.list`.
- Luego, se instala el servidor Mosquitto ejecutando el comando `apt-get install mosquitto`.

Una vez configurado el servidor Mosquitto se procede a instalar con el comando `apt-get install mosquitto-clients` que es una biblioteca que permite implementar clientes MQTT. Las funciones de los clientes más populares para cada rol en la conexión son:

Mosquitto_sub: Permite suscribirnos a un Topic al escuchar y mostrar los mensajes enviados. A continuación, se muestra el código utilizado para el cliente suscrito:

```
pi@raspberrypi: ~ $ mosquitto_sub -h localhost -t Casa/Sala/Dispensador
```

Mosquitto_pub: Permite publicar mensajes esto quiere decir, que publicará un mensaje en el Tópico al cual se encuentra suscrito. A continuación, se muestra el código utilizado para el cliente publicador:

```
pi@raspberrypi: ~ $ mosquitto_pub -h localhost -t Casa/Sala/Dispensador -m "Alimento: 60%".
```

Donde los parámetros utilizados para ambos clientes MQTT son:

- `-h`: Indica que lo siguiente a escribir es el host.
- `BROKER`: Se refiere a la dirección ip de Mosquitto obtenida de la red o por defecto localhost.
- `-t`: Indica que lo siguiente a escribir es el tópico al cual enviaremos mensaje.
- `TOPIC`: Nombre del tópico.
- `-m`: Indica que lo siguiente a escribir es el mensaje.
- `MENSAJE`: Como lo dice, es el mensaje a enviar y tiene que ir entre comillas dobles.

3.2.6. Arquitectura de un sistema MQTT

En la *Figura 33* se muestra que el protocolo MQTT utiliza una topología de estrella, es decir, un nodo central que funciona como servidor y es el encargado de gestionar la red y de transmitir los mensajes, los clientes mandan periódicamente un paquete y esperan la respuesta del broker (todos los mensajes pasan por él). Se basa en el patrón de publicación / suscripción, tiene tres tipos de actores:

- **Publicador:** La función del editor es conectarse al intermediario de mensajes y publicar contenido.
- **Suscriptor:** Se conectan al mismo agente de mensajes y se suscriben al contenido que les interesa.
- **Intermediario de mensajes:** Asegura que el contenido publicado se transmita a los suscriptores interesados.

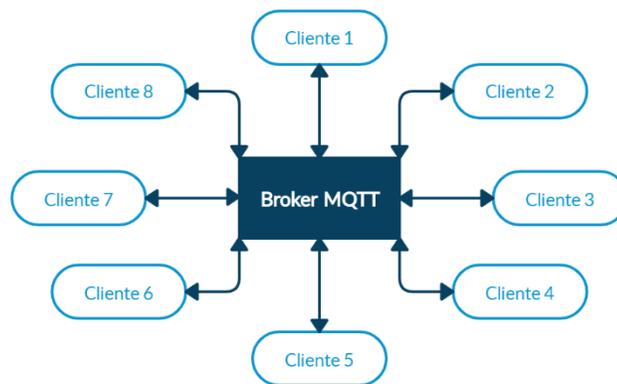


Figura 33. Topología MQTT.

Fuente: Elaboración propia

Como se muestra en la *Figura 34* la publicación/suscripción de MQTT es una arquitectura basada en eventos. Un cliente puede publicar un mensaje sobre un tópico, o puede suscribirse a un tópico en específico para poder recibir mensajes en donde llegan al broker y se encarga de repartir los mensajes a los clientes correspondientes. Por ejemplo:

- El módulo publica “60%” de alimento restante en el tópico Casa/Sala/Dispensador.
- El dispositivo móvil o cliente debe estar suscrito en el mismo tópico en el que publica el módulo: Casa/Sala/Dispensador.
- De tal manera, el dispositivo móvil recibe el mensaje “60%” de alimento restante.

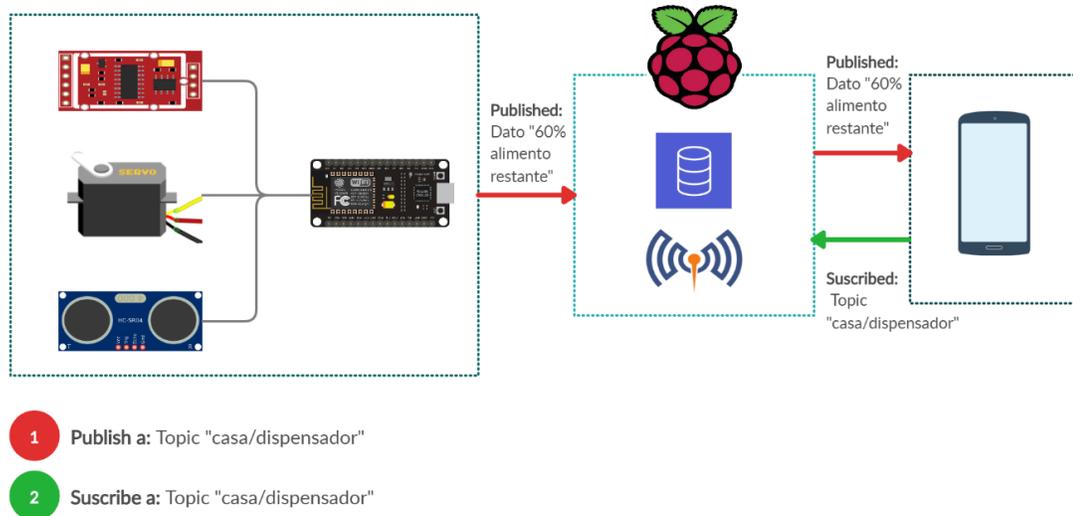


Figura 34. Publicación/Suscripción MQTT.

Fuente: Elaboración propia

3.2.7. Filtrar mensajes con base al Topic

El Broker MQTT aplica un filtro a los mensajes que son recibidos desde los publicadores, para discriminar a que clientes suscritos es entregado. A este filtrado se le denomina *Topic*, consiste en una cadena de texto UTF-8 y se distingue entre mayúsculas y minúsculas. El funcionamiento de los Topics en MQTT:

- El Broker acepta todos los Topics y cabe mencionar que, no es necesario crearlos explícitamente antes de publicar o suscribirse.
- Los clientes pueden suscribirse a uno o varios Topic.
- Los clientes publican mensajes indicando un único Topic.

En la *Figura 35* se muestra que los Topic están formados por uno o más niveles separados entre sí por una barra inclinada '/':

Casa/Sala/Luces
 Casa/Sala/Dispensador
 Casa/Jardin/Riego

Figura 35. Ejemplo de estructura de un Topic.

Fuente: Elaboración propia

Otra de las formas de establecer una suscripción a un Topic es empleando los Wildcards:

- Nivel único: En la *Figura 36* se muestra que el comodín + puede ser empleado para sustituir a un único nivel en cualquier lugar del Topic.



Figura 36. Wildcards nivel único.

Fuente: Elaboración propia

- Nivel múltiple: En la *Figura 37* se muestra que el comodín # puede ser empleado para sustituir cualquier número de niveles y ser usado únicamente al final del Topic.

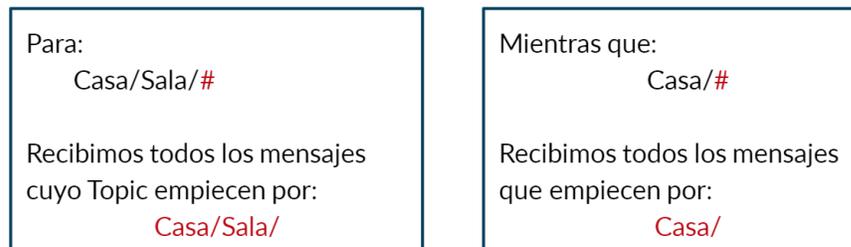


Figura 37. Wildcards nivel múltiple.

Fuente: Elaboración propia

3.2.8. Publicar desde el broker con cliente Python MQTT

Hasta este punto se realizan publicaciones con el cliente MQTT: *mosquitto_pub*, sin embargo, es necesario para el desarrollo de este proyecto hacerlo de manera programática siendo lo mejor utilizar la librería para Python. El proyecto Eclipse Paho provee clientes y librerías Open source para MQTT en lenguaje de Python, la descarga de esta librería es con el comando *sudo pip install paho-mqtt*. A continuación, se indica el uso de esta librería:

1. Se importa la clase cliente: *import paho.mqtt.client as mqtt*.

2. Se crea una instancia del cliente que toma cuatro parámetros opcionales siendo: `Client(client_id="", clean_session=True, userdata=None, protocolo=MQTTv311, transport="tcp")`. El `client_id` es una cadena única otorgada por el cliente cuando se conecta un intermediario, sino se proporciona una, el agente asigna el cliente. Al finalizar, para crear la instancia se realiza: `client=mqtt.Client(client_disp)`
3. Se establece la conexión a un intermediario que es el broker, usando el método de conexión del cliente Python MQTT: `connect(host, port=1883)` el parámetro `host` indica el nombre o la dirección del servidor, en este caso la Raspberry y el puerto es el que utiliza Mosquitto, al finalizar su sintaxis es: `connect(192.168.10, port=1883)`
4. Por último, se realizan las publicaciones, utilizando el método de publicación: `publish("casa/sala/dispensador", payload=1)` donde el `payload` consiste en el mensaje a publicar.

3.2.9. Automatización del dispensador

La dispensación del alimento es a disposición del usuario, lo cual se realiza en la configuración correspondiente desde la aplicación móvil, para que el módulo dispensador efectúe su tarea se debe guardar la información en la base de datos que se encuentra en el broker que es la Raspberry Pi. El servidor realiza una consulta a la base de datos donde se encuentra programado la dispensación por medio de un Script en Python, estos datos se almacenan en un archivo Json y posteriormente, se analizan para así efectuar la publicación MQTT de dispensar el alimento. Lo descrito con anterioridad se representa en la *Figura 38*.

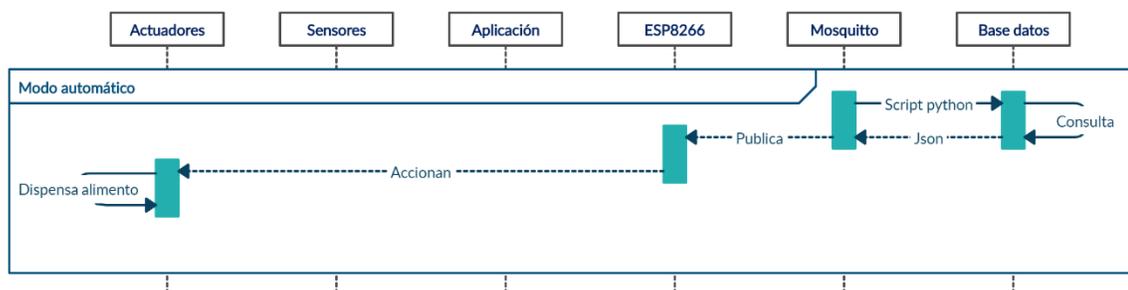


Figura 38. Diagrama de secuencia modo automático.

Fuente: Elaboración propia

3.2.10. Instalación y configuración del servidor LAMP

LAMP se compone de cuatro componentes: *Linux* siendo el sistema operativo, *Apache* es el servidor web, *MySQL/MariaDB* es una base de datos y *PHP* lenguaje de script del lado del servidor. Para proseguir con la instalación y configuración, se realiza la conexión *SSH* de la *Raspberry Pi*, y una vez en la línea de comandos en el cual debe tener privilegios de *root* deberá agregar *sudo* al comienzo del comando o usar *su i-* para cambiar a usuario *root*.

Instalación de la base de datos

Este sistema utiliza una gestión de base de datos *MySQL* (relacional). La base de datos fue diseñada para almacenar la información correspondiente a la gestión de almacén y venta de dispensadores, información de los usuarios, registro de la mascota y registro de los dispensadores además de guardar el nivel captado por el sensor *hc-sr04*.

- Para realizar la instalación de la base de datos y otros paquetes se ejecuta el comando `apt install mariadb-server mariadb-client`.
- Después que se instale el servidor *MariaDB* se verifica su estado con el comando `systemctl status mariadb`
- Permitir que *MariaDB* se inicie automáticamente en el momento de arranque con el comando `systemctl enable mariadb` y posteriormente ejecutar el script de seguridad con `mysql_secure_installation`.

Creación de la base de datos

Para continuar con la instalación se prosigue a la configuración y creación de la base de datos con el usuario que se le permite el acceso a la base de datos por seguridad para guardar la información del dispensador, el nivel de alimento restante y la mascota.

- Con el cliente *mysql* accedemos a la base de datos con el usuario *root* y una contraseña con `mysql -u root -p`
- Crear la base de datos con el comando `mysql> CREATE DATABASE dispensador;`

Creación y configuración de los usuarios

- Crear dos usuarios para acceder a dicha base de datos siendo el mismo nombre, pero con diferentes *hosts* de esta forma permite la conexión con otra dirección `mysql>`

```
CREATE USER 'pethome-dtil'@'localhost' IDENTIFIED BY 'usuariopethomedtil' y al usuario mysql> CREATE USER 'pethome-dtil'@'%' IDENTIFIED BY 'usuariopethomedtil'
```

- Por último, se otorgan los permisos para acceder a la base de datos, en este caso se le concede todos los permisos y se indica al host a conectarse con mysql> `GRANT ALL ON dispensador. * TO 'pethome-dtil'@'localhost';` finalizando con la configuración de la base de datos, cerrando con *quit*.

Instalación de PHP

- La instalación de php se realiza con el siguiente comando `apt install php7.2 libapache2-mod-php7.2 php7.2-mysql php-common php7.2-cli php7.2-common php7.2-json php7.2-opcache php7.2-readline`
- Se habilita el módulo Apache y reinicia el servidor web Apache con `a2enmod php7.2` y con `systemctl restart apache2`
- Para comprobar funcionamiento del servidor Apache se crea un archivo en el directorio raíz web `nano /var/www/html/info.php` en donde se escribe `<? php phpinfo();?>` y posteriormente se abre el navegador y en la barra de direcciones se ingresa la *dirección-ip-del-servidor/info.php* o *localhost/info.php*. Visualizando una página con la información de PHP. Pero, para la seguridad del servidor se debe eliminar el archivo creado con `rm /var/www/html/info.php`

3.4. Configuración de cliente y port forwarding para acceso remoto

Después de contar con una dirección IP estática a la Raspberry hasta este punto se puede hacer uso del proyecto de manera local sin poder acceder o controlarlo desde el exterior una de las formas para poder resolver este problema es utilizando un servicio *DNS* que facilita el tener un nombre de dominio asignado a una dirección Ip pública estática al asociar los puertos al router de la casa y así poder acceder de manera exterior al proyecto.

A continuación, se hará uso de *NO-IP* que provee una dirección Ip pública dinámica y permite el registro de un dominio público que puede ser gratis o de pago. Para el desarrollo de este proyecto se hizo el registro en el siguiente link <https://www.noip.com/> de forma gratuita pulsaremos en *Add Hostname* para seleccionar un nombre y una extensión de dominio aceptando la que se encuentre disponible, como se visualiza en la *Figura 39*.



Hostname	Last Update	IP / Target	Type	
albapethome.ddns.net Expires in 26 days	Feb 12, 2020 06:31 PST	201.105.191.169	A	Modify

Figura 39. Configuración NO-IP.

Fuente: Elaboración propia

Una vez creado el dominio en la página web, se prosigue a configurar el cliente NO-IP en la Raspberry antes de instalarlo, se crea un directorio en home con `mkdir noip`, se accede a la carpeta y se descarga el cliente con el comando `wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz` al finalizar la descarga se descomprime el contenido `tar -zxvf noip-duc-linux.tar.gz` y se accede a la carpeta `cd noip-2.1.9-1` (en caso de haber un cambio en la versión se tecldea `ls` y se busca el nombre de las carpetas y archivos y se sustituye), una vez dentro de la carpeta como se muestra en la *Figura 40* se procede a la instalación con `sudo make` y `sudo make install` este último preguntará el correo electrónico y la contraseña utilizado anteriormente en la página web de No-IP de forma que tomará el nombre de dominio creado `albapethome.ddns.net`. Hasta este punto el cliente NO-IP en la Raspberry se encuentra ya configurado.

```

pi@raspberrypi:~/noip/noip-2.1.9-1 $ sudo make install
if [ ! -d /usr/local/bin ]; then mkdir -p /usr/local/bin;fi
if [ ! -d /usr/local/etc ]; then mkdir -p /usr/local/etc;fi
cp noip2 /usr/local/bin/noip2
/usr/local/bin/noip2 -C -c /tmp/no-ip2.conf

Auto configuration for Linux client of no-ip.com.

Multiple network devices have been detected.

Please select the Internet interface from this list.

By typing the number associated with it.
0 eth0
1 wlan0
wlan0
Please enter the login/email string for no-ip.com pethome.alba@gmail.com
Please enter the password for user 'pethome.alba@gmail.com' *****
***

Only one host [albatpethome.ddns.net] is registered to this account.
It will be used.
Please enter an update interval:[30]
Do you wish to run something at successful update?[N] (y/N) y
Please enter the script/program name update

New configuration file '/tmp/no-ip2.conf' created.
mv /tmp/no-ip2.conf /usr/local/etc/no-ip2.conf

```

Figura 40. Configuración cliente NO-IP en servidor.

Fuente: Elaboración propia

Posteriormente se indica que el sistema ejecute el script en cada reinicio para ello se crea un fichero en el directorio `init.d` con `nano /etc/init.d/noip2` en donde se escribe lo siguiente `/usr/local/bin/noip2`, se establecen los permisos de ejecución `chmod +x /etc/init.d/noip2` y se modifica el fichero de actualización del servicio `update-rc.d noip2 defaults` para ponerlo en marcha.

A continuación, se abren los puertos necesarios del router para ellos se accede al router normalmente son las siguientes direcciones `192.168.1.254` o `192.168.1.1`, se agrega el host o nombre de dominio creado, la dirección ip del servidor en el apartado `DNS & DDNS`, y, por último, se realiza port forwarding como se visualiza en la *Figura 41* de los puertos 22 (`SSH`), 80 (`HTTP`) y 5900/5910(`VNC`) hacia la dirección de la Raspberry pi permitiendo acceso remoto a través del dominio que se creó.

No.	Descripción	IP LAN	Tipo de protocolo	Puerto LAN	Puerto publico	Habilitar	Modificar/Borrar
1	raspi.ssh	192.168.1.10(raspberrypi)	TCP&UDP	22	22	Deshabilitado	<input type="button" value="Modificar"/> <input type="button" value="Borrar"/>
2	raspi.vnc	192.168.1.10(raspberrypi)	TCP&UDP	5900	5910	Deshabilitado	<input type="button" value="Modificar"/> <input type="button" value="Borrar"/>
3	raspi.http	192.168.1.10(raspberrypi)	TCP&UDP	80	80	Deshabilitado	<input type="button" value="Modificar"/> <input type="button" value="Borrar"/>
<input type="button" value="Agregar"/>							

Figura 41. port forwarding

Fuente: Elaboración propia

3.5. Configurar NodeMCU ESP8266 como nodo lector

La *Figura 42* se muestra el NodeMCU que es una placa de desarrollo basada en el módulo ESP8266. En la que se incorpora un chip SoC (System on a Chip) que tiene un microcontrolador (MCU), su mayor ventaja es el incorporar un módulo WiFi permitiendo el crear o desarrollar proyectos conectados para el IoT o sistemas inalámbricos por ser eficientes, bajo consumo de energía y bajo costo.

Para este proyecto se utiliza el módulo ESP8266 v3 la cual es básicamente la v2 y pertenece a la 2da generación con algunas mejoras, aunque no es una nueva especificación oficial. A continuación, se mencionan sus características y mejoras:

- 13 pines digitales (D0 al D12).
- Un pin analógico (A0).
- Tres pines de 3.3V.
- Cinco pines GND
- Dos leds integrados uno asociado al pin D0 y D4.
- Puerto USB más robusto
- Dos pines que no se utilizan en la versión V2 se han utilizado como salida de 5V directas del USB y un GND adicional.

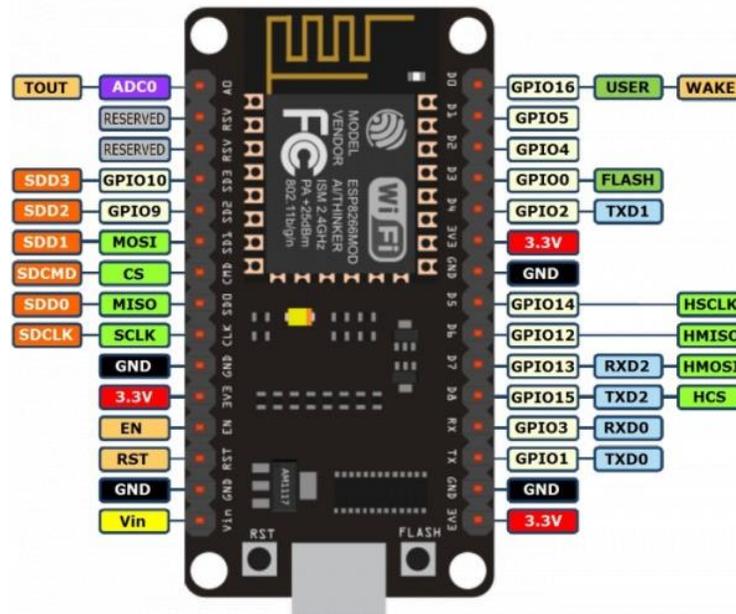


Figura 42. Especificaciones del módulo ESP8266 V3.

3.5.1. Preparación del entorno Arduino IDE

Antes de empezar a trabajar con el IDE se debe saber que el NodeMCU integra al ESP8266 y Arduino lo soporta, pero, no viene incorporada en él, por lo tanto, hay que descargar e incluir la tarjeta en el entorno. Algo que se debe saber es que se debe tener actualizado el IDE en su versión 1.6.4 o más reciente para evitar inconvenientes como al tratar de instalar el Pluggin ESP8266, posteriormente se deben realizar algunas configuraciones en el IDE:

- En el gestor de URLs adicionales de tarjetas en *Preferencias* se incluye la dirección `http://arduino.esp8266.com/stable/package_esp8266com_index.json` para el módulo Json.
- Es necesario descargar desde *Gestor de Tarjetas* el soporte para la tarjeta ESP8266 o desde la plataforma de GitHub se descarga él .zip y se coloca en las librerías de Arduino.
- Una vez instalado el soporte para el módulo, se verifica que se encuentre la placa a utilizar en este caso es “*NodeMCU ESP8266 V3*” y se realizan las siguientes configuraciones que se encuentran en el *datasheet* para que la placa se encuentre óptima al ser programada como se muestra en la *Figura 43*:

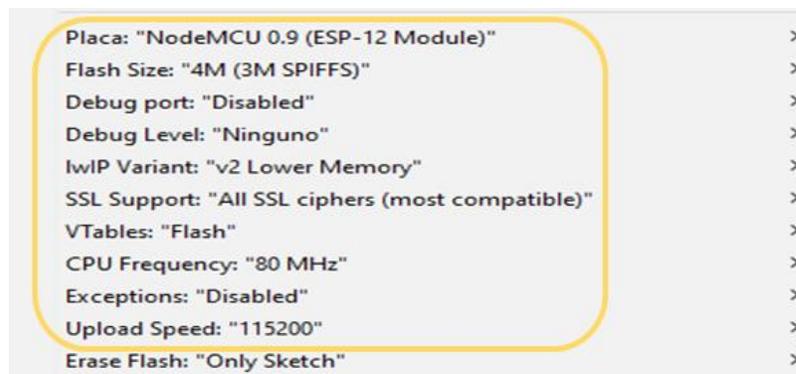


Figura 43. Configuraciones para la placa NodeMCU ESP8266

Fuente: Elaboración propia

3.5.2. Configuración de red WiFi

Para lograr que el módulo ESP8266 se conecte a una red WiFi se puede configurar de dos formas, siendo la primera como una estación, es decir, Modo estación STA y conectarlo a la red WiFi y la segunda como un punto de acceso conocido como Access Point (AP), estableciendo su propia red WiFi. La conexión a una red WiFi por modo estación STA, se

logra por las funciones que brinda la librería *ESP8266WiFi*, que se descarga junto con el gestor de placas al configurar el IDE Arduino como se mostró en el apartado anterior 3.5.1. *Preparación del entorno de Arduino*. Esto se logra facilitando el SSID que es el nombre de la red WiFi (máximo 32 caracteres), una contraseña siendo como mínimo 8 y máximo 64 caracteres. Pero, uno de los problemas que se presenta al hacer uso del modo estación STA es, la desconexión de red ya que el módulo no logra encontrar el SSID correspondiente al código. Para solucionar este inconveniente, se utiliza el modo *Access Point* o conocido como *AP* que proporciona acceso a la red WiFi a otros dispositivos y así conectar el módulo mediante un teléfono móvil, Tablet, etc, haciendo uso de la librería *WiFiManager*. Esta librería permite configurar una nueva red WiFi sin cargar nuevo código, además de ser una librería de código abierto que se puede encontrar en GitHub. En la *Figura 44* se muestra como crear un *AP*, activa un servidor DNS y un servidor web nombrado: “*Nodemcu-ALBA*”.

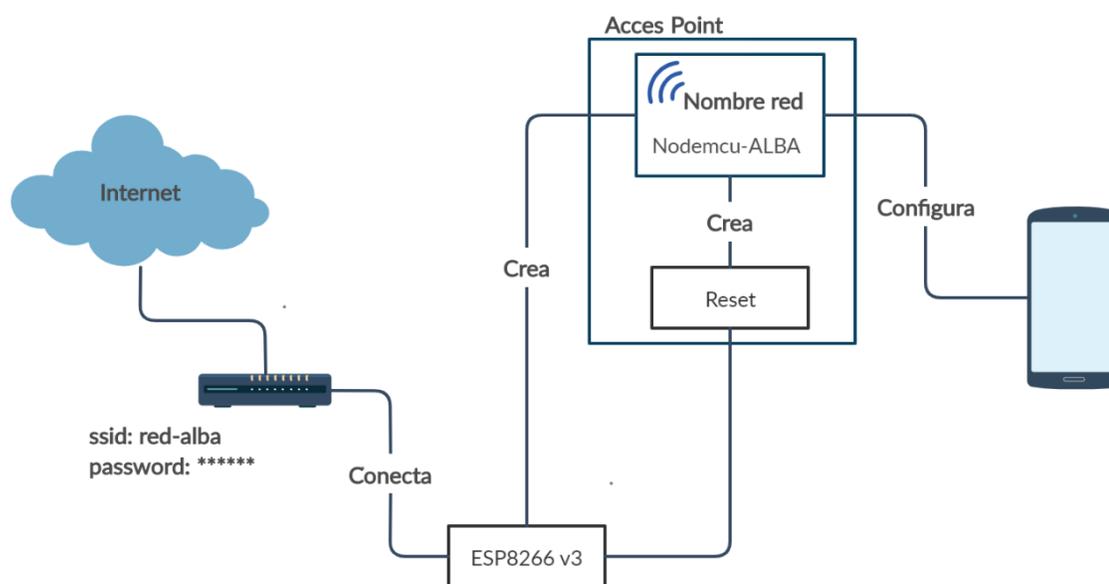


Figura 44. Diagrama de bloques conexión WiFi-Manager.

Fuente: Elaboración propia

Esto quiere decir que desde cualquier dispositivo se hace una conexión al AP y por medio de un portal cautivo muestra una ventana emergente, la cual permite realizar la configuración a la red disponible, se debe agregar que al hacer uso de la librería Ticker que ejerce como temporizador permitirá saber el estado de conexión del módulo ESP8266, dicho de otra manera, saber si está conectado o desconectado. La función Ticker se relaciona con el led

Pin D4 que se encuentra integrado al módulo. Siendo que, al principio el led parpadea no sólo cuando este en modo AP sino también con conexión fallida, por otra parte, si el pin se encuentra encendido significa que la conexión fue exitosa. El módulo ESP8266 tiene como característica almacenar las credenciales de la última conexión WiFi ingresada así que en caso de haber cambiado de SSID, se cuenta con la opción *RESET* de tal forma que se tiene la oportunidad de configurar la red para conectar al módulo ESP8266; lo cual se logra con la siguiente línea de código `wifiManager.resetSettings()`. Hay que mencionar, además que la librería WiFiManager fue la primera librería en aparecer en el panorama del ESP8266, por lo tanto, es la más popular. A continuación, se muestra en la *Figura 45* en resumen la funcionalidad que lleva a cabo esta librería.

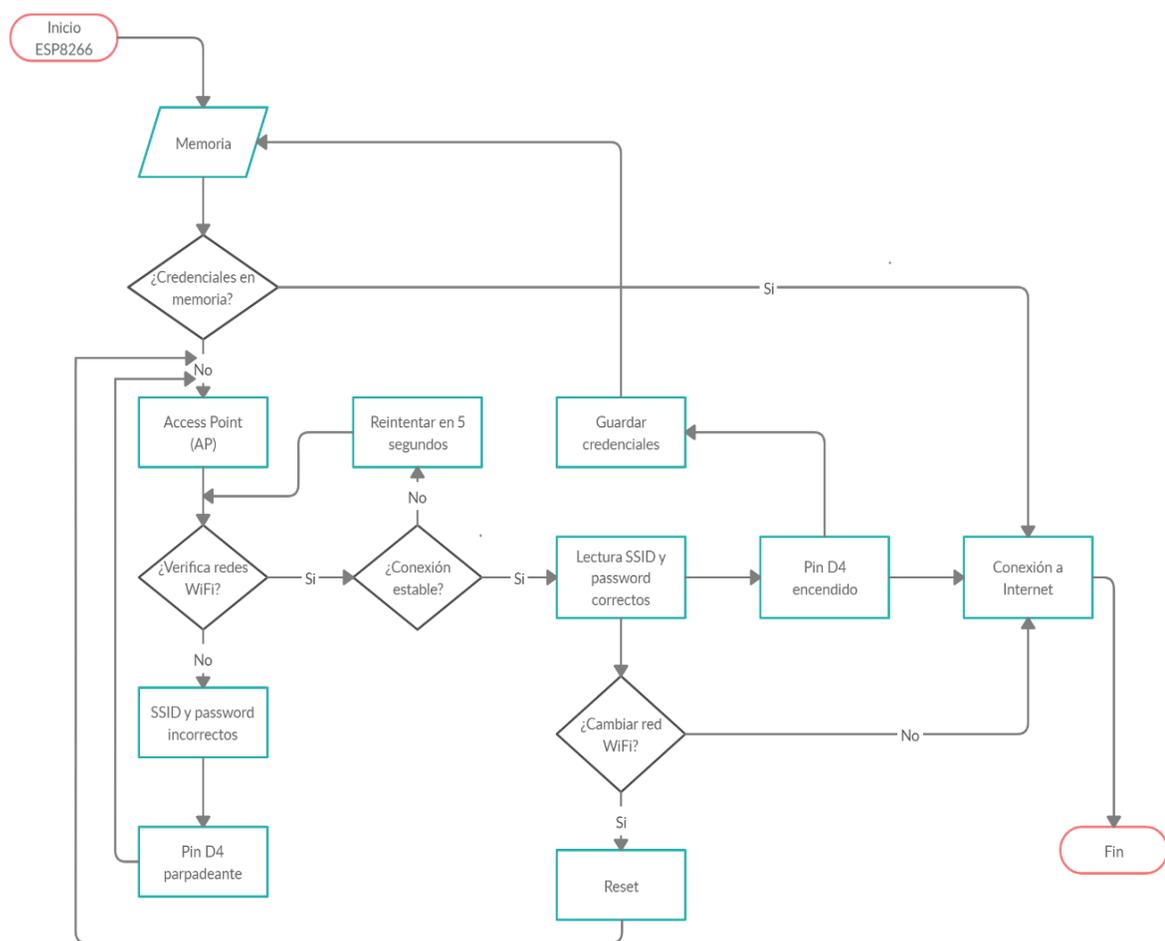


Figura 45. Diagrama de flujo WiFiManager.

Fuente: Elaboración propia

3.5.3. Implementación de MQTT

Se hace uso la librería *Arduino Client for MQTT*, para conseguir una comunicación MQTT. Dicha librería, se publica bajo licencia MIT y provee de un cliente que permite tanto la suscripción como publicación de contenidos con un servidor que admite MQTT. La biblioteca utiliza una API Arduino Ethernet Client para interactuar con el hardware de red, lo que hace compatible con algunas shield y placas como: Arduino Ethernet y Arduino Ethernet Shield, Arduino Yun, Arduino WiFi Shield, Spark fun WiFly Shield, TI CC3000 WiFi, Intel Galileo/Edison, ESP8266 y ESP32.

A continuación, se mostrarán partes relevantes de funciones necesarios, sin exponer el todo el código. El lenguaje Arduino se caracteriza por tener tres partes diferentes:

1. En la primera parte se define el uso de las librerías, constantes, variables y funciones en este caso para la librería *Arduino Client for MQTT* se define el puerto de broker el cual puede ser por default 1883 o el puerto TLS/SSL 8883 que requiere de un certificado para su uso, los tópicos *publish* y *subscribe* utilizados por el proyecto, así como la dirección del broker, es decir, la dirección de la raspberry.
2. Un `setup()` que se encarga del arranque del dispositivo en este caso se hace referencia a la conectividad con la librería *WiFiManager* que anteriormente se explicó así como se realizó la conexión a Wi-Fi del módulo se realiza lo mismo con la conexión al broker donde se indica las credenciales del servidor que son la dirección del broker y el puerto con `client.setServer(MQTT_SERVER, MQTT_PORT);` así mismo se establece un *callback* que hace referencia para recibir el mensaje a la función anteriormente definida `client.setCallback(MQTTCallback);`.
3. Por último, un `loop()` donde se ejecuta el código correspondiente de manera recurrente. Como se debe obtener la cantidad de alimento obtenido del sensor del dispensador de alimento se hace mediante el método `initRemainingFood();` una vez obtenido el dato del sensor se envía al sistema para que pueda visualizarse en el dispositivo del cliente mediante el método *subscribe* donde se especifica el tópico correspondiente al nivel de alimento mediante `client.subscribe(FOOD_TOPIC_SUBSCRIBE);`

De manera general el uso de esta librería en el proyecto, permite que el módulo ESP8266 se encontrará esperando de un mensaje que se recibe del tópico al cual se encuentra suscrito para accionar los actuadores y dispensar el alimento además de publicar la cantidad del

alimento leído por el sensor hc-sr04 el servidor lo procesa y posteriormente se almacena en la base de datos MySQL, como se muestra en la *Figura 46*.

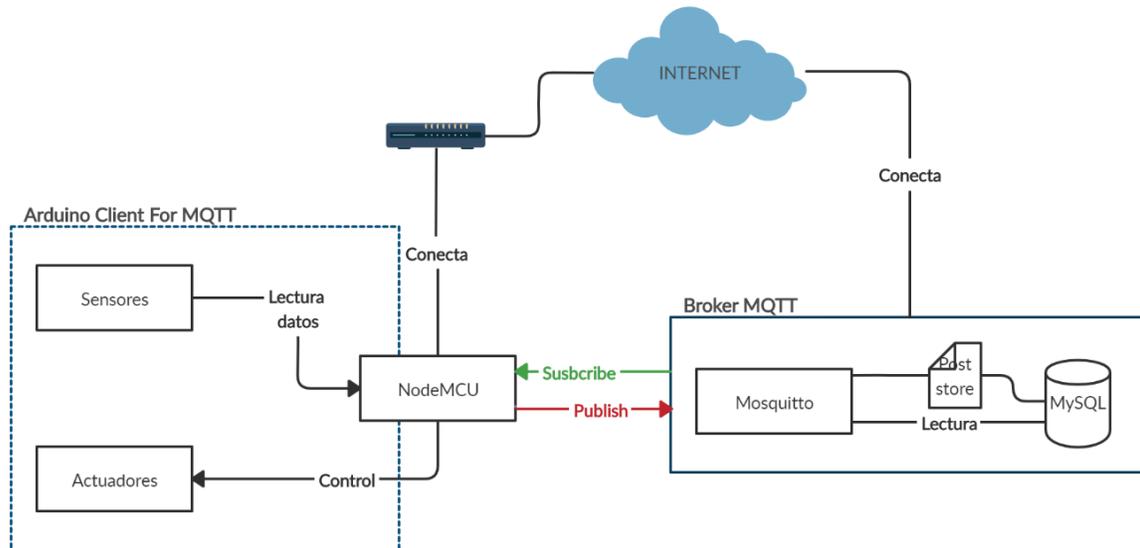


Figura 46. Diagrama de implementación MQTT en ESP8266.

Fuente: Elaboración propia

3.5.4. Conexión de los componentes

El módulo ESP8266 v3 cuenta con 13 pines digitales que van del D0 al D12, cabe destacar que se encuentran dos nomenclaturas para nombrar los pines, una que está escrita en la placa y la otra en ocasiones asociado al GPIOx. El módulo ESP8266 no cuenta con memoria flash, por lo tanto, necesita de una memoria externa donde se almacenan los programas y datos, para conectarse a ella, se necesita utilizar pines propios como resultado, algunos de los pines del ESP8266 no pueden ser utilizados estos son GPIO6, GPIO7, GPIO8, GPIO9, GPIO10 y GPIO11. Los pines GPIO9 y GPIO10 son conocidos como SD2 y SD3 y no son recomendables para conectar componentes y los pines GPIO3 y GPIO1 son para recibir (RX) y transmitir (TX) ya que son utilizados para transmitir un sketch o para la comunicación a través del puerto serie y al ser utilizados ocasionarían interferencias al cargar el sketch. A continuación, en la *Figura 47. Esquemático del circuito elaborada en Fritzing* se observa las conexiones de los componentes al módulo ESP8266 V3, conformado por:

- Un sensor ultrasónico HC-SR04 que sirve para obtener e indicar el nivel de alimento en el contenedor del dispensador.

- El servomotor que accionará y abrirá la compuerta para la dispensación del alimento, se encontrará alimentado por el pin de 5V el cual se suministra por una fuente de energía externa siendo que el módulo ESP8266 no le suministra el voltaje necesario para efectuar su tarea.
- Un módulo Hx711 al cual se encuentra conectado un Load cell o barra de carga de 1kg que indicará que la cantidad de alimento dispensada sea la correcta según el dato enviado desde la aplicación móvil.

Cabe mencionar que se hace uso de los pines GPIO3 y GPIO1, para llevar a cabo esto y no tener algún inconveniente como se mencionó anteriormente se prosigue a desconectar esos pines al cargar el sketch y conectarlos una vez cargado.

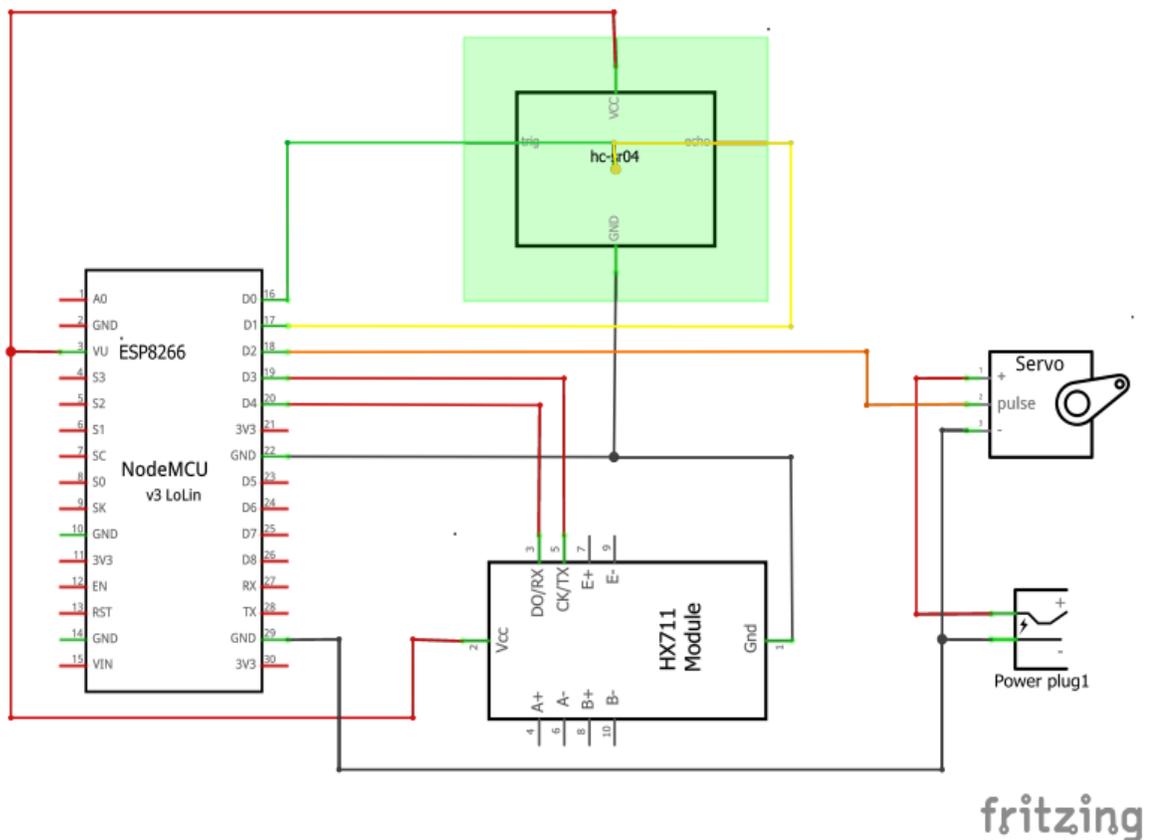


Figura 47. Esquemático del circuito elaborada en Fritzing.

Fuente: Elaboración propia.

Capítulo IV. Análisis de los datos y resultados

4.1. Datos Pet-Home ALBA

A continuación, se analizan los resultados obtenidos de la encuesta pertenecientes al apartado 3.1.4. que se aplicó a las 68 personas resultado del tamaño de la muestra:

1. En la *Figura 48* se determina la cantidad de personas que cuenta con internet en sus hogares y que disponen con al menos un Smartphone, a lo cual el 86.8% de los encuestados cuentan con internet y un 94.1% poseen un Smartphone mientras que el 13.2% no tiene internet y el 5.9% no disponen de un Smartphone.

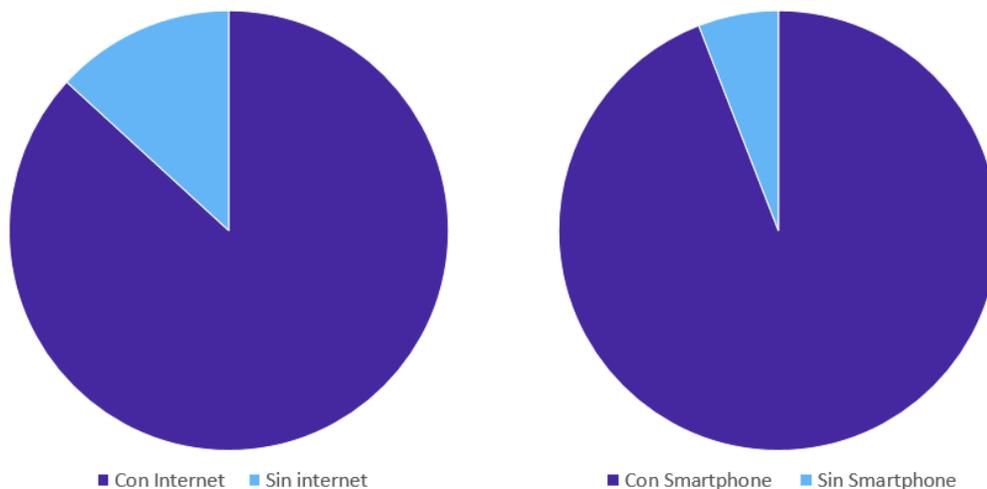


Figura 48. Resultado de la población que cuenta con Internet y un Smartphone.

Fuente: Elaboración propia

2. En la *Figura 49* se determina la cantidad de personas que cuentan con una mascota en sus hogares y de qué tipo son, sin embargo, en caso de que el individuo no tenga alguna mascota se preguntó qué tipo de mascota le gustaría tener, a lo cual, el 88.2% dice sí contar con una mascota mientras que el 11.8% indicó no tener una mascota por otra parte la mascota más popular con un 52.9% son los perros seguidos de un 35.3% que dicen tener un gato y por último un 11.8% indican tener algún otro tipo de mascota.

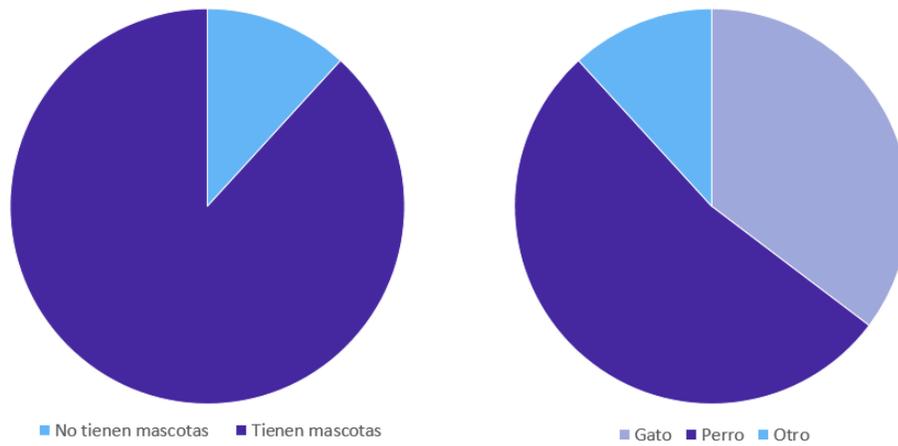


Figura 49. Resultado de la población que cuentan con una mascota y que tipo es.

Fuente: Elaboración propia

- En la *Figura 50* se determina la cantidad de personas que suelen estar fuera de casa y sus motivos siendo que el 85.3% indicó estar más tiempo fuera de su hogar con un 39.7% por trabajo, un 36.8% por la escuela, un 10.3% por otros motivos mientras que el 14.7% y un 13.2% indicó no tener motivos para estar fuera de casa.

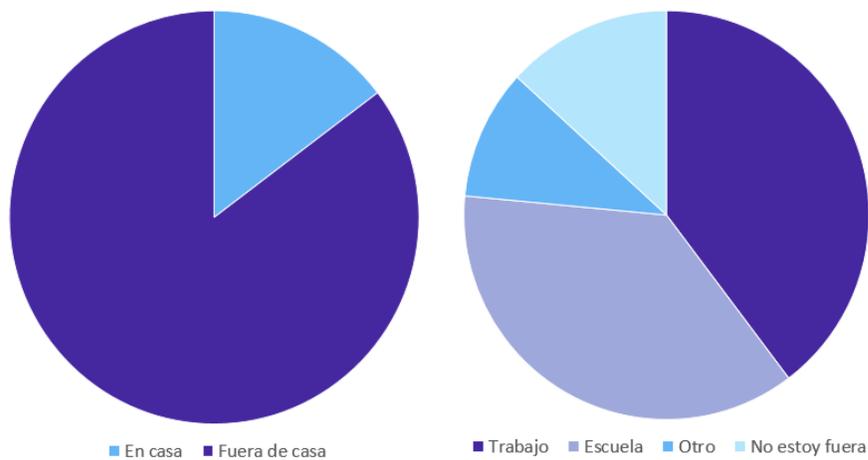


Figura 50. . Resultado de la población que estan fuera de casa y el motivo.

Fuente: Elaboración propia

- En la *Figura 51* se determina la cantidad de personas que en algún momento obtuvo un dispensador de alimento siendo que el 73.5% nunca ha adquirido uno y el 26.5% que si ha tenido un dispensador.

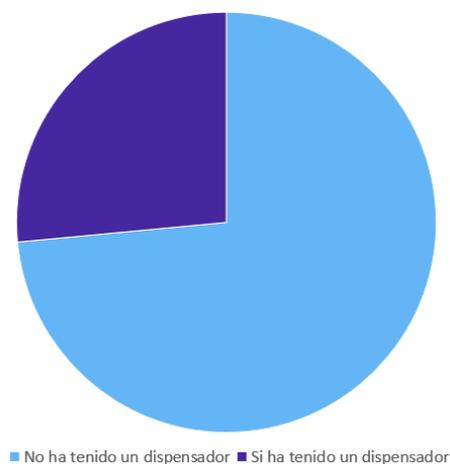


Figura 51. Resultado de la población que ha adquirido un dispensador.

Fuente: Elaboración propia

5. En la *Figura 52* se determina la cantidad de personas que están dispuestas en adquirir un dispensador de alimento para facilitar el suministro alimenticio de su mascota cuando no se encuentren en casa y la cantidad de dinero que están dispuestos a gastar por uno, siendo que un 83.8% afirmó estar interesado en un dispensador y que un 52.9% pagarían entre los \$400.00 a \$700.00 pesos, un 25% pagaría \$800.00 pesos y un 7.4% estarían dispuestos a pagar de \$1000.00 pesos en adelante mientras que un 16.2% y 14.7% no están interesados en adquirir un dispensador.

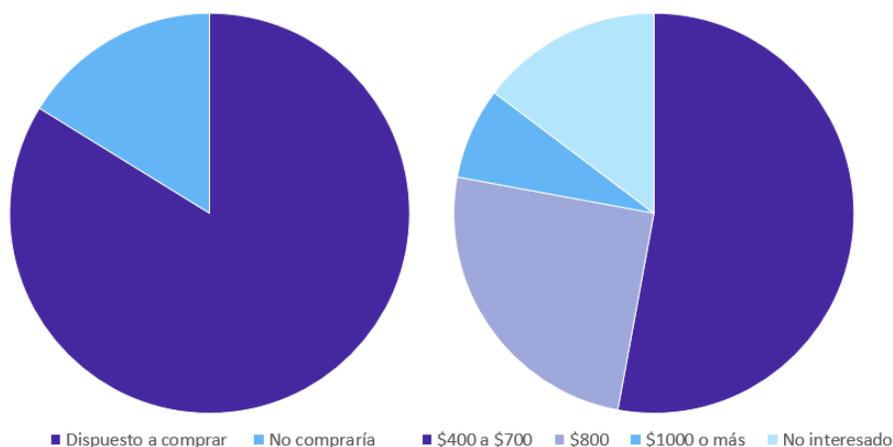


Figura 52. Resultado de la población que está dispuesto a comprar uno y cuanto pagaría.

Fuente: Elaboración propia

Luego de extraer los resultados anteriormente expuestos, se identifica que las respuestas obtenidas por los usuarios se relacionan entre el problema y la propuesta de solución, ya que la mayoría de la población encuestada indicó estar más tiempo fuera de casa y cuentan con una mascota, por lo tanto, se consideraría que presentan problemas al alimentarlas en tiempo, además que un 83.6% de la población dio como respuesta positiva al estar interesado por un dispensador de alimento permitiendo concluir el interés por una solución tecnológica que cumpla con las necesidades de su mascota. Por otra parte, la solución propuesta es factible ya que la mayor parte de la población cuenta con Internet y un Smartphone, haciendo viable la construcción del prototipo Pet-Home ALBA.

4.2. Pruebas de funcionamiento del nodo lector

Se carga el sketch correspondiente el cual incluye las librerías Arduino Client for MQTT para la conexión del módulo con el Broker de Mosquitto alojado en la Raspberry Pi y WiFiManager para la conexión en modo AP del módulo a WiFi. Una vez cargado el sketch como se muestra en la *Figura 53* se prosigue a hacer la conexión de los sensores y actuador al módulo nodo lector ESP8266.

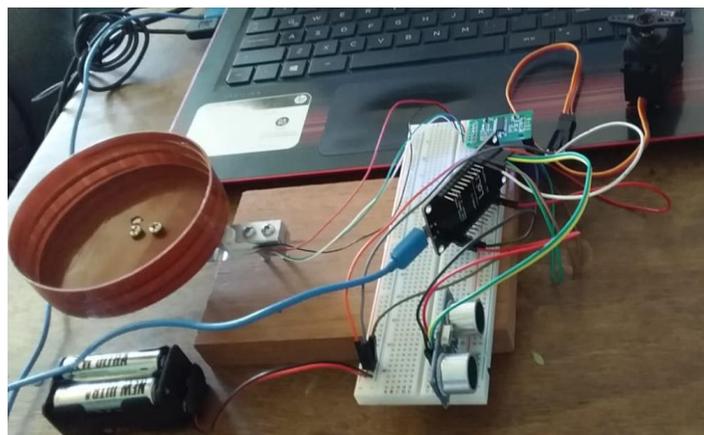


Figura 53. Circuito electrónico para el prototipo.

Fuente: Elaboración propia.

Cuando el sketch se encuentre cargado en el módulo lo primero en ejecutar es la librería WiFiManager y como es la primera vez del módulo al intentar conectarse a la red creará un

AP (Access Point), el cual se visualiza en el celular en el apartado de Wi-Fi o conexiones y veremos una red nombrada *ALBA-Dispensador* se realiza la conexión y se muestra un portal cautivo con la página web mostrando las redes disponibles, seleccionamos la red a conectarse e ingresamos la contraseña y guardamos. Una vez hecho esto, las credenciales se guardan en el módulo para que la próxima vez él mismo se conecte. Si la conexión a la red fue exitosa el led pin D4 se mostrará encendido, como se muestra en la *Figura 54*.

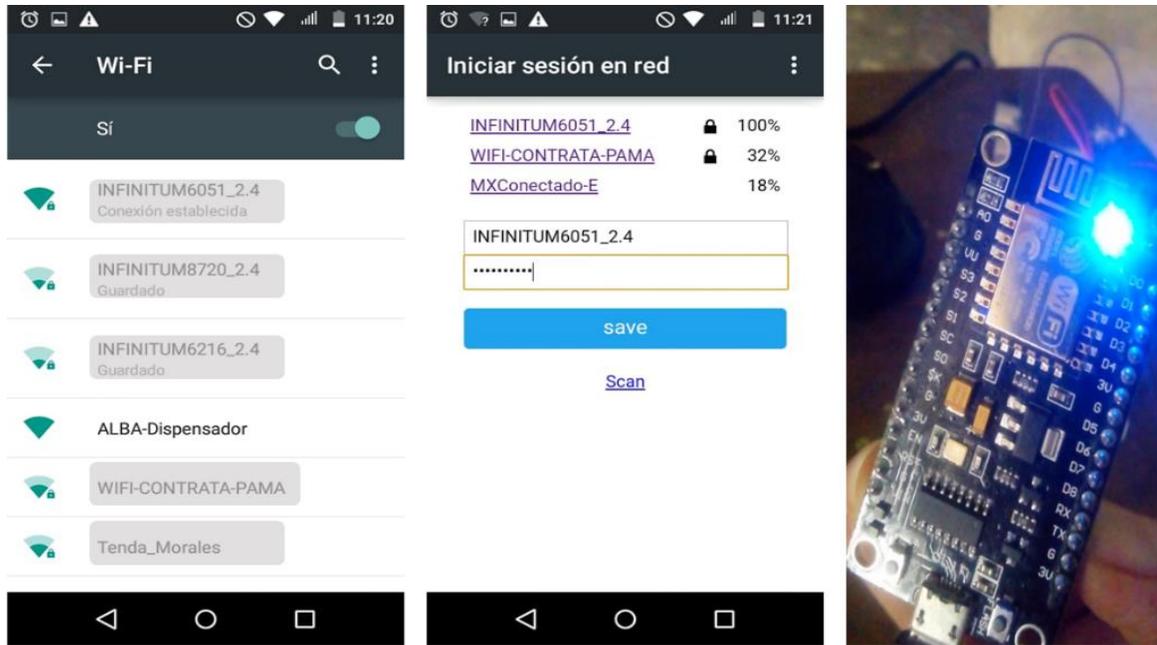


Figura 54. Conexión del nodo lector a wifi por WiFiManager.

Fuente: Elaboración propia.

Se prosigue a realizar las pruebas de funcionamiento del monto del circuito, comprobando la actividad de sensores y actuadores a través de comunicación serial con el monitor de serie del IDE de Arduino entre las pruebas que se realizan están probar la cantidad de alimento dispensado y la cantidad de alimento en el contenedor del dispensador. Una vez que al contenedor se le ha introducido cierta cantidad de comida, se verifica el nivel de alimento. Se procede a accionar el servomotor la cual abre la compuerta y se desliza el alimento para caer sobre el plato de la balanza del prototipo.

En esta prueba se trató de corregir y calibrar la balanza digital concluyendo con resultados satisfactorios primero se espera unos segundos hasta que se quite la tara, siendo que cualquier peso que se perciba no sea tomado en cuenta mientras se encuentre *Destarando* una vez

finalizado, se comienza con el pesaje del alimento en la *Figura 55* se muestra cuando la balanza no tiene alimento indicando: 0grs.



Figura 55. Balanza digital prueba sin alimento

Fuente: Elaboración propia.

4.3. Pruebas de funcionamiento del broker MQTT

A continuación, en el programa de *Advanced Ip Scanner* como se muestra en la *Figura 56* se muestra que la Raspberry se visualiza con el *Hostname* asignado en el proveedor de NO-IP que se configuró en el *apartado 3.4*

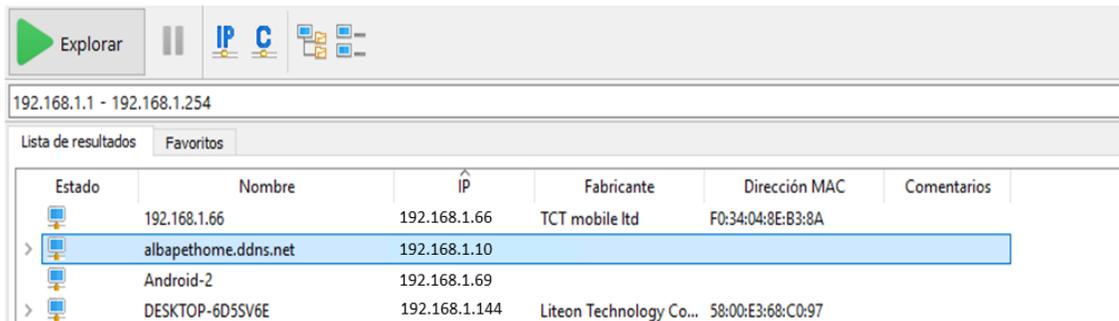


Figura 56. Hostname del cliente noip

Fuente: Elaboración propia.

Cuando el usuario se conecte desde la aplicación móvil al broker MQTT indicando la cantidad de alimento a dispensar a la mascota, para ser posible esto primero se ejecuta un script *controlador.py* que se muestra en la *Figura 57* donde se ejecuta el cliente Python MQTT y que realiza una conexión al subscribirse al tópico indicado desde la aplicación. El script se ejecutó de manera satisfactoria al crear un cliente, conectarse y subscribirse al tópico *casa/sala/dispensador* la primera tarea que se ejecutó es la indicación que el *nodo lector* se encuentra conectado.

```

pi@raspberrypi: /var/www/html/pethome
File Edit Tabs Help
pi@raspberrypi:/var/www/html/pethome $ ls
adddata.php  controlar.py  login.php      updatedata.php
conexion.php deletedata.php update_config.php
pi@raspberrypi:/var/www/html/pethome $ python controlar.py
Creando nuevo cliente
Realizando conexion
Subscribing to topic: sala/dispensador
Led pinD4...

```

Figura 57. Cliente Python MQTT: controlar.py

Fuente: Elaboración propia.

A continuación, se muestra por consola el funcionamiento de la base de datos al realizar la inserción de los datos correspondientes desde la aplicación móvil en este caso en la *Figura 58* se visualizan los datos de la tabla mascotas.

```

MariaDB [dispensador_alba]> select * from mascotas;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mascota | mascota_nombre | id_config | tipo | tamaño | etapa | edad | peso | cantidad |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Mily | 1 | NULL | NULL | NULL | NULL | NULL | NULL |
| 2 | Coco | 2 | Gato | Mediano | Gato Adulto | 6 meses a 1 año | 2 - 3.5 Kg | 350 grs |
| 3 | bolita | 2 | Perro | Mini | Cachorro | 3 - 5 meses | 1 - 5 Kg | 45 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.04 sec)

```

Figura 58. Base de datos Mascotas

Fuente: Elaboración propia

La *Figura 59* corresponde a la configuración del dispensador y, se muestra un campo denominado nivel el cual se almacena según el dato obtenido desde el sensor HC-SR04 por el módulo lector (ESP8266).

```

MariaDB [dispensador_alba]> select * from configdispensadores;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_config | dispensador_nombre | cantidad_dosis | horario_dosis1 | horario_dosis2 | horario_dosis3 | horario_dosis4 | nivel | id_cliente |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | sala | 205 | 14:41:00 | NULL | NULL | NULL | 80 | 2 |
| 2 | cocina | NULL | 14:41:00 | 15:41:00 | 18:41:00 | 19:41:00 | 45.6 | 3 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

Figura 59. Base de datos tabla configuración del dispensador.

Fuente: Elaboración propia

Conclusiones

Una vez finalizado el proyecto podemos concluir que se han alcanzado los principales objetivos previamente establecidos y una descripción personal de la realización de que comprendió el desarrollo de este proyecto.

- Se diseñó y creó una base de datos capaz de almacenar la información de los clientes, los dispensadores y las mascotas además de guardar la información en históricos para que en caso de pérdida de información se facilite la recuperación de dichos datos.
- Se desarrolló un servidor en la plataforma Raspberry Pi completamente funcional, como Broker MQTT se hizo uso de Mosquitto, servicios de LAMP como servidor web dinámico y se creó un host e instaló un cliente NO-IP para acceder de manera remota a los recursos del proyecto al ser servicios Open source el servidor fue de bajo coste. Además, que permite que los proyectos sean escalables al permitir conexión de módulos adicionales ya sea por sus pines o por comunicación Wi-Fi como el módulo NodeMCU, etc.
- Con los servicios que ofrece LAMP como servidor web dinámico se crearon usuarios para acceder de manera segura a la base de datos, así como los permisos para guardar la información proveniente del sensor HC-SR04 en la base de datos para indicar el nivel de alimento.
- La conexión del módulo NodeMcu ESP8266 a Wi-Fi se logró al implementar la librería de WiFiManager obteniendo beneficios como una conexión a red de manera rápida y sin necesidad de tocar una línea de código; permitiendo al usuario conectar el dispensador por el mismo.
- El desarrollo del nodo lector y el circuito electrónico permitió el suministro del alimento y al ser programados en la plataforma de Arduino que se libera bajo la licencia de código abierto no presentó coste alguno, de tal manera que los actuadores ejecutarán la tarea de suministrar y el sensor obtuvo el nivel de alimento del contenedor del dispensador en tiempo real; permitiendo al usuario la facilidad de suministrar el alimento en porciones adecuadas de acuerdo a la información mascota.
- El desarrollo la aplicación móvil logró controlar de forma remota el dispensador de alimento y de manera automática al indicar los horarios a dispensar. Al hacer uso de

Flutter, se logra un diseño atractivo e intuitivo para los usuarios, obteniendo un producto competitivo ante desarrollos híbridos.

Se concluye que la hipótesis no es la esperada ya que uno de los fines era lograr un producto de bajo costo y no se logró, ya que al incluir un servidor en este caso la Raspberry incrementa su valor, una forma para resolver este problema es adquirir los servicios de un alojamiento MQTT en la nube.

La conclusión del trabajo en ámbito personal por una parte presentó un reto académico desde su fase inicial hasta su fase final que fue solventado por medio de la investigación y la contribución de personas con conocimientos sobre las tecnologías utilizadas y, por otra parte, resultó ser un proyecto enriquecedor permitiendo ser una herramienta de aprendizaje e introducción para temas que en la actualidad son tendencias como el Internet de las cosas y el desarrollo de aplicaciones en Flutter comprobando la aplicabilidad del protocolo MQTT en distintos entornos y lenguajes de programación, además de que posibilitó reforzar las competencias adquiridas durante la carrera obtenidas es distintas materias de electrónica, base de datos, ingeniería de software, sistemas operativos y redes haciendo que este proyecto merezca la pena buscar otras alternativas tecnológicas para dar solución a un problema.

Bibliografía

- [1] A. & B. C. Ortega, «Prospectiva de las Tecnologías de Información y Comunicación: La explotación racional de informaciones tecnológicas,» *MULTICIENCIAS*, vol. 8, nº 2, p. 227, 2008.
- [2] C. Á. Arreola, «En México, 57 de cada 100 hogares tienen alguna mascota,» *La Jornada Maya*, 30 Enero 2019 [En línea]. Available: <https://www.lajornadamaya.mx/2019-01-30/En-Mexico--57-de-cada-100-hogares-tienen-alguna-mascota>
- [3] INEGI, «Estadísticas a propósito del día mundial del internet (17 de Mayo) Datos nacionales,» 15 Mayo 2019. [En línea]. Available: https://www.inegi.org.mx/contenidos/saladeprensa/aproposito/2019/internet2019_Nal.pdf. [Último acceso: 28 Septiembre 2019].
- [4] A. e. a. Adriansyah, «Design of Pet Feeder using Web Server as Internet of Things Application,» de *Conference: 2nd International Conference on Electrical Engineering and Informatics (Icon-EEI 2016)*, At: Pekanbaru, Riau, Indonesia, 2016.
- [5] A. Muñoz Chaparro, «Base de datos objeto-relacionales,» de *Oracle 11g PL/SQL*, Madrid, RC Libros, 2012, p. 258.
- [6] K. Ashton, «That Internet of Things Thing,» 2009. [En línea]. Available: <https://www.rfidjournal.com/articles/view?4986>.
- [7] Á. Arias, *Bases de Datos con MySQL*, 2014.
- [8] M. P. é. a. Zea Ordóñez, «PostgreSQL,» de *Administración de bases de datos con PostgreSQL*, Alicante, ÁREA DE INNOVACIÓN Y DESARROLLO, S.L, 2017, p. 12.
- [9] WAMP, «wampserver,» 24 Agosto 2019. [En línea]. Available: <http://www.wampserver.com/en/>. [Último acceso: 10 Septiembre 2019].
- [10] «Laragon,» 24 Agosto 2019. [En línea]. Available: <https://laragon.org/>. [Último acceso: 10 septiembre 2019].
- [11] R. E. a. F. Erick, «Setting up LAMP: Getting Linux, Apache, MySQL and PHP Working Together,» de *Chapter I. Introducing LAMP*, San Francisco, SYBEX, INC, 2004, p. 2.
- [12] F. é. a. Luna, *PROGRAMACION WEB Full Stack*, Buenos Aires, ARG: RedUsers, 2018.

- [13 J. é. a. Novillo-Vicuña, «Protocolos de comunicación,» de *Arduino y el internet de las cosas*, Alcoy, ALC, Área de innovación y desarrollo, S.L, 2018, p. 35.
- [14 «HiveMQ,» © 2020 HiveMQ, 17 Agosto 2019. [En línea]. Available: <https://www.hivemq.com/>. [Último acceso: 12 Agosto 2019].
- [15 Octavo Labs AG, «VernMQ,» 17 Agosto 2019. [En línea]. Available: <https://vernemq.com/>. [Último acceso: 12 Agosto 2019].
- [16 P. Desai, «Mosquitto - an open source MQTT broker,» de *Python Programming for Arduino*, Birmingham - Mumbai, Packt Publishing Ltd, 2015, p. 242.
- [17 S. Chazallet, «Presentación de Python,» de *Python 3: Los fundamentos del lenguaje*, Cornellá de Llobregat, BCN, Editions ENI, 2016, pp. 59-60.
- [18 «PlatformIO,» 2014-2019 PlatformIO, 17 Agosto 2019. [En línea]. Available: <https://platformio.org/platformio-ide>. [Último acceso: 20 Noviembre 2019].
- [19 O. Torrente Artero, *ARDUINO curso práctico de formación*, México: Alfaomega, 2013.
- [20 C. Peña Millahual, «Shields disponibles,» de *ARDUINO: De cero a experto*, Buenos Aires, Six Ediciones, 2017, p. 299.
- [21 M. McGrath, «Raspberry Pi 3,» de *Introducing Raspberry Pi*, Leamington Spa, Inglaterra, In Easy Steps Limited, 2016.
- [22 P. Bertoleti, «Capítulo 2: Introducción a ESP32,» de *Proyectos com ESP32 y LoRA.*, Brasil, Instituto Newton C Braga, 2019.
- [23 M. Schwartz, «NodeMCU ESP8266,» de *ESP8266 Internet of Things Cookbook*, Birmingham, Packt, 2017, pp. 105-106.
- [24 Carnegie Mellon Robotics Academy, «Robotics Academy,» 27 Agosto 2019. [En línea]. Available: http://cmra.rec.ri.cmu.edu/content/electronics/boe/ultrasonic_sensor/1.html. [Último acceso: 5 Abril 2019].
- [25 A. Semiconductor, «Sparkfun,» 27 Agosto 2019. [En línea]. Available: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf. [Último acceso: 2019 Diciembre 07].
- [26 Icaza Álvarez, D. O. ét al, «Dispensador automático de alimento para mascotas,» *FIGEMPA: Investigación y Desarrollo*, vol. 2, nº 2, pp. 99-106, 2017.

- [27 TG Berhan ét al, «Programmable Pet Feeder,» *International Journal of Scientific Engineering and Research (IJSER)*, vol. 3, n° 11, pp. 99-104, 2015.
- [28 A. G, Giggiola del Pilar ét al, «Happy pet Dispensador electrónico para mascotas de agua y comida,» Universidad Peruana de Ciencias Aplicadas UPC, Lima, 2018.
- [29 K. Aasavari y G. (Mohite), Vaishali, «Automatic Pet Feeder,» SSRN, Mubai, IN, 2018.
- [30 L. –A, Nelson Andrés ét al, «Diseño de dispensador automático de alimento para mascotas, usando una celda de carga,» *Revista Científica UNINCCA*, vol. VIII, n° 1, pp. 29-41, 2017.
- [31 M. R, Alberto ét al, «Dispensador automático de alimento seco para mascotas caninas,» *Proyectos de Ingeniería Multidisciplinaria Una muestra del talento mexicano*, vol. I, pp. 69-74, 2017.
- [32 Vania, & Karyono, Kanisius & Nugroho, Hargyo, «Smart dog Feeder design using Wireless communication, MQTT and Android Client,» de *de 2016 International Conference on Computer, Control, Informatics and its Applications*, Tangerang, IDN, 2016.
- [33 A. A., Sandra Nathaly , «Desarrollo de un prototipo para dispensar alimento y agua para mascotas con tecnología IOT monitoreado y controlado a través de un dispositivo móvil,» Escuela Politécnica Nacional (EPN), Quito, EC, 2019.
- [34 C. Own, H. Shin and C. Teng., «The study and Application of the IoT in Pet System,» *Advances in Internet of Things*, vol. III, n° 1, pp. 1-8, 2013.
- [35 de la Cal. C., Javier, «Control domótico basado en el protocolo MQTT,» Universidad de Valladolid, Valladolid, 2019.
- [36 C. C. Mario y G. L. Joan , «Monitorización de sensores con Arduino utilizando el protocolo MQTT,» Universitat Politècnica de Catalunya, Catalunya, 2019.
- [37 P. T, C. Laura , «Diseño e implementación de un sistema dispensador de alimento para mascota, controlado vía ethernet,» Escuela Superior de Ingeniería mecánica y eléctrica unidad Culhuacan, D.F, MEX, 2013.
- [38 B. –Jr, Angelo A. ét al, «Arduino-based Food and Water Dispenser for Pets with GSM Technology Control,» *IJSET*, vol. 4, n° 231-234, p. IV, 2015.
- [39 L. Q, John David y R. A, Daniel , «Dispensador automático de comida para mascotas, programable y controlado remotamente,» Facultad de Ingeniería UNIVALLE, Santiago de Cali, COL, 2013.

- [40] Asaner, Ugur & Elibol, Armagan, «Low-Cost IoT Design and Implementation of a Remote Food and Water Control System for Pet Owners,» *HJSE*, vol. V, n° 4, pp. 317-320, 2018.
- [41] Cañamas, Jeremie, «Automatización y asistencia remota para cuidado de mascotas,» Escola Tècnica Superior d'Enginyeria Informàtica (UPV), Valencia, ESP, 2017/2018.
- [42] Unidad de microrregiones, «INEGI. Censo de Población y Vivienda 2010.,» 2013. [En línea]. Available: <http://www.microrregiones.gob.mx/zap/poblacion.aspx?entra=nacion&ent=30&mun=109>. [Último acceso: 03 Junio 2019].
- [43] «Purina Cat Chow,» Purina Latam, 24 Julio 2019. [En línea]. Available: <https://www.purina-latam.com/mx/catchow>. [Último acceso: 29 Julio 2019].
- [44] «Purina Latam,» 24 Julio 2019. [En línea]. Available: <https://www.purina-latam.com/mx/dogchow>. [Último acceso: 29 Julio 2019].
- [45] «Raspberry,» Raspberry Pi Foundation, 12 Octubre 2019. [En línea]. Available: <https://www.raspberrypi.org/downloads/raspbian/>. [Último acceso: 30 Noviembre 2019].
- [46] D.-A. a. B. R. LeBlanc, «Part IV: Junior administrator boot camp,» de *Linux for dummies 8th edition*, Indianapolis, IN, Wiley Publishing, Inc, 2007, p. 308.
- [47] GlavSoft LLC, «TightVNC,» 2006-2019. [En línea]. Available: <https://www.tightvnc.com/>. [Último acceso: 1 Noviembre 2019].

Anexos

Cuestionario Pet-Home ALBA

El presente instrumento forma parte del trabajo de investigación titulada: "Prototipo dispensador de alimento para mascotas Pet-Home ALBA".
La información que nos proporcione será utilizada para analizar y determinar la factibilidad del prototipo. Solo le tomará unos minutos. Agradezco anticipadamente su valiosa colaboración.

1. ¿Cuenta con internet en casa?

Si No

2. ¿Cuenta con un Smartphone?

Si No

3. ¿Tiene mascotas en su hogar?

Si No

4. ¿Qué tipo de mascota es?

Si no tiene una mascota, ¿Qué tipo de mascota le gustaría tener?

Gato Perro Otro

5. ¿Suele estar mucho tiempo fuera de casa?

Si No

6. ¿Motivo por el cual está fuera de casa?

Escuela Trabajo Otros motivos Casi no salgo

7. ¿Ha obtenido un dispensador de alimento?

Si No

8. ¿Estaría dispuesto a comprar un dispensador inteligente que alimente a su mascota cuando no se encuentre en casa?

Si No

9. ¿Cuánto dinero está dispuesto a pagar por un dispensador inteligente?

\$400 a \$700 \$800 \$1000 o más No interesado