

**Diseño e implementación de prototipo de dispensador remoto de comida para perros y
gatos a través de IOT**

John Jaiver Ardila Vargas

Asesor

Ing. Jairo Luis Gutiérrez

Universidad Nacional Abierta y a Distancia UNAD
Escuela de Ciencias Básicas, Tecnología e Ingeniería - ECBTI

Ingeniería Electrónica

2025

Resumen

En los hogares colombianos actuales es muy frecuente que se adopten mascotas en especial gatos y perros. Algunas veces las personas presentan dificultades para alimentar sus mascotas durante el día, por sus extensos horarios laborales o en días de ausencia por vacaciones, por ejemplo. Por tanto, este proyecto busca desarrollar un prototipo de un dispensador de comida para perros gatos que funciona remotamente a través de internet.

El dispensador cuenta con un sistema mecánico diseñado para la dispensación controlada del alimento. El alimento se almacena en un recipiente y, mediante un conducto llega a un mecanismo de dosificación que regula la cantidad liberada. Este mecanismo se activa a través de un servomotor, el cual es controlado por la tarjeta ESP32. La tarjeta cuenta con un módulo Wi-Fi conectado a internet y a un bróker MQTT, desde donde se envía la instrucción de activación. Una aplicación creada permite al usuario enviar esta instrucción a través del bróker MQTT en su versión gratuita, llegando así a la tarjeta ESP32 para activar la dispensación.

El proyecto demuestra una de las tantas aplicaciones del internet de las cosas en la época actual, el cual permite integrar muchos procesos domésticos e industriales a través de la red facilitando la vida diaria, así como mejorando la productividad en la industria.

Palabras Clave: *Internet de las Cosas, Dispensador automático, , ESP32, Alimentación remota, MQTT*

Abstract

In today's Colombian homes, adopting pets, especially cats and dogs, is very common. Sometimes people struggle to feed their pets during the day, due to long work schedules or vacation days, for example. Therefore, this project seeks to develop a prototype of a food dispenser for dogs and cats that operates remotely via the internet.

The dispenser features a mechanical system designed for controlled food dispensing. The food is stored in a container and, through a conduit, reaches a dosing mechanism that regulates the amount released. This mechanism is activated by a servomotor, which is controlled by the ESP32 card. The card has a Wi-Fi module connected to the internet and an MQTT broker, from which the activation instruction is sent. A created application allows the user to send this instruction through the free MQTT broker, thus reaching the ESP32 card to activate the dispensing.

The project demonstrates one of the many applications of the Internet of Things today, which allows many domestic and industrial processes to be integrated through the network, making daily life easier and improving productivity in industry.

Keywords: *Internet of Things, Automatic dispenser, , ESP32, Remote feeding, MQTT*

Contenido

| | |
|--|----|
| Introducción | 10 |
| Planteamiento del problema | 12 |
| Justificación..... | 13 |
| Objetivos | 15 |
| Objetivo general | 15 |
| Objetivos específicos..... | 15 |
| Marco conceptual y teórico..... | 16 |
| Internet de las cosas (IoT) | 16 |
| Protocolo MQTT..... | 17 |
| Placa de desarrollo | 18 |
| Definición de Diseño CAD | 19 |
| Arduino IDE..... | 20 |
| ¿Qué es MIT App inventor? | 21 |
| Metodología | 22 |
| Fase 1 – Selección de Componentes | 22 |
| Fase 2 – Diseño del Sistema..... | 23 |
| Fase 3 – Implementación del Prototipo..... | 23 |
| Fase 4 – Evaluación y análisis..... | 23 |
| Desarrollo Ingenieril | 24 |

| | |
|---|----|
| Selección de Componentes..... | 24 |
| Elementos Estructurales, Mecánicos y Físicos..... | 24 |
| Elementos Electrónicos y de Control | 27 |
| Diseño del Sistema | 35 |
| Circuito de control..... | 35 |
| Plano electrónico del circuito | 36 |
| Cálculo de resistencia base..... | 38 |
| Diseño de la PCB | 40 |
| Código microcontrolador..... | 42 |
| Diseño de aplicación móvil en MITAPP inventor..... | 49 |
| Diseño de la tolva..... | 53 |
| Diseño de la Tapa Superior | 54 |
| Diseño de las Tapas Laterales Izquierda y Derecha | 55 |
| Diseño de la Tapa Frontal..... | 57 |
| Diseño de la Tapa Posterior..... | 58 |
| Diseño Conceptual del prototipo..... | 60 |
| Implementación del Prototipo | 61 |
| Ensamblaje Mecánico | 61 |
| Ensamble electrónico y programación | 74 |
| Evaluación y análisis..... | 80 |

| | |
|--|----|
| Pruebas 1: funcionamiento local | 81 |
| Prueba 1.1: Funcionamiento Local mediante Botón de Inicio..... | 81 |
| Prueba 1.2: Control mediante Aplicación Local | 83 |
| Demostración de funcionamiento del prototipo en video | 85 |
| Prueba 2: Funcionamiento Remoto..... | 85 |
| Prueba 2.1: Control Remoto desde Bogotá (Ubicación Remota)..... | 86 |
| Prueba 2.2: Control Remoto desde Medellín (Ubicación Remota Nacional) | 88 |
| Prueba 2.3: Control Remoto desde Andalucía, España (Ubicación Remota Global) | 90 |
| Resultados | 94 |
| Conclusiones | 95 |
| Referencias Bibliográficas..... | 96 |

Lista de figuras

| | |
|--|----|
| Figura 1 <i>Arquitectura MQTT</i> | 18 |
| Figura 2 <i>Placas de Desarrollo</i> | 19 |
| Figura 3 <i>Entorno de Programación Arduino</i> | 20 |
| Figura 4 <i>Pote de Plástico Redondo</i> | 25 |
| Figura 5 <i>Semicodo con Tapa Reguladora</i> | 26 |
| Figura 6 <i>Paleta de Flujo</i> | 27 |
| Figura 7 <i>Funcionamiento del Pulsador</i> | 28 |
| Figura 8 <i>Pulsador Normalmente Abierto (NO)</i> | 29 |
| Figura 9 <i>Pines Tarjeta ESP32</i> | 30 |
| Figura 10 <i>Servo MG90S</i> | 31 |
| Figura 11 <i>Diagrama del Sistema</i> | 36 |
| Figura 12 <i>Plano Electrónico del Sistema</i> | 38 |
| Figura 13 <i>Diseño Final PCB</i> | 41 |
| Figura 14 <i>Visualización 3D de la Placa</i> | 41 |
| Figura 15 <i>Sección Diseñador Mit App Inventor</i> | 51 |
| Figura 16 <i>Diseño de Bloques Mit App Inventor</i> | 53 |
| Figura 17 <i>Plano de la Tolva</i> | 54 |
| Figura 18 <i>Diseño Tapa Superior</i> | 55 |
| Figura 19 <i>Diseño Lateral Izquierdo</i> | 56 |
| Figura 20 <i>Diseño Tapa Lateral Derecho</i> | 57 |
| Figura 21 <i>Diseño Tapa Frontal</i> | 58 |
| Figura 22 <i>Diseño Tapa Posterior</i> | 59 |

| | |
|--|----|
| Figura 23 <i>Modelo 3D del Prototipo Ensamblado</i> | 61 |
| Figura 24 <i>Ensamble Recipiente de Almacenamiento y Tolva</i> | 62 |
| Figura 25 <i>Fijación Lamina y Servomotor</i> | 63 |
| Figura 26 <i>Soporte Motor Vibratorio</i> | 64 |
| Figura 27 <i>Brazo Plástico Servomotor</i> | 65 |
| Figura 28 <i>Acople de la Paleta de Flujo</i> | 66 |
| Figura 29 <i>Fijación del Dispositivo de Dosificación a la Base</i> | 67 |
| Figura 30 <i>Tolva Impresa en 3D</i> | 69 |
| Figura 31 <i>Acople de Tolva y Semicodo</i> | 70 |
| Figura 32 <i>Tapa Superior</i> | 71 |
| Figura 33 <i>Panel Frontal</i> | 72 |
| Figura 34 <i>Tapa Posterior</i> | 73 |
| Figura 35 <i>Soporte Horizontal</i> | 74 |
| Figura 36 <i>Generación de Enlace d e Descarga</i> | 76 |
| Figura 37 <i>Instalacion de Fuentes Desconocidas</i> | 77 |
| Figura 38 <i>Análisis de Seguridad de la Aplicación</i> | 78 |
| Figura 39 <i>PetFoodApp</i> | 79 |

Lista de tablas

| | |
|--|----|
| Tabla 1 <i>Características principales del Servomotor</i> | 31 |
| Tabla 2 <i>Características Principales del Adaptador de Corriente</i> | 33 |
| Tabla 3 <i>Especificaciones Motor Vibratorio</i> | 34 |
| Tabla 4 <i>Resultados Prueba 1.1</i> | 82 |
| Tabla 5 <i>Resultados Prueba 1.2</i> | 84 |
| Tabla 6 <i>Resultados prueba 2.1</i> | 87 |
| Tabla 7 <i>Resultados Prueba 2.2</i> | 89 |
| Tabla 8 <i>Resultados Prueba 2.3</i> | 92 |

Introducción

En la actualidad, la presencia de mascotas, especialmente perros y gatos ha adquirido un nuevo significado en los hogares. Ya no se les considera simples animales de compañía, sino miembros plenos de la familia. Esta transformación del vínculo humano-animal ha generado una demanda creciente por hallar soluciones que aseguren su bienestar, sobre todo cuando el estilo de vida moderno dificulta el cumplimiento de rutinas básicas, como la alimentación.

Muchos propietarios de mascotas enfrentan problemáticas debido a extensas jornadas laborales, viajes imprevistos o ausencias prolongadas. Estas situaciones pueden interferir con una alimentación regular y adecuada de las mascotas, lo que no solo genera preocupación en los cuidadores, sino que también puede tener consecuencias negativas para la salud del animal, como el sobrepeso, la ansiedad alimentaria o la desnutrición. Por lo tanto, garantizar una nutrición oportuna, en porciones controladas y con la debida frecuencia, se ha convertido en una necesidad fundamental en el cuidado de mascotas.

Para abordar estas problemáticas, la integración de tecnologías modernas como el Internet de las Cosas (IoT) se presenta como una gran oportunidad. Gracias a ellas, podemos desarrollar soluciones orientadas al cuidado animal. La posibilidad de interconectar dispositivos físicos a través de internet permite que los usuarios puedan supervisar y controlar procesos en tiempo real desde cualquier ubicación, siempre que cuenten con acceso a la red. Esta capacidad ha dado lugar a avances significativos en la automatización del hogar, lo que se conoce como domótica, y su aplicación en el ámbito del cuidado de mascotas representa un gran avance para el bienestar animal.

En este contexto se enmarca el presente proyecto, cuyo objetivo central es el diseño y la implementación de un prototipo funcional de dispensador automático y remoto de alimento para

perros y gatos, controlado mediante tecnología IoT. La propuesta busca responder a la necesidad concreta de asegurar la alimentación de las mascotas de forma eficiente, programada y controlada, incluso en ausencia del cuidador, promoviendo al mismo tiempo una experiencia de uso sencilla e intuitiva.

El prototipo desarrollado consta de tres componentes principales: un sistema mecánico de dosificación precisa, un circuito electrónico basado en la tarjeta de desarrollo ESP32, y una aplicación móvil diseñada para facilitar la interacción del usuario con el dispositivo. Esta interfaz permite activar el dispensador previamente calibrado según la necesidad, mediante el protocolo de mensajería MQTT como también activar el dispensador manualmente en caso necesario.

Este proyecto es importante no solo por la comodidad que ofrece al garantizar la alimentación de las mascotas en ausencia de sus dueños, sino también por mostrar cómo el IoT puede integrarse en la vida diaria del hogar. Además, sienta las bases para futuras soluciones inteligentes en el cuidado animal, con posibles mejoras como sensores, cámaras o algoritmos de aprendizaje automático. El documento detalla el desarrollo del prototipo, su integración con una app móvil, los resultados de las pruebas y recomendaciones para futuras mejoras orientadas a distintas necesidades.

Planteamiento del Problema

En la actualidad es muy común que en los hogares colombianos se adopten mascotas domésticas, inclusive muchas personas y parejas jóvenes optan por cuidar mascotas en lugar de hijos. Según una encuesta multipropósito del DANE (2022) para el 2021, el 40% de los hogares en Bogotá dijeron tener al menos una mascota, de este dato el 65% corresponde a perros y cerca del 43.7% pertenece a gatos. Estos datos nos indican que existe una población considerable de perros y gatos en la ciudad de Bogotá. Ahora bien, por cuestiones de trabajo o estudio muchas de estas personas tienen que dejar solas a sus mascotas varias horas, o en ocasiones, varios días si se presentan viajes o vacaciones, lo que genera un gran reto para los cuidados que tiene una mascota y sobre todo la alimentación correcta.

Algunas personas dejan los alimentos de sus mascotas servidos durante horas lo cual trae más problemas que beneficios al estar expuestos al aire libre. Según Noguez (2020) esta práctica inadecuada trae consecuencias como la pérdida de nutrientes, por lo que ya no será apetitosa para los animales y ya no querrán comerla. También podría causar problemas de salud en los animales al ingerir comida contaminada. Teniendo en cuenta estas consecuencias se generarían pérdidas económicas y una generación de desechos extra. Por lo que surge el interrogante si existe la manera de alimentar los perros y gatos hogar desde el lugar que se encuentre en caso de no estar en casa. La implementación de esta solución tecnológica busca responder a la pregunta de cómo el desarrollo de un prototipo de comedero activado remotamente a través de internet puede solucionar el problema de la alimentación fresca de perros y gatos en hogares bogotanos con dueños ausentes, mejorando su calidad de vida y la de sus dueños.

Justificación

El presente proyecto se enfocará en la investigación, desarrollo e implementación de un prototipo que exponga una posible solución a la problemática que se presenta en la vida diaria de muchas personas en Bogotá al ausentarse varias horas o, algunas veces, días enteros, y no poder alimentar a sus mascotas o no tener alguien de confianza para encargarse de esta tarea. En respuesta a las consecuencias negativas de la alimentación inadecuada, como la pérdida de nutrientes y el riesgo de contaminación del alimento expuesto al aire libre (Noguez, 2020), este proyecto busca proponer una solución que permita a los dueños alimentar a sus mascotas efectivamente sin retrasos, en horarios establecidos con porciones adecuadas y sin una gran inversión que afecte sus finanzas. Esta necesidad se ve respaldada por datos como la encuesta multipropósito del DANE (2022), que indica que un significativo 40% de los hogares en Bogotá tienen al menos una mascota, lo que subraya la amplia población que podría beneficiarse de una solución como la propuesta.

Con el auge de las nuevas tecnologías y el internet de las cosas se encuentra una ventaja que se puede aplicar en este proyecto, ya que se pueden ejecutar acciones en tiempo real desde cualquier lugar, siempre y cuando exista una conexión a internet. La creciente conciencia sobre la salud y el bienestar animal impulsa la búsqueda de métodos de alimentación más controlados y seguros, evitando los riesgos asociados con la sobrealimentación o la ingesta de alimentos en mal estado. Se busca proponer un prototipo de dispensador de comida activado remotamente a través de internet, usando una placa de desarrollo programada a través de software especializado para el manejo de unos actuadores, y aplicando técnicas de diseño para la parte física estructural del proyecto. Este enfoque tecnológico del prototipo también abre la posibilidad de integrar funcionalidades adicionales a futuro, como el monitoreo de la cantidad de alimento restante, la

programación de planes de alimentación personalizados según las necesidades específicas de cada mascota, la monitorización visual de la alimentación a través de una cámara y la posibilidad de adicionar la comunicación de voz reforzando su vínculo.

La preocupación de los dueños por el bienestar de sus mascotas durante sus ausencias se aborda directamente con esta solución, brindando la tranquilidad de poder alimentarlas de manera oportuna desde cualquier ubicación. Este enfoque tecnológico del prototipo también abre la posibilidad de integrar funcionalidades adicionales a futuro, como el monitoreo de la cantidad de alimento restante, la programación de planes de alimentación personalizados según las necesidades específicas de cada mascota, la monitorización visual de la alimentación a través de una cámara y la posibilidad de adicionar la comunicación de voz, permitiendo a los dueños hablar a sus mascotas a distancia, brindándoles consuelo y reforzando su vínculo.

Las condiciones de alimentación mejorarán porque el proyecto permitirá que se brinden porciones adecuadas y en horas justas, se evita que la comida quede expuesta al aire libre por mucho tiempo conservando sus nutrientes y reduce el riesgo de contaminación. Adicionalmente, esta solución tiene el potencial de ofrecer una alternativa innovadora para el cuidado de mascotas en ausencia de sus dueños, brindando mayor control y tranquilidad a los dueños, contribuyendo significativamente al bienestar de sus compañeros animales al asegurar una alimentación fresca, en porciones adecuadas y a horarios establecidos.

Objetivos

Objetivo General

Desarrollar un prototipo de dispensador de comida para perros y gatos que sea activado remotamente a través de internet, en los hogares colombianos.

Objetivos Específicos

Determinar los componentes electrónicos y físicos adecuados para el dispensador.

Diseñar el sistema electrónico, la estructura física y el control del dispositivo.

Implementar el prototipo ensamblando los componentes y sistemas diseñados.

Evaluar el prototipo mediante pruebas funcionales para asegurar su correcto desempeño.

Marco Conceptual y Teórico

En el diseño del prototipo se integran diferentes conceptos de ingeniería tales como desarrollo electrónico, programación de microcontroladores a través de lenguajes de alto nivel y diseño asistido por computadora para algunas partes necesarias en la estructura del prototipo. El dispensador automático está fundamentado en principios teóricos y conceptos que están asociados a cada componente que lo integra los cuales se relacionan a continuación.

Internet de las Cosas (IoT)

El internet de las cosas se puede definir como la interconexión de distintas cosas u o elementos que intercambian datos a través de internet. Según Fernández (2021) el termino lo acuño Kevin Aston en 2009 para nombrar su red de sensores e identificadores conectados entre sí. La aparición de esta red de información surge de la evolución de conceptos como SCADA y la comunicación entre maquinas M2M que usan distintas tecnologías y protocolos de comunicación. El IOT se basa en tres elementos básicos:

La Tecnología Embebida. Se refiere a la amplia gama de dispositivos que integran hardware y software dedicados para interactuar con su entorno. Estos dispositivos están equipados con sensores, transductores que convierten magnitudes físicas (como temperatura, luz o movimiento) en señales eléctricas. También incorporan actuadores, mecanismos que permiten al dispositivo realizar acciones físicas en respuesta a los datos procesados. La capacidad de procesamiento reside en microprocesadores o microcontroladores, unidades de cómputo de bajo consumo diseñadas para ejecutar tareas específicas y gestionar las operaciones del dispositivo.

Conectividad. Este elemento abarca las diversas redes y protocolos que facilitan la comunicación bidireccional entre los dispositivos IoT y otros sistemas. La

interconexión se logra mediante tecnologías como Wi-Fi Bluetooth. redes celulares (LTE, 5G), entre otras. También se refiere a los protocolos de comunicación (por ejemplo, MQTT, HTTP) que se eligen dependiendo factores como el volumen de datos, la latencia, el consumo energético y el alcance de la red.

Datos. Datos: Constituyen la información fundamental que circula en el ecosistema IoT. Se trata de toda la información generada por los sensores de los dispositivos y los datos que se envían, ya sea como órdenes directas a otros dispositivos o como información base para la ejecución de toma de decisiones automáticas por sistemas inteligentes. También incluye los datos intercambiados entre los propios dispositivos para coordinar acciones o compartir información localmente.

Protocolo MQTT

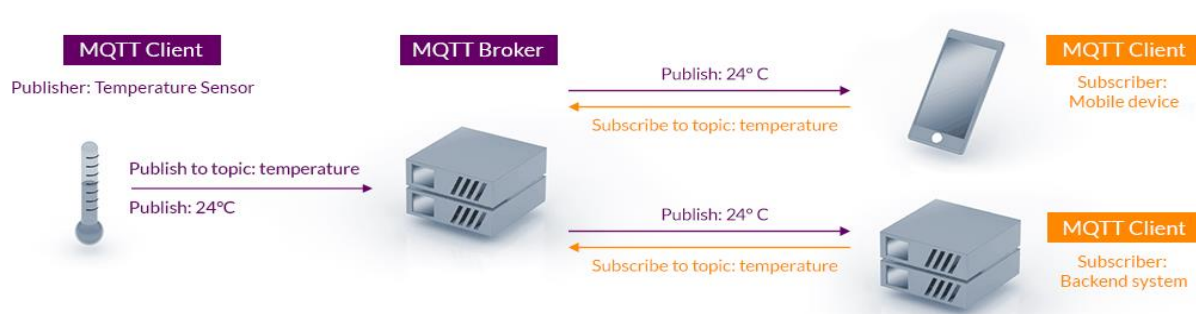
Según Llamas (2019) el protocolo MQTT es uno de los principales protocolos del IOT por su sencillez y su poco consumo de recursos. Sus siglas vienen del inglés Message Queing Telemetry Transport, inicialmente se usaba para conectar dispositivos en la industria petrolera. El protocolo MQTT se usa sobre TCP/IP y utiliza la topología de Publish /suscribe.

Este protocolo ha llamado la atención por su utilidad en el Internet de las Cosas (IoT), gracias a características clave como su ligereza, simplicidad, bajo consumo de recursos y el reducido ancho de banda que requiere. Según Cerón (2024), el protocolo está compuesto por varios elementos fundamentales: el bróker, que actúa como servidor encargado de distribuir la información entre los clientes conectados que estén interesados en ella; el cliente, que es el dispositivo que se conecta al bróker con el objetivo de publicar o recibir información; el topic, que puede entenderse como el nombre del mensaje que se puede publicar o al que uno puede suscribirse; publish, que se refiere al momento en que los clientes envían un topic al bróker para

que otros clientes interesados puedan acceder a esa información; y subscribe, que es cuando los clientes se suscriben a un topic para recibir los datos relacionados.

Figura 1

Arquitectura MQTT



Nota. Arquitectura protocolo Mqtt. [Imagen] Tomado de mqtt.org

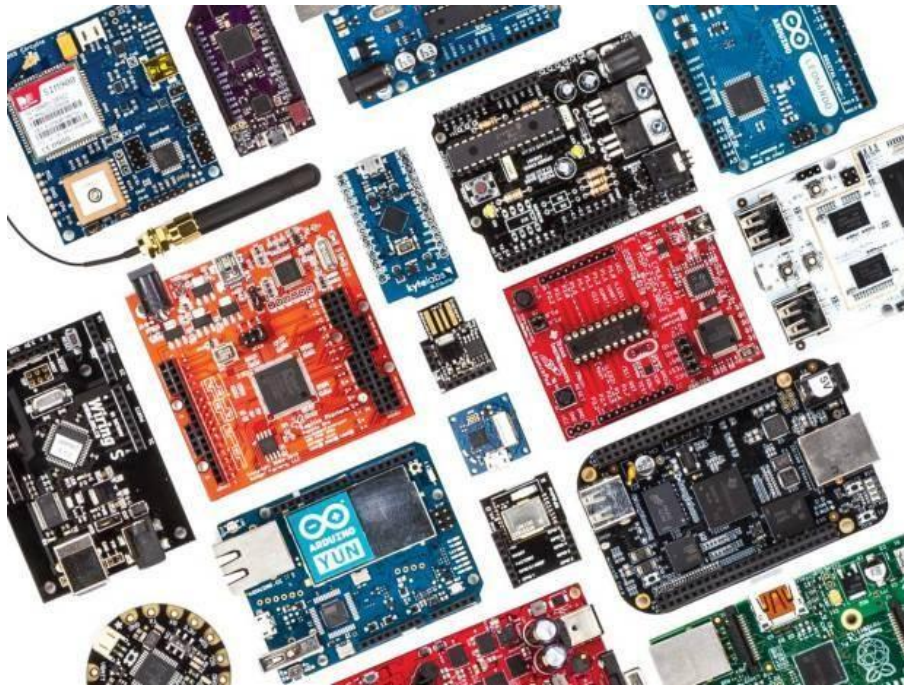
Placa de Desarrollo

Una placa de desarrollo es una pequeña placa de circuito que contiene un microcontrolador, un microprocesador que suministra los componentes y el software necesarios para el diseño y la programación. Actualmente en el mercado existen una gran variedad de tarjetas, placas, sistemas embebidos, que cuentan con distintas características en cuanto a memoria, capacidad, rendimiento, compatibilidad con periféricos sistemas de comunicación. (María, 2024).

Una de las principales ventajas del uso de estas placas es que muchos de sus recursos son de código abierto, lo que permite el uso libre del hardware y software. Esto proporciona que miles de colaboradores alrededor del mundo creen, modifiquen y difundan diferentes versiones de software y adapten los programas desarrollados a cada necesidad.

Figura 2

Placas de Desarrollo



Nota. Adaptado de ¿qué tarjetas de desarrollo elegir parte 1? [Imagen].

<https://hacedores.com/que-tarjeta-de-desarrollo-elegir-parte-1/>

Definición de Diseño CAD

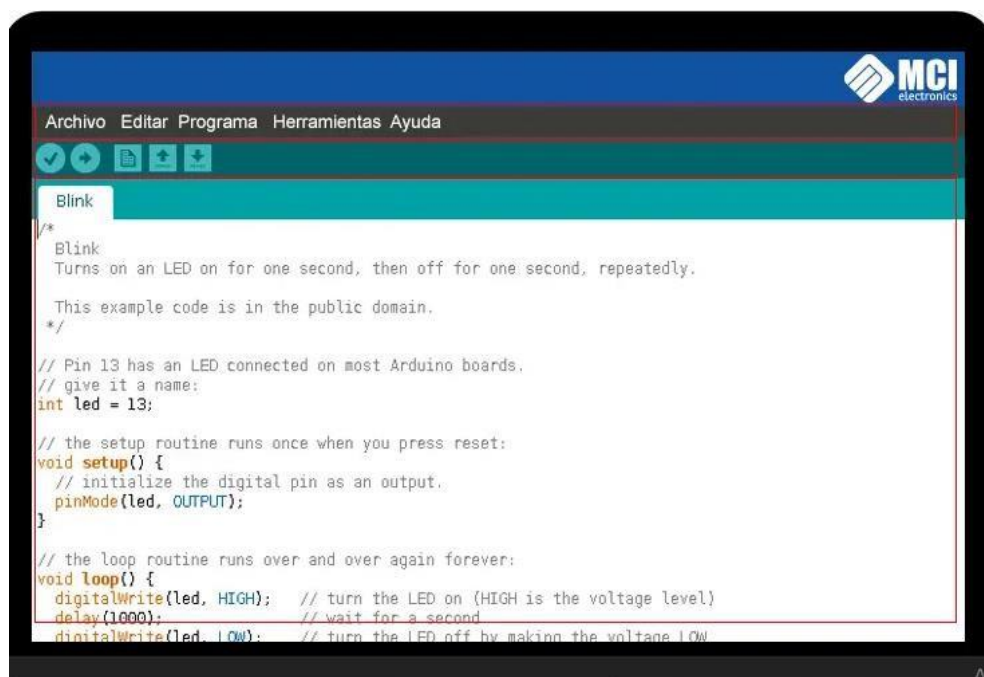
Moraguez (2023) asegura que el software de diseño asistido por ordenador conocido por sus siglas en inglés *Computer- Aided Design* (CAD), permite la creación de modelos de objetos físicos en 2 o 3 dimensiones. Este tipo de software permite aumentar la calidad y precisión, reduciendo los errores en los diseños de productos, ya que facilita la interacción con los modelos con la comodidad de hacer mejoras o cambios muy precisos. Funciona mediante una interfaz gráfica de usuario (GUI) (*Grphical User Interface en inglés*) en donde *el* usuario o diseñador interactúa con el programa manipulando distintos elementos como líneas, formas y puntos para crear el diseño que se desea. En el proyecto se usa con el fin de diseñar distintas piezas del mecanismo de dispensado y acople de distintos elementos del sistema general.

Arduino IDE

Aguayo (2025) comenta que Arduino IDE es un entorno de desarrollo integrado (integrated development environment) que cuenta con las herramientas necesarias para que los desarrolladores programar y cargar el código para que el microcontrolador funcione como se requiere. Inicialmente fue creado para las placas Arduino, pero al ser un software de código abierto permite la interacción con distintas placas de diversos fabricantes que permitan el código abierto. Este entorno de desarrollo permite escribir, compilar y grabar el código de forma sencilla por lo que es una gran característica para su uso. Para programar el microcontrolador del proyecto se eligió el Arduino IDE ya que solo basta con descargar las librerías compatibles con la ESP32, donde están todas las características y funciones para desarrollar este código específico.

Figura 3

Entorno de Programación Arduino

The image shows a screenshot of the Arduino IDE software interface. At the top, there is a menu bar with options: 'Archivo', 'Editar', 'Programa', 'Herramientas', and 'Ayuda'. Below the menu bar is a toolbar with icons for file operations. The main workspace displays the 'Blink' example code. The code is as follows:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
}
```

Nota. Tomado de Arduino.cl

¿Qué es MIT App Inventor?

Según la página oficial de MIT App Inventor, lo define como un entorno de programación visual muy intuitivo que permite crear aplicaciones totalmente funcionales para teléfonos inteligentes y tabletas. El entorno está basado en la construcción por bloques que facilita la creación de aplicaciones complejas mucho más rápido que los entornos de programación tradicionales. Según la página oficial MIT App Inventor tiene un valor social al fomentar de manera libre el desarrollo de tecnología para que las personas pasen del consumo de tecnología a la creación de tecnología.

El entorno de desarrollo de aplicaciones para móviles que funciona a través de internet de forma gratuita está sustentado por un pequeño grupo de estudiantes y personas del laboratorio de ciencias de la computación e inteligencia artificial del instituto de tecnología de Massachusetts, dirigido por el profesor Hal Abelson.

Metodología

El presente proyecto adopta una metodología de investigación aplicada de tipo experimental, ya que, como señalan Hernández-Sampieri, Fernández-Collado y Baptista (2014), este tipo de investigación busca resolver problemas prácticos mediante la utilización del conocimiento científico para intervenir y generar mejoras en contextos específicos. En línea con este planteamiento, se emplea un enfoque metodológico mixto que combina métodos cuantitativos y cualitativos, lo cual permite evaluar tanto el desempeño técnico del prototipo como las percepciones y experiencias de los usuarios durante su implementación.

Para el desarrollo del prototipo, se aplicaron métodos cuantitativos enfocados en los cálculos técnicos que permitieron evaluar su rendimiento a través de datos objetivos y replicables. Estos datos incluyen parámetros clave como el tiempo de respuesta y el consumo energético, entre otros. Paralelamente, se emplearon métodos cualitativos para captar y analizar las percepciones y experiencias de los usuarios, considerando aspectos como la usabilidad del sistema, la satisfacción del usuario y la aceptación general del dispositivo. Estos elementos resultan esenciales para asegurar la implementación efectiva de soluciones tecnológicas en contextos reales.

Para alcanzar los objetivos específicos planteados en el proyecto, se llevaron a cabo las siguientes fases metodológicas:

Fase 1 – Selección de Componentes

Se realizó una revisión técnica para identificar los microcontroladores, sensores, motores y materiales más adecuados para el sistema. La selección se basó en criterios de compatibilidad, eficiencia, disponibilidad y costo, apoyándose en documentación técnica, fichas de fabricantes y proyectos de referencia.

Fase 2 – Diseño del Sistema

En esta etapa se desarrollaron los esquemas electrónicos, diagramas de conexión, modelado 3D de la estructura y programación del sistema de control remoto. Se integraron el hardware y el software, asegurando la coherencia funcional entre los distintos subsistemas mediante simulaciones previas.

Fase 3 – Implementación del Prototipo

Se ensamblaron los componentes seleccionados y se construyó físicamente el dispositivo. También se integró el sistema de control programado, permitiendo validar la operatividad general del diseño en condiciones reales de uso.

Fase 4 – Evaluación y Análisis

Se realizaron pruebas para verificar el correcto funcionamiento del prototipo bajo control local y remoto mediante la aplicación móvil, evaluando la precisión en la dispensación de alimento, la estabilidad de la conexión y la respuesta del sistema. Las pruebas generales se hicieron desde diversas ubicaciones geográficas para determinar el alcance y la fiabilidad del control remoto a diferentes escalas. También se validó la usabilidad de la interfaz con usuarios en distintas ubicaciones. Los resultados mostraron un desempeño estable, una dosificación precisa y una experiencia satisfactoria, confirmando la viabilidad técnica y funcional del sistema.

Desarrollo Ingenieril

Selección de Componentes

En esta parte del proyecto se describen los elementos estructurales, mecánicos y físicos, que comprenden todos aquellos componentes relacionados con la estructura, el mecanismo y la integridad física del sistema. Por otro lado, se abordan los elementos electrónicos y de control, los cuales permiten la ejecución de comandos y la automatización del funcionamiento a través de los actuadores, microcontroladores y sistemas de comunicación.

Elementos Estructurales, Mecánicos y Físicos

Recipiente de Almacenamiento. El recipiente de almacenamiento se basa en un pote redonde de plástico transparente con una capacidad de 2.8 litros, fabricado por la reconocida marca colombiana Plásticos Kendy SAS. Se eligió por su diseño, el costo y el material de fabricación, el cual es considerado apto para el depósito del alimento de la mascota. Según el símbolo impreso en el pote, está moldeado a partir de Polietileno de Baja Densidad (LDPE).

Cuenta con aprobación por las autoridades reguladoras como un material seguro para el contacto directo con productos alimenticios, lo que minimiza el riesgo de contacto de sustancias químicas que pudieran afectar la salud de la mascota. Según Maldonado Álvarez(2023), El LDPE es una variante de polímero semirrígido que se caracteriza por su bajo costo. Presenta buena resistencia a los alcoholes y ácidos, lo cual lo hace atractivo para la producción de nuestro prototipo. También presenta resistencia al impacto temperatura ambiente, lo que asegura la durabilidad del recipiente ante posibles golpes o caídas accidentales. Es importante destacar su resistencia a la intemperie, aunque el prototipo esté diseñado para uso en interiores, esta propiedad añade una capa adicional de durabilidad. El LDPE también presenta excelentes

propiedades de aislamiento eléctrico y una absorción de agua muy baja, lo que contribuye a mantener el alimento seco y protegido.

Figura 4

Pote de Plástico Redondo



Nota. Pote de plástico redondo [Fotografía]. Autoría propia

Dispositivo de Dosificación. Este dispositivo ha sido diseñado para la dispensación controlada de alimento para mascotas, con el objetivo de liberar cantidades predefinidas de manera precisa y confiable. El sistema se compone principalmente de dos partes fundamentales que trabajan en conjunto:

Semicodo con Tapa Reguladora. Es el conducto por donde circula el alimento desde el recipiente de almacenamiento, tiene una apertura de 2 pulgadas y es de PVC. Se eligió un semicodo por la forma de su curva y su ángulo de inclinación, lo que facilita que el alimento fluya fácilmente por gravedad, minimizando la resistencia lo que reduce posibles acumulaciones

en las esquinas. En la salida de este semicodo se encuentra la tapa reguladora, la cual está fabricada en aluminio de 2 milímetros de espesor que cubre una parte de la apertura. La abertura de dosificación tiene forma triangular, la cual fue calibrada mediante distintos ensayos, permitiendo un flujo constante y limitado de alimento, actuando como una restricción para controlar la tasa de paso del alimento.

Figura 5

Semicodo con Tapa Reguladora



Nota. Semicodo con tapa reguladora en aluminio [Fotografía]. Elaboración propia

Paleta de Flujo. El sistema cuenta con un elemento fundamental para regular el paso del alimento, denominado paleta de flujo. Consiste en una lámina plana de aluminio de 2 milímetros de espesor, material seleccionado por su resistencia a la oxidación, asegurando un contacto seguro con el alimento para mascotas. funciona como una compuerta móvil, encargada de permitir o bloquear el flujo del alimento dentro del sistema.

la paleta está anclada al brazo de plástico del servomotor, lo cual permite que sea desmontable para facilitar su limpieza o reemplazo. El montaje se realiza mediante la alineación

de pequeños orificios presentes en la lámina de aluminio con los huecos correspondientes ubicados en el brazo de plástico blanco del servomotor. A través de estos huecos, se pasan unos pequeños tornillos que aseguran la paleta y el brazo.

Cuando el sistema de control envía una señal, el servomotor realiza una rotación precisa de 30 grados en sentido antihorario durante un segundo. Este movimiento, transmitido a través del brazo de plástico, desplaza la paleta, abriendo el conducto del semicodo y permitiendo que el alimento caiga hacia el recipiente de la mascota. Transcurrido ese segundo, el servomotor revierte su rotación, cerrando el conducto mediante la paleta y deteniendo el flujo hasta la siguiente activación.

Figura 6

Paleta de Flujo



Nota. Paleta reguladora [Fotografía]. Autoría propia

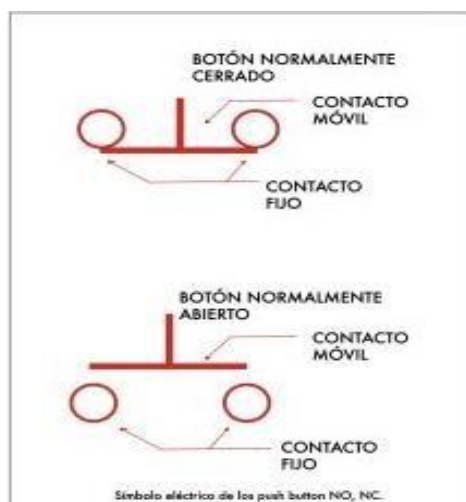
Elementos Electrónicos y de Control

Pulsador. Según Rodríguez (2024) son dispositivos electromecánicos cuya función es abrir o cerrar un circuito mediante una acción de presión. Su principio de funcionamiento

consiste en que, al presionar el botón, el contacto cambia de estado y al liberar la presión vuelve a su estado inicial. Por lo general son normalmente abiertos (NO) o cerrados (NC).

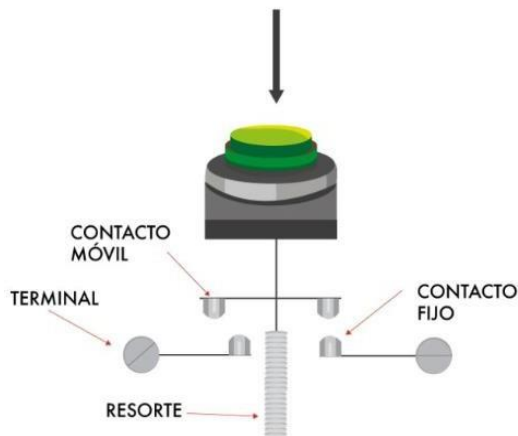
Figura 7

Funcionamiento del Pulsador



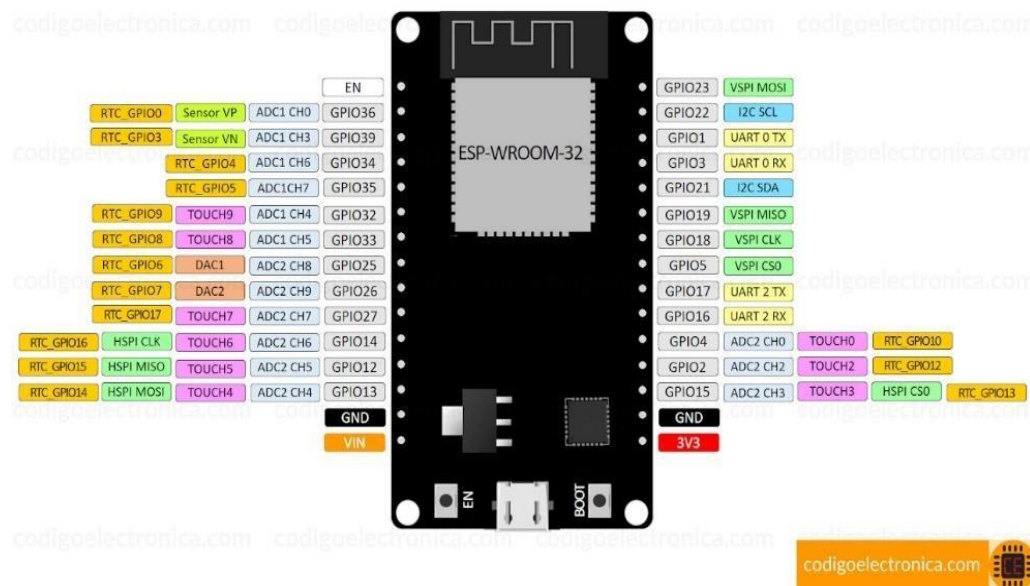
Nota. Símbolo eléctrico y funcionamiento del botón NC y NO [Imagen]. Tomado de <https://masam.cuautitlan.unam.mx/dycme/ce/push-buttons/>

En este caso nos interesa en uso del pulsador normalmente abierto (NO) que como su nombre lo indica su estado inicial es abierto, y al recibir la presión permite el paso de la corriente momentáneamente mientras este presionado, cuando es liberado de la presión mecánica se corta el paso de la corriente. Por este motivo se incluye en el proyecto, con el fin de enviar un pulso para que se interprete como el inicio de una instrucción, en este caso la de poner en marcha el servomotor.

Figura 8*Pulsador Normalmente Abierto (NO)*

Nota. Funcionamiento mecánico botón normalmente abierto NO [Imagen]. Tomado de <https://masam.cuautitlan.unam.mx/dycme/ce/push-buttons/>

ESP32-WROOM-32. Fue desarrollado por la compañía china Espressif Systems según la hoja de datos de la página oficial cuenta con un procesador doble núcleo a 240 MHz como máximo, memoria de 520 kB, conectividad inalámbrica Wi-Fi: 802.11 b/g/n y bluetooth versión 4.2. debido a sus características de conexión a internet y bajo consumo de recursos se consideró que es la más apta para el desarrollo del presente proyecto.

Figura 9*Pines Tarjeta ESP32*

Nota. Distribución de pines ESP32 [Imagen]. Tomado de

<http://codigoelectronica.com/blog/hoja-datos-esp32>

Servomotor MG90S. Según Ricardo (2022) el servomotor es un mecanismo capaz de convertir energía eléctrica en energía mecánica, con una gran precisión y gran torque. Es muy utilizado en la industria en la robótica por su fiabilidad y precisión, ya que se puede controlar su velocidad y posición de manera exacta. Se compone por un motor de corriente continua, aunque también en la industria existen unos de corriente alterna, una serie de engranajes que reducen la velocidad y aumentan el torque, el sistema de control que envía la información al motor para que se mueva en la dirección deseada y un enconder que se encarga de retroalimentar la posición del motor.

Figura 10

Servo MG90S



Nota. Dimensiones del servomotor MG90S[imagen]. Tomado de

<https://www.alldatasheet.com/datasheet-pdf/download/1132104/ETC2/MG90S.html>

Para el desarrollo del proyecto, se ha seleccionado el servomotor MG90s, el cual, de acuerdo con su hoja de datos (ETC., s.f.), presenta las siguientes características relevantes:

Tabla 1

Características Principales del Servomotor

| Características principales servomotor MG90s | |
|--|---|
| Peso | 9 gramos |
| Dimensiones | 22.2 * 11.8 * 31 mm |
| Torque | 1.8 kg/cm |
| Velocidad de Operación | 0.1 segundos por cada 60 grados de rotación |
| Voltaje de alimentación recomendado | 5V |

| | |
|--|--------------|
| Rango de temperatura de funcionamiento | 0° C a 55° C |
|--|--------------|

Nota. Elaboración propia.

Adaptador de Corriente Alterna. Según Boylestad y Nashelsky (2013), el adaptador de corriente es un componente esencial en cualquier sistema electrónico, ya que se encarga de transformar la corriente alterna (CA) de alto voltaje proveniente de la red eléctrica en corriente continua (CC) de bajo voltaje, segura y estable. Este tipo de corriente es la que requieren la mayoría de los dispositivos electrónicos para funcionar correctamente y sin riesgo de fallos. Una de las cualidades más importantes es que el adaptador entrega una salida constante y confiable, protegiendo así los componentes del circuito frente a variaciones de energía que podrían dañarlos.

En el contexto de este proyecto, el adaptador no solo cumple con la función de alimentar el sistema, sino que también representa una solución práctica y segura, ya que se reutiliza un modelo previamente existente, que cumple con normativas internacionales como la certificación UL y la directiva RoHS. Esto garantiza su seguridad y, al mismo tiempo, contribuye a un diseño más sostenible al reducir el consumo de nuevos materiales.

Gracias a este adaptador, el microcontrolador ESP32 y los demás componentes electrónicos del dispensador pueden operar de forma estable, eficiente y segura, permitiendo que el sistema cumpla sus funciones sin interrupciones. A continuación, se relacionan sus características:

Tabla 2*Características Principales del Adaptador de Corriente*

| Características principales Adaptador de corriente alterna | |
|--|---|
| Fabricante | Kanghao Technology Co., Ltd. |
| Modelo | C-P56 SPN5947A |
| Entrada | 100–240 V ~ 50/60 Hz 0.13 A |
| Salida | 5.0 V --- 1.0 A |
| Certificaciones | UL (Underwriters Laboratories), RoHS (Restriction of Hazardous Substances) |

Nota. Elaboración propia.

Motor Vibratorio Mini: según Ineed Electronics (2022), es un actuador ERM (*Eccentric Rotating Mass*) que utiliza un motor de corriente continua (CC) con una masa excéntrica en su eje. Al aplicar un voltaje de 3V a 6V, el motor gira y la masa genera una fuerza centrífuga asimétrica que produce vibraciones.

Por su bajo consumo y tamaño compacto, es ideal para dispositivos portátiles. En el proyecto, este motor se emplea para generar vibraciones que evitan la compactación y obstrucción del alimento dentro del conducto del dispensador. Esto asegura un flujo constante y eficiente, incluso con croquetas de diferentes tamaños y formas, mejorando la fiabilidad del sistema. El motor vibratorio elegido para este proyecto es el modelo 1220, cuyas especificaciones técnicas son proporcionadas por el fabricante:

Tabla 3*Especificaciones Motor Vibratorio*

| Características principales servomotor MG90s | |
|--|--------------------------|
| Cabezal de vibración | 9.5 mm × 11 mm |
| Dimensiones | 22.2 *11.8 * 31 mm |
| Peso | 12 g |
| Rango Voltaje | 3.0 V – 6.0 V CC |
| Corriente | 90 mA a 3V – 160 mA a 5V |

Nota. Datos suministrados por el fabricante. Elaboración propia.

Estos datos reflejan una relación directa entre el voltaje aplicado, la velocidad de rotación y el consumo de corriente. Esta característica permite un control preciso sobre la intensidad de vibración del motor, lo cual resulta clave para la personalización de la experiencia táctil en el prototipo.

Transistor 2N2222. es un transistor bipolar de unión (BJT) de silicio tipo NPN de pequeña señal y propósito general. Este componente es fundamental por su capacidad para operar como un interruptor electrónico. Según su hoja de datos (*Datasheet*), las características más importantes del 2N2222 incluyen su capacidad para soportar hasta 40V entre colector y emisor, una corriente continua de colector (*IC*) de hasta 600mA, y una frecuencia de transición de hasta 300 MHz, lo que lo hace apto para conmutación rápida. Además, su voltaje de saturación colector-emisor es bajo, aproximadamente 0.4V, minimizando las pérdidas de potencia cuando está activado. Para nuestro prototipo, se usó el transistor en una configuración de tipo interruptor para el control eficiente del motor vibratorio.

Diseño del Sistema

En esta fase se diseña tanto la estructura física del dispensador como el sistema electrónico de control. Se define la arquitectura general del sistema, integrando los componentes mecánicos, electrónicos y de software en un prototipo funcional. Esto incluye el diseño de la estructura del dispensador, el circuito electrónico, la implementación en la PCB, la programación del microcontrolador, la lógica de control y la interfaz de usuario, asegurando que todos los elementos trabajen de forma coordinada y eficiente.

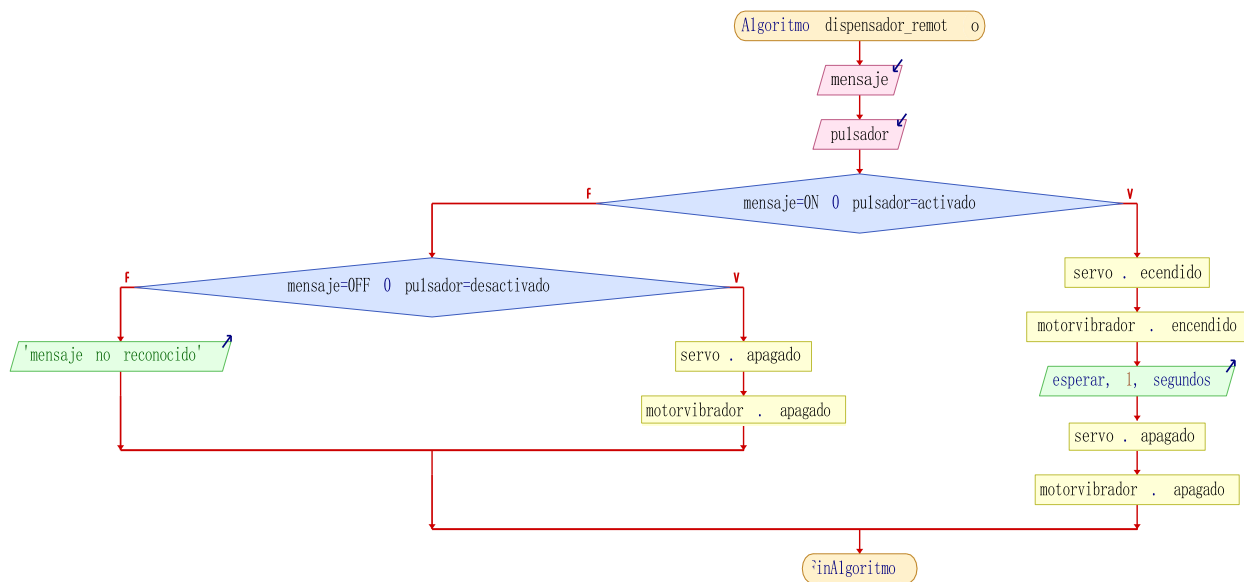
Circuito de Control

El circuito que se ha diseñado tiene como función controlar el mecanismo que se encarga de dispensar la comida y el motor de vibración, a través de una aplicación móvil o localmente mediante un botón físico. El algoritmo básico del circuito coordina la operación del servomotor y motor de vibración, al recibir el mensaje 'ON' a través del servidor MQTT o al activar el botón físico, el servomotor y el motovibrador se activan durante el tiempo ajustado y se apagará automáticamente. Por otro lado, si el mensaje recibido es OFF entonces el servomotor se detendrá inmediatamente.

A continuación, se presenta un diagrama de flujo que explica el algoritmo de una forma clara y ordenada.

Figura 11

Diagrama del Sistema



Nota. Elaboración propia

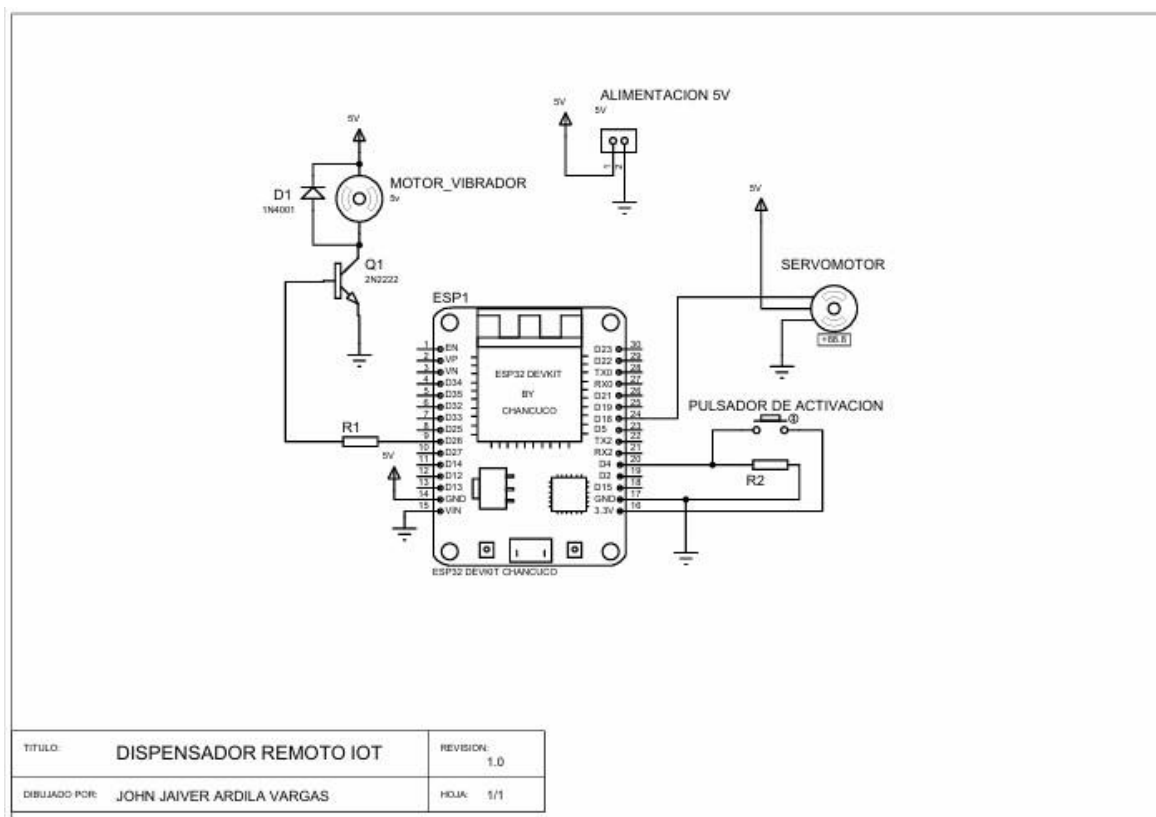
Plano Electrónico del Circuito

En el plano se plasma la idea inicial del circuito electrónico, el cual sirve como guía detallada para el montaje y la conexión correcta de los componentes electrónicos. Este diseño esquemático se realizó utilizando el software Proteus, una herramienta utilizada para la simulación y diseño de circuitos electrónicos.

En el plano podemos relacionar las siguientes conexiones:

- **Conexión del Pulsador.** Uno de los pines del pulsador se conecta al pin 4 del Esp32, y el otro pin se conecta a la fuente de 3.3 voltios del ES32. Además, se incluye una resistencia de 330 Ohmios conectada a tierra y al nodo del pin 4 estableciendo la conexión llamada *pull-down* para asegurar es estado lógico bajo o 0, cuando el pulsador este inactivo, esto evita lecturas erróneas.

- **Servomotor.** El servomotor cuenta con tres pines, señal, VCC y GND. El pin de VCC se conecta a alimentación de 3.3v de la ESP32, el pin de GND va conectado a tierra, y el pin de señal se conecta al pin 18 del ESP32.
- **Motor vibratorio.** El motor vibratorio de 5 V requiere una corriente de 160 mA, valor que excede la capacidad de los pines del ESP32, los cuales solo suministran hasta 40 mA y operan a 3.3 V, por lo que no es posible conectarlo directamente. Además, al ser una carga inductiva, el motor genera picos de voltaje al apagarse que pueden dañar el microcontrolador. Para resolver estas limitaciones, se utiliza un transistor NPN (Q1) como interruptor de potencia: el motor se conecta entre la fuente de 5 V y el colector del transistor, mientras que el emisor va a tierra (GND). La base del transistor recibe la señal de control del pin D26 del ESP32 a través de una resistencia (R1), permitiendo activar el motor con una baja corriente de control. Para proteger el circuito de los picos de voltaje generados por la inductancia del motor, se coloca un diodo de protección (D1) en paralelo con el motor, en polarización inversa, evitando daños en el transistor y el ESP32.

Figura 12*Plano Electrónico del Sistema*

Nota. Plano electrónico de conexiones [Imagen]. Elaboración propia.

Cálculo de Resistencia Base

Para determinar el valor adecuado de la resistencia de base (R1), se realiza el siguiente cálculo, basado en las especificaciones del motor y el transistor obtenidas de sus respectivas hojas de datos:

Calculo Corriente Motor Vibratorio. A continuación, se calculará la corriente de operación del motor vibratorio considerando un margen adicional del 25 %. Este margen permite cubrir posibles incrementos de corriente debido a cargas mecánicas leves o variaciones en las condiciones de funcionamiento. Teniendo en cuenta que la corriente sin carga del motor es

aproximadamente 160 mA a 5 V (0.16 A), la corriente de diseño se determina de la siguiente manera:

$$I_c = I_{Csin_carga} + (0.25 \times I_{Csin_carga}) = 0.16A + (0.25 \times 0.16A) = 0.20A$$

Cálculo de la Corriente de Base Necesaria (I_b). Para que el transistor funcione correctamente como un interruptor, es decir, operando en saturación, debe recibir una corriente de base suficiente que le permita conducir la corriente requerida por el motor vibratorio. Este valor depende de la corriente de colector (I_c) y de la ganancia de corriente continua (hFE) del transistor. Según la hoja de datos del 2N2222, la ganancia hFE varía entre 100 y 300; para asegurar el funcionamiento incluso en el peor caso, se adopta el valor mínimo:

$$hFE_{min} = 100$$

La corriente mínima necesaria en la base será:

$$I_b = \frac{I_c}{hFE_{min}} = \frac{0.20 A}{100} = 0.002 A = 2 mA$$

Por lo tanto, se requiere una corriente de base de 2 mA para que el transistor 2N2222 opere en saturación y actúe eficazmente como un interruptor, permitiendo conducir los 200 mA requeridos por el motor vibratorio desde una fuente externa bajo el control del ESP32.

Cálculo de la Resistencia de Base ($R1$). Una vez determinada la corriente de base requerida (I_b), se procede a calcular el valor de la resistencia de base ($R1$) que permita suministrar dicha corriente desde un pin digital del ESP32, que entrega un voltaje de salida de 3.3 V; para ello, se considera la caída de voltaje entre la base y el emisor del transistor, conocida como V_{BE} , que típicamente es de 0.7 V para el 2N2222. Por lo tanto, la tensión efectiva sobre la resistencia de base es:

$$R1 \approx (V_{cc} - V_{be}) / I_b$$

$$R1 \approx (3.3V - 0.7V) / 0.002 A \approx 1300 \Omega$$

El cálculo indica que la resistencia de base (R1) debe ser aproximadamente 1.3 kΩ, lo que permitirá suministrar cerca de 2.0 mA de corriente de base.

Diseño de la PCB

El diseño de la Placa de Circuito Impreso (PCB) es una etapa crucial donde el esquemático electrónico toma forma física. En este proyecto, la PCB fue diseñada utilizando el software Proteus (módulo ARES), una herramienta integral para el diseño y simulación de circuitos electrónicos. El proceso comenzó con la importación del archivo generado previamente en el módulo ISIS de Proteus, el cual contenía toda la información de los componentes y sus interconexiones lógicas.

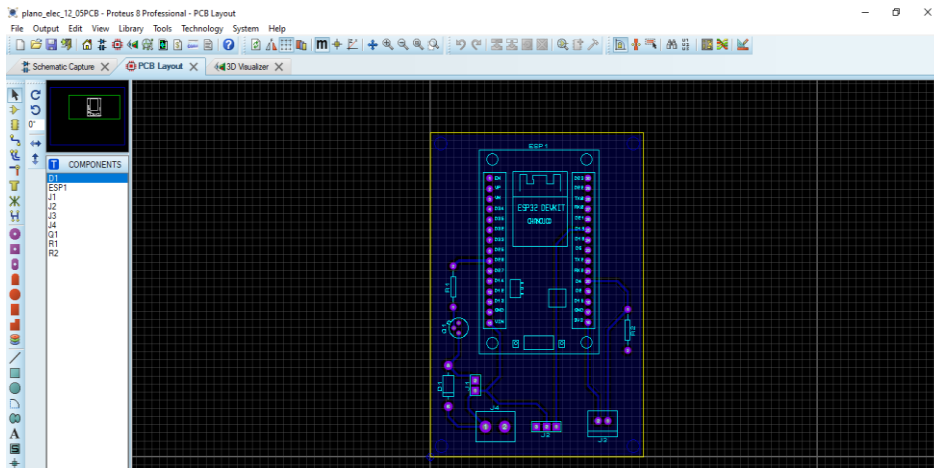
Durante el diseño en ARES, se aplicaron principios fundamentales para optimizar la funcionalidad y la manufactura de la placa. Se realizó una cuidadosa ubicación de los componentes, buscando minimizar la longitud de las pistas, optimizar el espacio disponible y orientar los elementos para facilitar el proceso de soldadura. Además, se consideró la separación adecuada entre componentes para evitar interferencias térmicas, crucial para la fiabilidad a largo plazo del dispositivo.

Posteriormente, se llevó a cabo el enrutamiento de las pistas, que interconectan eléctricamente los componentes según el esquemático. Un aspecto clave de este proceso fue la incorporación del plano de tierra (GND), que ayuda significativamente a reducir el ruido eléctrico y a mejorar la distribución de energía en toda la placa. El resultado final fue un archivo de diseño de PCB completo, listo para su fabricación, que incluía el trazado de las pistas, la

definición de las capas y toda la información necesaria para la producción física de la placa. Las figuras a continuación muestran el diseño final y una visualización 3D de la PCB.

Figura 13

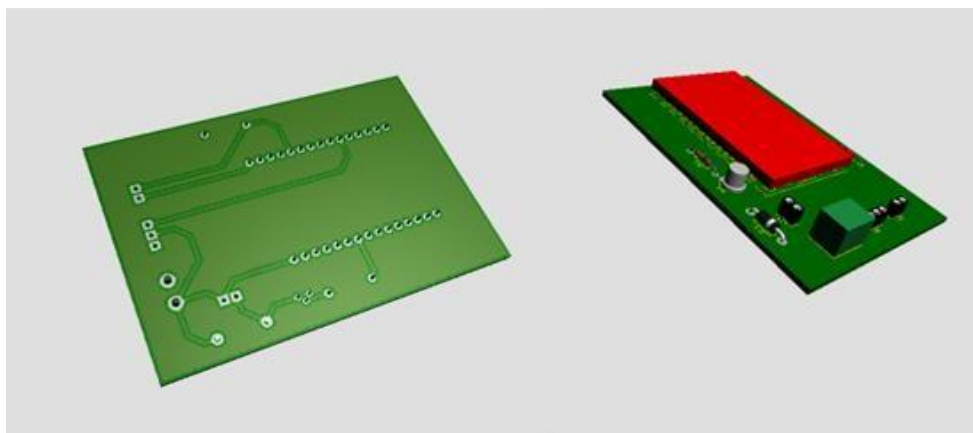
Diseño Final PCB



Nota. Diseño final placa electrónica en ARES [Imagen]. Elaboración propia

Figura 14

Visualización 3D de la placa



Nota. Visualización 3D de la PCB [Imagen]. Elaboración propia

Código Microcontrolador

Para la creación del código se ha seleccionado el entorno de desarrollo integrado (IDE) Arduino por su facilidad de uso y su compatibilidad con la ESP32. Además, por ser una plataforma de código abierto existe una amplia comunidad de colaboradores que aportan librerías y métodos. La simplicidad del lenguaje C ++, el acceso a códigos y librerías de uso libre para implementar con MQTT además de la compatibilidad con la ESP32 la convierten en la mejor herramienta para desarrollo del presente proyecto. A continuación, se muestra el código:

```
#include <WiFi.h>

#include <AsyncMqttClient.h>

#include <ESP32Servo.h>

// Configuración de la red WiFi

const char* ssid = "Ardila";

const char* password = "ardila18*";

// Configuración del servidor MQTT

const char* mqttServer = "broker.emqx.io";

const int mqttPort = 1883;

// Tema MQTT definir temas

const char* servoTopic = "casa/food"; // Este tema recibe los mensajes para controlar el
servo y el motovibrador
```

```
const char* servoStatusTopic = "casa/servo/estado"; //este tema envía mensaje de  
funcionamiento del sistema
```

```
//definir constantes
```

```
const int pinServo = 19; // pin del servomotor
```

```
const int buttonPin = 4; // pin del pulsador
```

```
const int vibraPin = 26; // pin del motovibrador
```

```
// constantes de estado del servomotor
```

```
const int SERVO_ON = 45;
```

```
const int SERVO_OFF = 0;
```

```
// cONFIGURACION DEL TIEMPO
```

```
const unsigned long tiempoApagado = 1000; // Tiempo deseado
```

```
//VARAIBLES GLOBALES
```

```
Servo miServo;
```

```
AsyncMqttClient mqttClient;
```

```
unsigned long tiempoEncendido = 0; // variable para almacenar el tiempo de encendido
```

```
// Variables para el pulsador
```

```
unsigned long lastButtonPress = 0;
```

```
const unsigned long debounceTime = 50; // Tiempo de debounce en milisegundos
```

```
// Conexión WiFi
```

```
void setupWifi() {  
    WiFi.begin(ssid, password);  
    while (WiFi.status() != WL_CONNECTED) {  
        delay(500);  
        Serial.print(".");  
    }  
    Serial.println("\nConectado a la red WiFi");  
}  
  
// Evento al conectar con el servidor MQTT  
void onMqttConnect(bool sessionPresent) {  
    Serial.println("Conectado al servidor MQTT");  
  
    // Suscribirse al tema del servo  
    mqttClient.subscribe(servoTopic, 1);  
}  
  
// Evento al desconectar del servidor MQTT  
void onMqttDisconnect(AsyncMqttClientDisconnectReason reason) {  
    Serial.println("Desconectado del servidor MQTT. Intentando reconectar...");  
    delay(5000); // Espera 5 segundos antes de reconectar  
    mqttClient.connect();  
}
```

```
// Evento para manejar los mensajes recibidos

void onMqttMessage(char* topic, char* payload, AsyncMqttClientMessageProperties
properties, size_t len, size_t index, size_t total) {

    Serial.print("Mensaje recibido en el tema: ");

    Serial.println(topic);

    // Convertir el payload en un String para compararlo fácilmente

    String message;

    for (size_t i = 0; i < len; i++) {

        message += (char)payload[i];

    }

    Serial.print("Contenido del mensaje: ");

    Serial.println(message);

    // Encender o apagar el LED según el mensaje recibido

    if (String(topic) == servoTopic) {

        if (message == "ON") {

            digitalWrite(vibraPin, HIGH);

            miServo.write(SERVO_ON);

            Serial.println("servo encendido");

            tiempoEncendido = millis();

            Serial.println("Publicando en casa/servo/estado: ON"); // Mensaje de depuración
```

```

    mqttClient.publish(servoStatusTopic, 0, true, "ON");
} else if (message == "OFF") {
    digitalWrite(vibraPin, LOW); // Apaga el LED

    Serial.println("servo apagado");

    miServo.write(SERVO_OFF);

    Serial.println("Publicando en casa/servo/estado: OFF");

    mqttClient.publish(servoStatusTopic, 0, true, "OFF");

} else {

    Serial.println("Mensaje no reconocido"); // En caso de mensaje desconocido

}

}

}

void setup() {

    Serial.begin(115200);

    // Configuración del pin del LED como salida

    pinMode(vibraPin, OUTPUT);

    pinMode(buttonPin, INPUT_PULLDOWN); // Configura el pin del pulsador con pull-
down

    miServo.attach(pinServo, 500, 2500);

    // Configurar la conexión WiFi

    setupWifi();

```

```
// Configuración del cliente MQTT y eventos
mqttClient.onConnect(onMqttConnect);
mqttClient.onDisconnect(onMqttDisconnect);
mqttClient.onMessage(onMqttMessage);
mqttClient.setServer(mqttServer, mqttPort);

// Conectar al servidor MQTT
mqttClient.connect();
}
int pos = 0;
void loop(){
  // Manejo del temporizador
  if (digitalRead(vibraPin) == HIGH &&(millis()-tiempoEncendido > tiempoApagado))
  {
    digitalWrite(vibraPin, LOW);
    miServo.write(SERVO_OFF);
    Serial.println("servo apagado");
    Serial.println("Publicando en casa/servo/estado: OFF"); // Mensaje de depuración
    mqttClient.publish(servoStatusTopic, 0, true, "OFF");
  }
  // Manejo del pulsador
  int buttonState = digitalRead(buttonPin);
```

```

if (buttonState == HIGH && (millis() - lastButtonPress > debounceTime)) {
    digitalWrite(vibraPin, HIGH);
    miServo.write(SERVO_ON);
    Serial.println("servo encendido");
    tiempoEncendido = millis();
    mqttClient.publish(servoStatusTopic, 0, true, "ON");
    lastButtonPress = millis();
}
}

```

Para la creación del anterior código, inicialmente se consideró el uso de una librería para establecer la comunicación del bróker con el ESP32. Se encontró que la librería *AsyncMqttClient.h* desarrollada por Marvin Roger (marvinroger, 2023) de GitHub es la más adecuada para establecer la comunicación, ya que tiene capacidad para establecer comunicación asíncrona, permitiendo que el ESP32 ejecute otras tareas al mismo tiempo, optimizando así el rendimiento y evitando bloqueos en la ejecución. El Código describe eventos para la configuración del cliente MQTT y el manejo de eventos de conexión, desconexión, recepción y envío de mensajes. Con el fin de establecer una comunicación bidireccional se asignaron dos tópicos para la publicación y suscripción de mensajes, para la recepción de comandos de control (ON/OFF) para el servo y el motovibrador, se creó el tópico "casa/food" y por otro lado para la publicación del estado del servomotor, se creó el tópico "casa/servo/estado".

El código implementa las siguientes funcionalidades:

Conexión Wifi. El ESP32 establece conexión con la red WiFi utilizando las credenciales provistas (SSID y contraseña).

Conexión MQTT. Se establece una conexión con el broker MQTT "broker.emqx.io" en el puerto 1883.

Control de los Motores. El motovibrador (pin 26) y el servomotor (pin 18) se controlan mediante comandos MQTT recibidos en el tópico "casa/food". Los comandos "ON" y "OFF" activan/desactivan el motovibrador y posicionan el servomotor en ángulos predefinidos: SERVO_ON = 45 grados y SERVO_OFF = 0 grados, respectivamente.

Control Manual. Un pulsador (pin 4) permite encender el servo y el motovibrador, permitiendo la activación del sistema manualmente

Publicación del Estado. El estado del servomotor y motovibrador (ON/OFF) se publica en el tópico "casa/servo/estado".

Se utilizó la librería ESP32Servo.h (madhepha, 2023) para el control del servomotor, debido a la incompatibilidad de la librería estándar de Arduino con el ESP32. Se implementó un temporizador utilizando la función millis (), adaptado de la documentación de Arduino (Arduino, s.f.), para el apagado automático del sistema y el control manual mediante el pulsador.

La comunicación entre la aplicación móvil y el ESP32 se realiza mediante el protocolo MQTT, a través del intercambio de mensajes con el broker MQTT. Desde la aplicación, se puede monitorizar el estado del servomotor (encendido/apagado) y enviar comandos para controlar el encendido del sistema.

Diseño de Aplicación Móvil en MITAPP Inventor

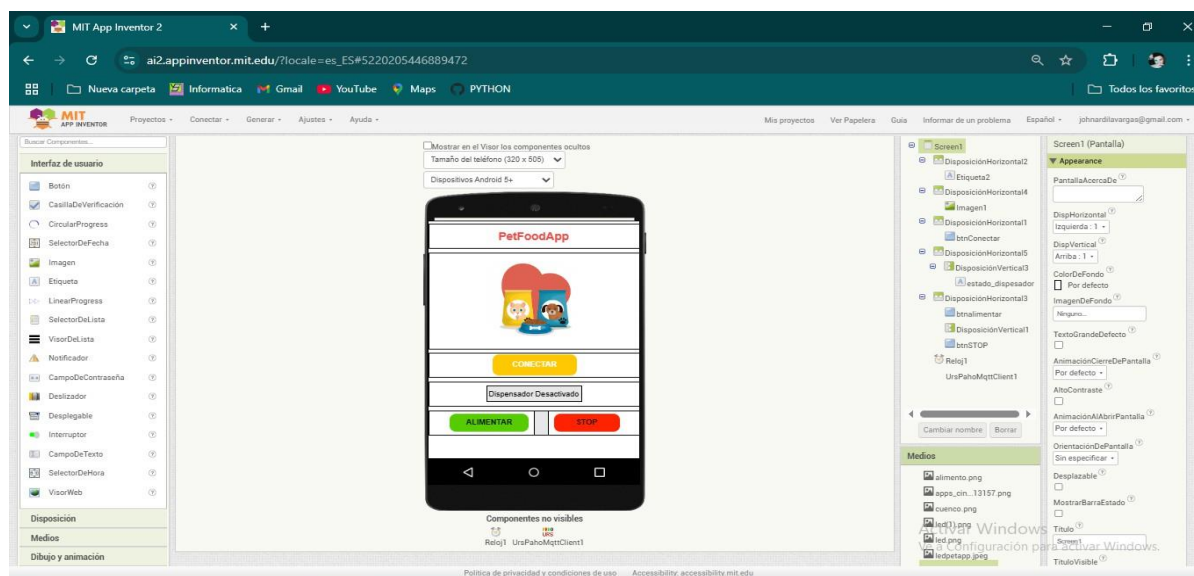
El bróker establece la comunicación entre cliente y el suscriptor en este caso el cliente se conecta al bróker a través de la app móvil y el publicador sería Esp32. Para establecer esta comunicación con el ESP32 se desarrolló una aplicación móvil utilizando MIT App Inventor, un

entorno de desarrollo visual que facilita la creación de aplicaciones para Android. Tiene una interfaz intuitiva y su programación se basa en bloques, lo que permite el desarrollo fácil de aplicaciones funcionales. El entorno de desarrollo se divide en dos secciones principales: el Diseñador y el entorno de Bloques.

En la sección del Diseñador, se construyó la interfaz de usuario de la aplicación. Se seleccionaron componentes visuales como botones, etiquetas e imágenes. Para el título principal se usó el nombre PetFoodApp, el cual es una combinación de las palabras en inglés Pet (mascota), Food (comida) y App (aplicación), reflejando directamente su propósito relacionado con la alimentación de mascotas. El diseño también cuenta con varios botones que al ser pulsados realizan acción, son de diferentes colores para que visualmente se asocien con las diferentes funciones. Un botón de color amarillo permite la conexión con el bróker y cambia de estado 'conectar' a 'conectado' cuando se establece una conexión exitosa con el bróker. Un botón verde llamado alimentar envía en mensaje ON para que se ejecuten las instrucciones en la ESP32, y otro botón rojo llamado STOP detiene cualquier acción enviada antes.

Figura 15

Sección Diseñador Mit App Inventor



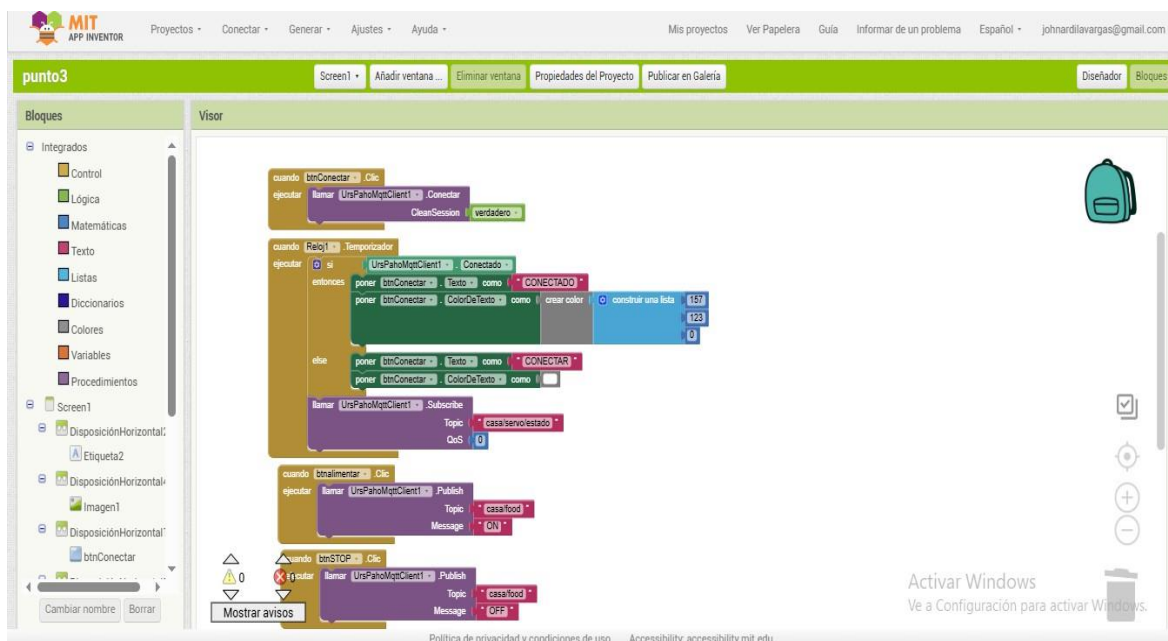
Nota. Entorno de desarrollo de la interfaz gráfica (elaboración propia).

La extensión AI2 MQTT, descargada de la página de Ullis (Ullis Roboter Seite, s.f.), es una biblioteca que permite la comunicación MQTT en aplicaciones desarrolladas con MIT App Inventor. Esta biblioteca contiene los bloques para establecer las funciones de la aplicación diseñada, la extensión AI2 MQTT, a través del componente "UrsPahoMqttClient1", actúa como un cliente MQTT, facilitando el envío de comandos y la recepción de datos de estado del dispositivo. En la sección de Bloques, se implementó la lógica de la aplicación basándose en las funcionalidades proporcionadas por la extensión AI2 MQTT. Se emplearon bloques de control de flujo para gestionar la interacción del usuario y la comunicación MQTT. Los bloques condicionales ("if") implementados verifican el estado de la conexión MQTT y el contenido de los mensajes recibidos. Esto asegura que la aplicación responda adecuadamente a diferentes situaciones, como la conexión exitosa al broker, la recepción de mensajes del dispensador.

La comunicación entre la aplicación creada y el dispensador se basa en el envío de mensajes MQTT a través del bróker. La función "ALIMENTAR" en la aplicación traduce la acción del usuario en un mensaje MQTT con el contenido "ON", enviado al tópico "casa/food". Este mensaje es recibido por el ESP32, que ejecuta el código para activar el servo y liberar la comida. De manera similar, la función "STOP" envía un mensaje "OFF" al mismo tópico, deteniendo el servo. Las interacciones del usuario, como los clics en los botones "ALIMENTAR" y "STOP", se traducen en mensajes MQTT que se envían a través de "UrsPahoMqttClient1". El ESP32 también publica mensajes en el tópico "casa/servo/estado" para informar a la aplicación sobre el estado del dispensador ("ON" o "OFF"). Los datos recibidos del dispensador se procesan y se muestran en la interfaz gráfica de la aplicación, proporcionando retroalimentación al usuario para poder saber en qué momento está trabajando el dispensador.

Figura 16

Diseño de Bloques Mit App Inventor



Nota. Diseño de los bloques MIT App Inventor [Imagen]. (elaboración propia)

Diseño de la Tolva

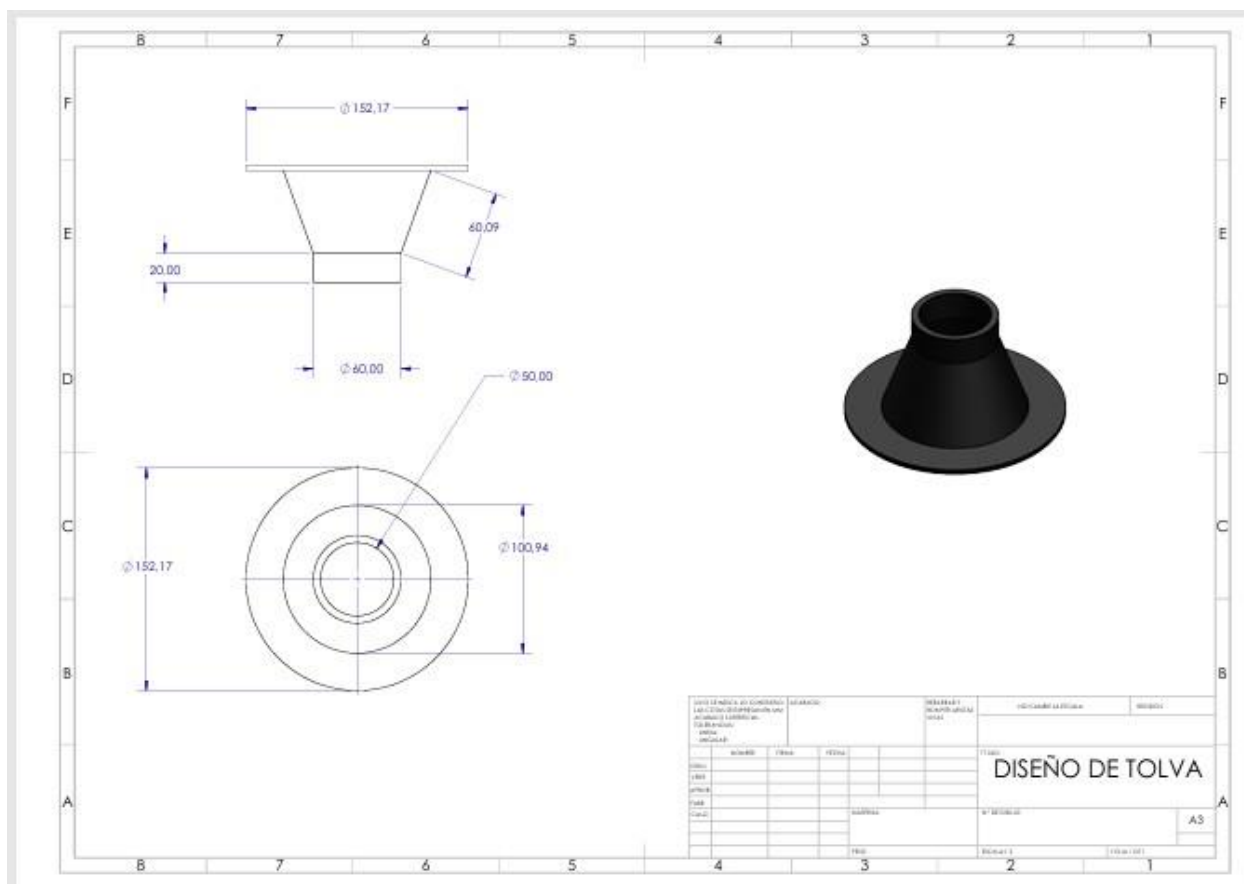
La tolva fue diseñada específicamente para conectar de manera precisa la base del recipiente de almacenamiento con la entrada del semicodo de PVC, que forma parte del sistema de dispensación. Para ello, se tomaron las medidas exactas de ambas partes y se desarrolló el modelo en el software SolidWorks, lo que permitió crear una forma que se ajustara correctamente a los componentes. Durante el diseño, se buscó que la tolva permitiera un flujo continuo y sin obstrucciones del alimento, evitando ángulos cerrados o zonas planas donde las croquetas pudieran atascarse. También se tuvo en cuenta que la parte superior de la tolva coincidiera con la forma del recipiente de almacenamiento, asegurando un buen sellado, mientras que la parte inferior se adaptó a la abertura del semicodo, facilitando así el paso directo del alimento hacia el mecanismo de dispensación. Gracias a este diseño, se logra una conexión

eficiente, sin fugas, estable y fácil de ensamblar. Una vez finalizado el modelo digital, la pieza fue fabricada mediante impresión 3D.

El plano correspondiente al diseño de la tolva se muestra a continuación, donde se pueden apreciar sus dimensiones y forma final.

Figura 17

Plano de la Tolva



Diseño de la Tapa Superior

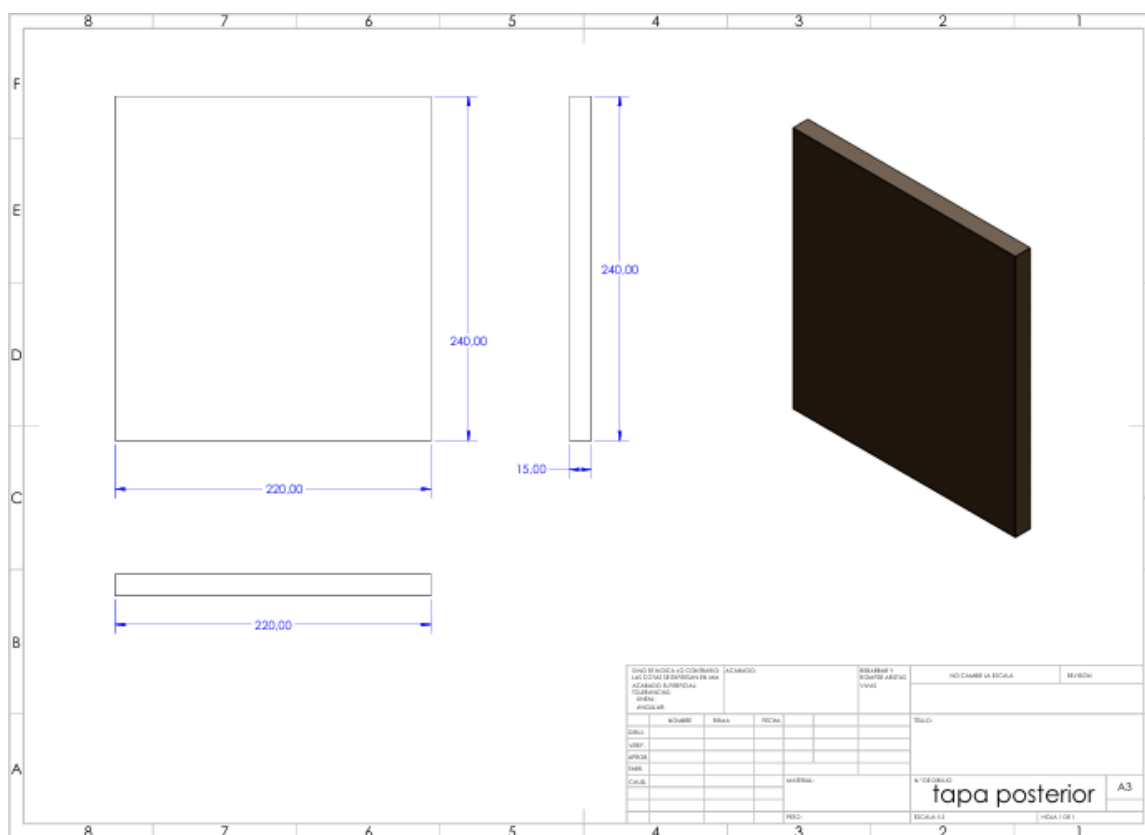
La tapa superior de la carcasa fue diseñada como una pieza clave para el soporte del sistema de dispensación. Tiene una forma cuadrada, con dimensiones de 22 cm x 22 cm, y está fabricada en MDF, al igual que el resto de la estructura. En el centro de esta tapa se realizó un

por 22 cm de ancho, lo que le permite encajar con precisión en la estructura formada por las tapas laterales, la base y la tapa superior. Su diseño contempla el espacio interior necesario para el soporte horizontal donde se encuentra la caja que aloja el circuito impreso, permitiendo que esta tapa pueda retirarse fácilmente en caso de mantenimiento o ajustes. Aunque su función principal es de cierre, también aporta rigidez al conjunto, ayudando a mantener la estabilidad y solidez de la carcasa.

A continuación, se presenta el plano correspondiente al diseño de la tapa posterior, con sus dimensiones:

Figura 22

Diseño Tapa Posterior



Diseño Conceptual del Prototipo

Como parte del desarrollo del dispensador de comida para perros y gatos, se elaboró un modelo 3D conceptual del prototipo utilizando el software SolidWorks. Este diseño incluyó la tolva, las tapas, y una representación del recipiente de almacenamiento con su respectiva tapa.

El objetivo de este modelo fue visualizar la estructura general del dispensador antes de pasar a la fabricación, permitiendo evaluar aspectos estéticos, proporciones, distribución de componentes y compatibilidad entre piezas. La vista integrada del conjunto facilitó la detección de posibles ajustes, asegurando una integración adecuada y un diseño equilibrado.

Además, el modelo conceptual sirvió como una herramienta para anticipar la apariencia final del producto, guiar decisiones de diseño estructural y validar la viabilidad del ensamblaje funcionales, el enfoque fue generar una representación clara y coherente que sirviera como base para el desarrollo de los planos y la posterior fabricación del prototipo.

coincidentes. Para el ensamblaje, se alinearon cuidadosamente estos orificios y se insertaron los tornillos M3 a través de ellos. En cada tornillo, se colocaron arandelas tanto entre la cabeza del tornillo y una de las bridas, como entre la otra brida y la tuerca M3 utilizada para asegurar la unión. El uso de tornillos y tuercas en agujeros pasantes permite que la tolva pueda ser desmontada del recipiente de manera sencilla, facilitando así los procesos de limpieza del dispensador. En la siguiente figura se ve el estable de las piezas.

Figura 24

Ensamble Recipiente de Almacenamiento y Tolva



Nota. [fotografía]. elaboración propia

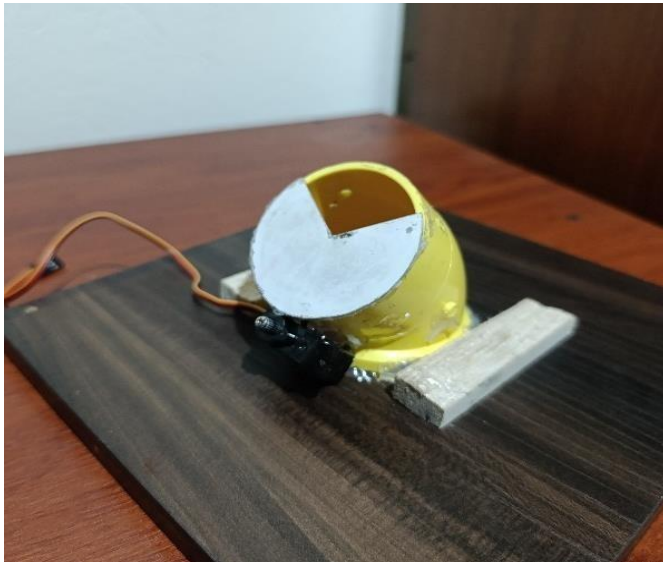
Integración del Dispositivo de Dosificación. El dispositivo encargado de regular el flujo de alimento se integró al prototipo mediante la modificación de un semicodo de PVC y la

incorporación de un servomotor MG90S y una lámina de aluminio con una abertura con forma triangular. El proceso se llevó a cabo de la siguiente manera:

Fijación del Servomotor y la lámina al Semicodo. El servomotor MG90S fue sujeto a la pared exterior del semicodo de PVC utilizando silicona resistente para asegurar una fijación firme y estable. La posición del servomotor se determinó cuidadosamente para que su eje de rotación quedara alineado con la zona donde se controlaría la salida del alimento. Luego, se adhirió con silicona resistente una lámina de aluminio cortada con una abertura triangular para regular el flujo de concentrado a un extremo del semicodo de PVC. Esta lámina se colocó estratégicamente cubriendo la sección transversal, dejando la abertura como el conducto principal de dispensación, asegurando una unión duradera y evitando movimientos durante el uso.

Figura 25

Fijación Lamina y Servomotor



Nota. Fijación de la lámina con abertura triangular y el servomotor al semicodo. [fotografía].

elaboración propia.

Fijación de Motovibrador: Un pequeño motor vibrador se fijó cuidadosamente en un codo del conducto de salida para optimizar la transmisión de suaves vibraciones. La sujeción se realizó con una lámina metálica adaptada y tornillos de acero inoxidable asegurados con tuercas. La parte activa del motor se atornilló firmemente a la pared interior del codo con tornillos resistentes a la corrosión, logrando una conexión sólida para propagar la vibración al alimento.

Al activarse, el motor vibrador genera una vibración de baja intensidad que se transmite al codo. Esta sutil vibración busca minimizar la compactación y el apelmazamiento del alimento, especialmente en la curva propensa a obstrucciones, promoviendo un descenso uniforme y reduciendo la necesidad de intervención manual. La solidez de la fijación asegura la operatividad continua y fiable del dispensador

Figura 26

Soporte Motor Vibratorio



Nota. Sujeción del motovibrador al semicodo de dispensación [Fotografía]. Elaboración propia

Integración del Mecanismo de Control (Paleta de Flujo). El mecanismo de control, o paleta de flujo, se integró directamente al eje del servomotor SG90 con la finalidad de controlar

el paso del alimento a través de la abertura con forma triangular presente en la lámina de aluminio. Inicialmente, se seleccionó uno de los brazos plásticos que se incluyen de fábrica el cual sirvió como estructura principal para la paleta.

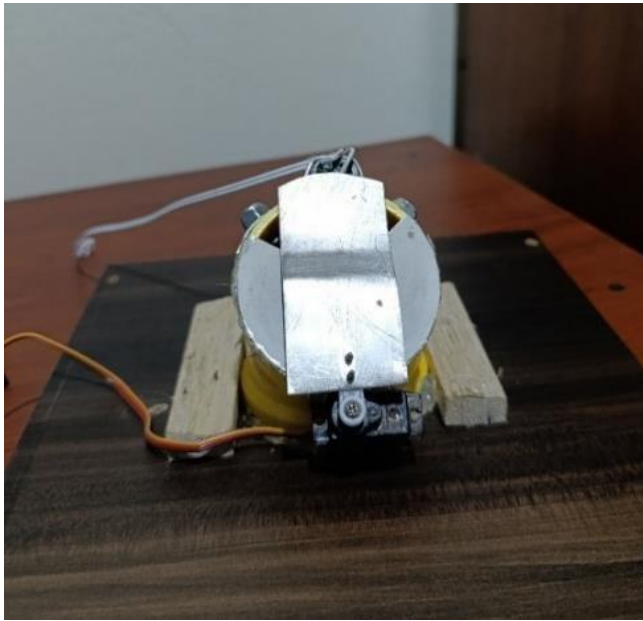
Figura 27

Brazo Plástico Servomotor



Nota. Esta pieza viene con el servomotor [Fotografía]. Elaboración propia

Posteriormente, se preparó la lámina de aluminio que contiene la abertura con forma triangular. El brazo se posicionó cuidadosamente sobre la lámina y se unieron ambas piezas mediante la inserción de pequeños tornillos que atravesaron la lámina y se fijaron al brazo. Finalmente, el brazo del servomotor, ya ensamblado a la lámina, se insertó en el eje estriado del servomotor MG90s, asegurándose con un tornillo que venía incluido, tal como se aprecia en la Figura 22. De esta manera, el movimiento rotatorio del servomotor permite que la paleta se posicione para permitir o bloquear el flujo de alimento a través de la abertura con forma triangular, dosificando así la cantidad deseada.

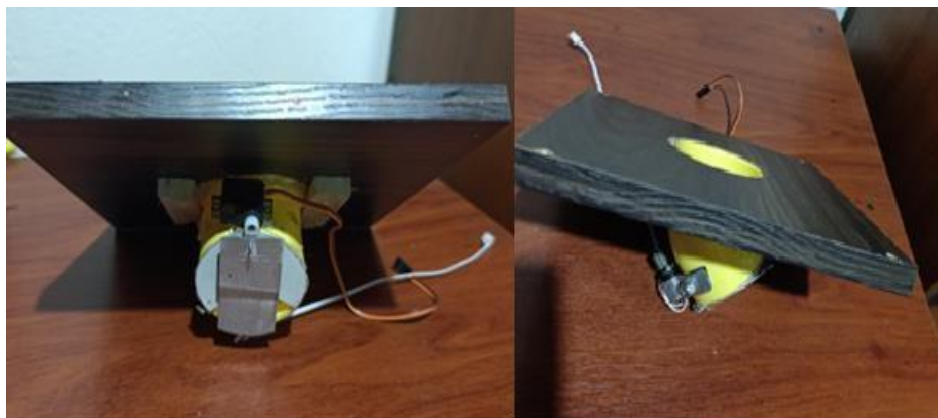
Figura 28*Acople de la Paleta de Flujo*

Nota. La paleta de flujo se fija con un tornillo al eje del servomotor [Fotografía]. Elaboración propia

Fijación del Dispositivo de Dosificación a la Base de MDF: Para proporcionar un soporte estable al dispositivo de dosificación, se utilizó una base cuadrada de madera MDF con un agujero central. El semicodo de PVC se insertó a través de este agujero y se fijó con silicona resistente. Es fundamental comprender que la base de MDF, una vez unida al dispositivo de dosificación, conformará una unidad que será fácilmente desmontable de la carcasa principal del dispensador para facilitar el mantenimiento.

Figura 29

Fijación del Dispositivo de Dosificación a la Base



Nota. Al unir queda una unidad desmontable. [Fotografía]. Elaboración propia

Impresión Tolva . La función principal de la tolva es establecer una conexión eficiente entre la salida del recipiente de almacenamiento y la entrada del dispositivo de dispensación (semicodo de PVC), que constituye una parte fundamental del dispositivo de dispensación. El diseño de la tolva asegura un flujo de alimento suave y constante, evitando la formación de ángulos pronunciados o superficies irregulares que pudieran generar atascos en el flujo del concentrado.

Para su diseño, se tomaron como referencia las medidas previamente definidas de la base del recipiente de almacenamiento y la boca del semicodo de PVC. Este proceso se realizó utilizando el software de diseño asistido por computadora SolidWorks, lo que permitió ajustar con precisión la geometría de la tolva a los elementos conectados. Una vez finalizado el diseño, se procedió a su fabricación mediante impresión 3D.

La forma específica de la tolva se adapta a la geometría del pote de almacenamiento en su parte inferior, evitando fugas de alimento o la entrada de contaminantes. Esta conexión se diseñó

para asegurar la continuidad del conducto de alimentación, manteniendo un flujo ininterrumpido hacia el mecanismo de dispensación.

La tolva está fabricada mediante impresión 3D con PLA (Ácido Poliláctico) como material. La elección del PLA se fundamenta en su seguridad para el contacto con alimentos, una característica esencial dado que esta pieza estará en contacto directo con las croquetas de la mascota durante su tránsito del almacenamiento al sistema de dispensación. Como señalan Serna, Rodríguez de S. y Albán (2003) en su trabajo académico sobre el Ácido Poliláctico (PLA), este biopolímero termoplástico es ampliamente utilizado en diversas industrias, incluyendo la alimenticia, donde se emplea en la producción de envases y empaques para alimentos. Esta versatilidad y su aplicación directa en el sector alimentario respaldan la idea de uso del PLA para la fabricación de la tolva de nuestro dispensador. Por otro lado, al ser uno de los filamentos más asequibles para la impresión 3D, permite mantener bajos los costos de fabricación.

Figura 30

Tolva Impresa en 3D



Nota. Tolva impresa en 3D [Fotografía]. Autoría propia

Unión de la Tolva de Conexión al Dispositivo de Dosificación. El extremo inferior de la tolva de conexión se unió al extremo superior del semicodo de PVC mediante un ajuste a presión. Este diseño permite una conexión firme para el funcionamiento y un desmontaje sencillo para la limpieza de ambos componentes. La precisión en las dimensiones asegura la estabilidad de la unión.

Figura 31*Acople de Tolva Y Semicodo*

Nota. Ensamble de la tolva y semicodo. [Fotografía]. Elaboración propia

Ensamblaje de la Carcasa de MDF. La carcasa del dispensador de alimento para mascotas fue construida utilizando paneles de madera MDF, ensamblados para formar una estructura que alberga y protege los componentes mecánicos y electrónicos del sistema. El diseño de la carcasa se centró en la funcionalidad, la estabilidad y la facilidad de acceso para el mantenimiento.

La tapa superior de la carcasa, una pieza cuadrada de 22cm x 22cm, fue diseñada para ser removible. Esta tapa tiene la función de soportar y fijar el dispositivo de dispensación en su parte inferior. Su fijación al resto de la carcasa se realizó mediante cuatro tornillos ubicados en las esquinas, permitiendo su fácil desmontaje para tareas de limpieza o mantenimiento del sistema de dispensación y del recipiente de almacenamiento de alimento.

Figura 32*Tapa Superior*

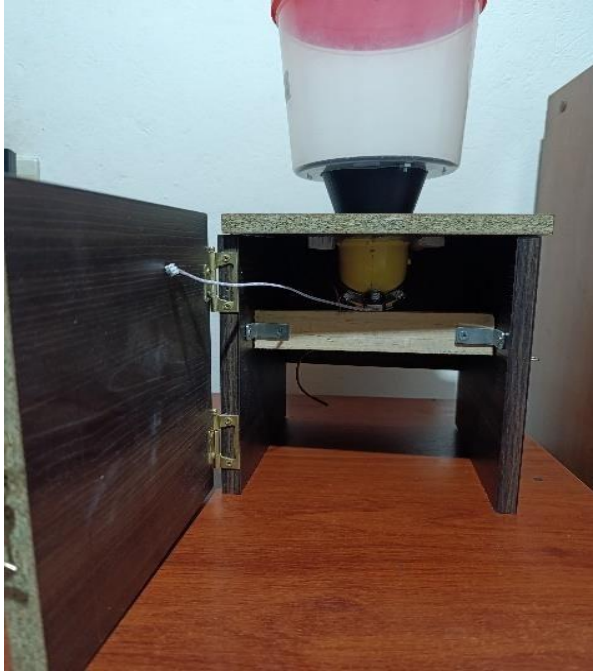
Nota. Ensamble de la tolva y semicodo. [Fotografía]. Elaboración propia.

El panel frontal de la carcasa incorpora una apertura rectangular en la parte inferior. Esta abertura fue diseñada específicamente para permitir la colocación del tazón donde se recolectará el alimento dispensado, asegurando que el concentrado caiga directamente en el recipiente destinado para la mascota.

Figura 33*Panel Frontal*

Nota. El panel tiene una abertura para la caída del alimento. [Fotografía]. Elaboración propia.

Para facilitar el acceso a los componentes electrónicos que controlarán el funcionamiento del dispensador, se implementó una tapa posterior con bisagras. Esta tapa se encuentra unida a la tapa lateral derecha mediante bisagras, lo que permite abrir y cerrar el compartimento electrónico de manera rápida y sencilla, sin necesidad de desmontar paneles completos de la carcasa.

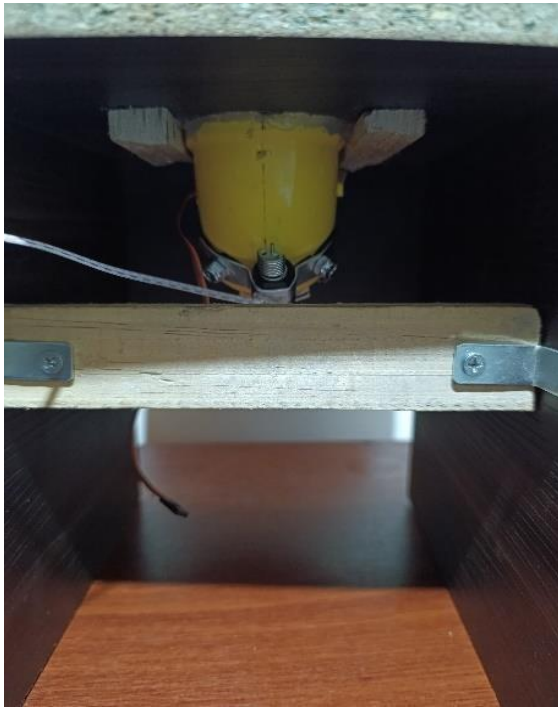
Figura 34*Tapa Posterior*

Nota. Tapa posterior con bisagras [Fotografía]. Elaboración propia

Internamente, la carcasa cuenta con un soporte horizontal, asegurado a las paredes laterales mediante ángulos metálicos de refuerzo y tornillos. Este soporte cumple una doble función: proporciona una base estable y segura para empotrar la caja plástica que contendrá el circuito impreso del sistema electrónico, y además contribuye a la rigidez y estabilidad estructural general de la carcasa, reforzando las tapas laterales y el conjunto.

Figura 35

Soporte Horizontal



Nota. Soporte horizontal para fijar la tarjeta electrónica [Fotografía]. Elaboración propia

Ensamble Electrónico y Programación

Soldadura y Ensamblaje de la PCB. El proceso de ensamblaje se realizó manualmente, usando un cautín de punta fina y soldadura de estaño. Los componentes se posicionaron según la serigrafía de la PCB, después de verificar sus especificaciones. Para la conexión del ESP32, se soldó una regleta header hembra, facilitando su inserción y extracción.

Para facilitar el mantenimiento o posible remplazo de algunos de algunos componentes, se utilizaron conectores hembra en la PCB que se interconectan con los pines macho del servomotor, motovibrador y pulsador respectivamente. La alimentación externa se aseguró mediante una bornera de bloque de dos pines. Finalmente, se montaron las resistencias y el diodo

rectificador (en paralelo al motovibrador) en sus ubicaciones correspondientes, concluyendo con una inspección visual de las soldaduras.

Instalación de la Tarjeta Electrónica. Pensando en su protección, la tarjeta electrónica fue colocada dentro de una caja plástica especial para PCB. Es ideal porque evita el contacto con polvo o la humedad, facilitando el manejo del cableado y garantizando la seguridad del sistema al mantener los componentes eléctricamente aislados. La sujeción interna de la tarjeta se realizó mediante postes metálicos para PCB. Una vez la tarjeta protegida y fijada en su caja, el conjunto se ancló al soporte horizontal interno de la carcasa de MDF.

Compilación y Carga Del Programa al Microcontrolador. La funcionalidad principal del dispensador de alimento depende del código programado que controla el comportamiento del microcontrolador. Una vez desarrollado y depurado el programa en el entorno de desarrollo integrado (IDE) de Arduino, el siguiente paso fue cargarlo en el módulo ESP32-WROOM-32. El proceso de subida del código implicó los siguientes pasos:

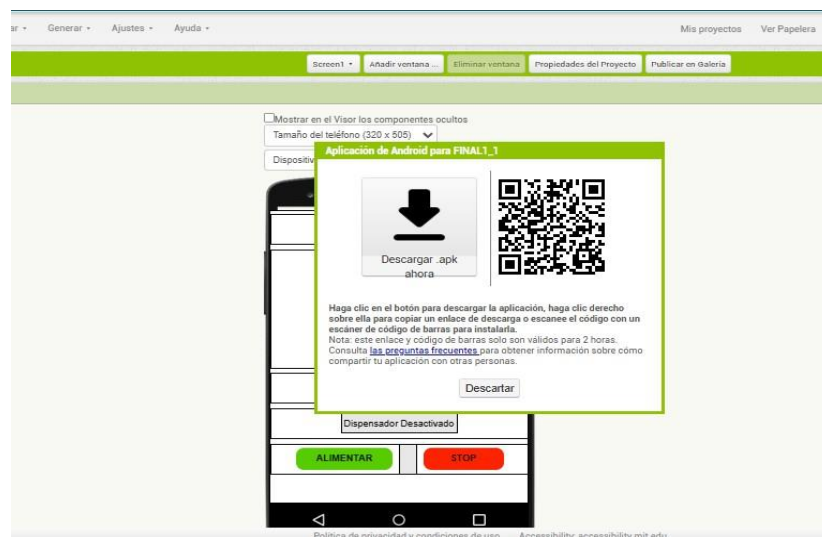
- Conectar el ESP32 al puerto USB del ordenador mediante un cable de datos.
- Seleccionar en el IDE el modelo de placa correspondiente al ESP32-WROOM-32.
- Elegir el puerto COM al que estaba conectado el microcontrolador.
- Compilar el programa para verificar que no existieran errores.
- Subir el programa al módulo ESP32 y esperar a que se completara el proceso de carga.
- Verificar la correcta ejecución del programa mediante mensajes de depuración o pruebas funcionales.

Descarga e Instalación de la App "PetFoodApp". La aplicación móvil "PetFoodApp", desarrollada con MIT App Inventor, es el componente clave para la interacción remota con el dispensador de alimento. Para su uso, es necesario descargar e instalar el archivo de la aplicación en un dispositivo Android. El proceso general de descarga e instalación implica:

Generación del Archivo APK: Desde el entorno de MIT App Inventor, se genera el archivo instalable de la aplicación, que tiene la extensión *.apk*. Este archivo contiene todo el código y los recursos necesarios para que la aplicación funcione en un dispositivo Android.

Figura 36

Generación de Enlace de Descarga



Nota. Desde la App de Mit App inventor se genera el enlace que ser compartido con los voluntarios para las pruebas [imagen] Elaboración propia.

Descarga al Dispositivo Móvil. El archivo .apk se puede descargar directamente al smartphone o Tablet Android. Esto se puede hacer escaneando un código QR generado por App Inventor o descargando el archivo desde un enlace proporcionado.

Habilitación de "Fuentes Desconocidas" (si es necesario). Dado que la aplicación no se descarga desde una tienda oficial como Google Play Store, es posible que el dispositivo Android requiera que se habilite la opción "Instalar aplicaciones de fuentes desconocidas" o "Fuentes desconocidas" en la configuración de seguridad. Este paso es necesario para permitir la instalación de aplicaciones obtenidas fuera de los canales habituales.

Figura 37

Instalación de Fuentes Desconocidas



Nota. Desde la App de Mit App inventor se genera el link que ser compartido con los voluntarios para las pruebas[imagen] Elaboración propia.

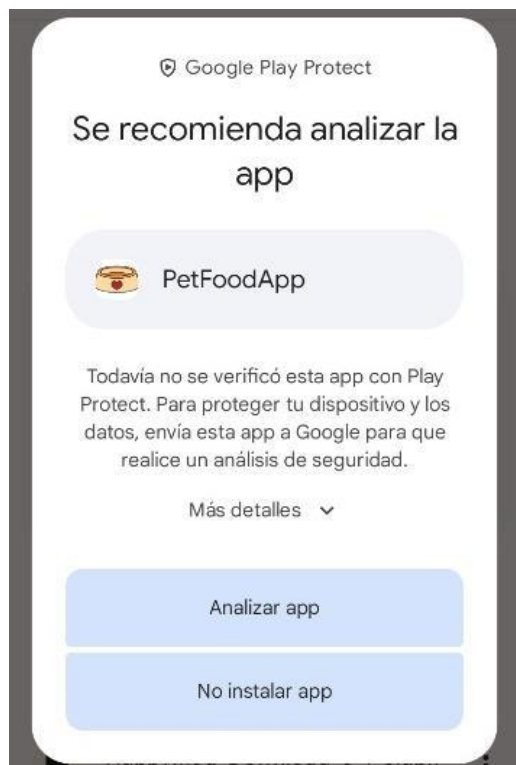
Instalación de la Aplicación

Una vez descargado el archivo y ajustada la configuración de seguridad si fue necesario, se procede a abrir el archivo .apk desde el explorador de archivos del

dispositivo y seguir las instrucciones para completar la instalación de la "PetFoodApp". Una vez instalada, la aplicación estará lista para establecer comunicación con el dispensador a través del broker MQTT, permitiendo al usuario controlar y monitorear el dispositivo.

Figura 38

Análisis de Seguridad de la Aplicación



Nota. análisis de seguridad de Google [imagen] Elaboración propia.

Figura 39*PetFoodApp*

Nota. Interfaz PetFoodApp[imagen] Elaboración propia.

Evaluación y Análisis

Esta fase se enfocó en la verificación del funcionamiento óptimo del prototipo, la validación de sus características técnicas y la evaluación de la experiencia del usuario. Para ello, se realizaron pruebas exhaustivas que incluyeron el control local mediante el botón físico, el control local a través de la aplicación móvil, y el control remoto desde distintas ubicaciones geográficas como Bogotá, Medellín y Almería, Andalucía (España). Estas evaluaciones aseguraron que el dispensador cumpliera con los requisitos de diseño y fuera eficiente en su operación diaria, abarcando tanto el rendimiento de los componentes mecánicos y electrónicos como la facilidad de uso y la eficacia de la interacción a través de la aplicación móvil.

Las pruebas de funcionamiento tuvieron como fin principal verificar la correcta operación del sistema dispensador de comida para perros y gatos. Se examinó su desempeño tanto en el modo de control local a través del botón y la aplicación, como también en el modo remoto, mediante la aplicación móvil diseñada.

El conjunto de pruebas se enfocó en verificar la capacidad efectiva de dispensar el alimento, la calidad de la retroalimentación sobre el estado del servo en diversos escenarios de conectividad y la usabilidad general de la aplicación móvil. Para las distintas pruebas, previamente el dispensador se calibró para generar porciones de alrededor de 60 gramos de concentrado para gatos. Se tomó como referencia las porciones que se deben proporcionar a gatos adultos según información proporcionada por la marca Mirringo en su artículo titulado "¿Cuánto debo alimentar a mi gato cachorro y a mi gato adulto?" en la página oficial (Mirringo, s.f.).

Tanto las pruebas locales como las remotas se hicieron, cada una, durante 5 días consecutivos. Como el proceso abarcó un total de cuatro pruebas distintas, el tiempo total de las

pruebas fue de 20 días consecutivos. Para las pruebas de control remoto, se colaboró con voluntarios en distintas ubicaciones quienes, mediante videollamada, confirmaron las activaciones del prototipo y el comportamiento de la aplicación.

Pruebas 1: Funcionamiento Local

Como parte de la validación funcional del dispensador automático de comida, se realizaron pruebas locales para verificar su correcto funcionamiento en condiciones controladas. Se evaluó el control manual mediante un botón físico y el control remoto a través de una aplicación móvil vía Wi-Fi. En ambos casos se analizaron la respuesta del sistema, la consistencia en la cantidad de alimento dispensado y la estabilidad operativa durante varios días de uso continuo.

Prueba 1.1: Funcionamiento Local Mediante Botón De Inicio

Esta prueba tuvo como objetivo verificar la funcionalidad del control local del dispensador utilizando el botón de inicio integrado en el prototipo. Se evaluó la consistencia de la dispensación a lo largo de un período de cinco días, con activaciones en diferentes momentos del día. Durante toda la prueba, el prototipo se mantuvo constantemente conectado a la energía eléctrica en el lugar de desarrollo (Bogotá, Colombia).

Preparación. Al inicio de cada sesión de prueba, se aseguró que el prototipo estuviera encendido y listo para funcionar.

Activaciones Programadas. Durante cinco días consecutivos, el botón de inicio local se presionó dos veces al día: una por la mañana y otra por la tarde. Ambas activaciones se realizaron aproximadamente a la misma hora cada día para mantener la consistencia de los intervalos.

Registro de Observaciones. Cada día, en las activaciones de la mañana y de la tarde, se observó la activación del servo y la dispensación de comida. La cantidad precisa de alimento dispensado fue medida con una gramera.

Datos de Entrada. Se registraron diez pulsaciones del botón de inicio local, dos veces al día durante cinco días.

Tabla 4

Resultados Prueba 1.1

| Día | Horario | Activación Servo (Sí/No) | Cantidad dispensada (grs) | Anomalías |
|-----|---------|-----------------------------|------------------------------|-------------|
| 1 | mañana | si | 60 | Sin novedad |
| | Tarde | si | 67 | Sin novedad |
| 2 | mañana | si | 66 | Sin novedad |
| | Tarde | si | 65 | Sin novedad |
| 3 | mañana | si | 67 | Sin novedad |
| | Tarde | si | 63 | Sin novedad |
| 4 | mañana | si | 68 | Sin novedad |
| | Tarde | si | 65 | Sin novedad |
| 5 | mañana | si | 66 | Sin novedad |
| | Tarde | si | 67 | Sin novedad |

Nota. Fuente: Elaboración propia

Conclusiones de la Prueba 1.1. El dispensador de alimentos demostró un rendimiento muy consistente al ser activado manualmente. En las activaciones realizadas y registradas, el mecanismo de dispensación operó sin inconvenientes, liberando la comida correctamente. La cantidad de alimento entregado fue muy consistente, con un promedio de 65.40 gramos por porción. La variación fue mínima, de apenas un 8.26% sobre el promedio, lo cual se considera

muy adecuado para un dispositivo de este tipo. Durante la prueba no se presentaron anomalías lo que demuestra que el prototipo es estable y cumple bien su función.

Prueba 1.2: Control mediante Aplicación Local

Esta prueba de cinco días buscó verificar la operación y precisión del dispensador usando una aplicación móvil personalizada. Dicha aplicación, desarrollada con MIT App Inventor, se conectaba al dispensador (que usa un chip ESP32) a través de un broker MQTT por Wi-Fi.

Durante la evaluación, se comprobó la conexión al bróker, cómo interactuaba la aplicación con el dispositivo, su respuesta, y qué tan uniforme era la dispensación del alimento. Todas las pruebas se realizaron con el prototipo constantemente conectado a la corriente en Bogotá, Colombia.

Preparación. se aseguró que el prototipo estuviera encendido y conectado a la red Wi-Fi local. Posteriormente, se verificó que el broker MQTT estuviera operativo y fuera accesible para el dispositivo. Finalmente, se abrió la aplicación móvil en un dispositivo compatible.

Activaciones programadas. Para cada sesión (mañana y tarde, aproximadamente a la misma hora cada día):

- Dentro de la aplicación, se presionó el botón "Conectar".
- Se verificó que el botón "Conectar" cambiara su estado a "Conectado", confirmando la conexión exitosa con el broker MQTT.
- Una vez conectado, se presionó el botón "Alimentar" en la aplicación.

Registro de Observaciones. se observó la activación del servo y la dispensación de comida. Se verificó que el mensaje "Dispensador Activado" aparecía en la aplicación durante aproximadamente un segundo mientras el prototipo dispensaba el alimento, confirmando la

acción, y luego desaparecía. Finalmente, se pesó con una gramera la cantidad de comida dispensada para obtener una medición precisa.

Entorno de Prueba. Prototipo completo en el lugar de desarrollo (Bogotá, Colombia), con conectividad Wi-Fi estable y acceso a internet.

Datos de Entrada. Secuencia de interacción con la aplicación móvil (presionar "Conectar" y luego "Alimentar" dos veces al día durante 5 días).

Tabla 5

Resultados Prueba 1.2

| Día | Horario | Estado conexión App | Activación Servo (Sí/No) | Cantidad dispensada (grs) | Notas /anomalías |
|-----|---------|------------------------|-----------------------------|------------------------------|---------------------|
| 1 | mañana | Conectado | Sí | 65 | Sin novedad |
| | tarde | Conectado | Sí | 61 | Sin novedad |
| 2 | mañana | Conectado | Sí | 59 | Sin novedad |
| | tarde | Conectado | Sí | 64 | Sin novedad |
| 3 | mañana | Conectado | Sí | 63 | Sin novedad |
| | tarde | Conectado | Sí | 64 | Sin novedad |
| 4 | mañana | Conectado | Sí | 64 | Sin novedad |
| | tarde | Conectado | Sí | 65 | Sin novedad |
| 5 | mañana | Conectado | Sí | 59 | Sin novedad |
| | tarde | Conectado | Sí | 58 | Sin novedad |

Nota. Elaboración propia

Conclusiones de la Prueba 1.2. La evaluación del dispensador de comida, controlado localmente a través de la aplicación móvil y la conectividad Wi-Fi/MQTT durante cinco días, confirmó su eficacia y fiabilidad. El prototipo mostró una conexión consistente y exitosa con el broker MQTT en todas las activaciones.

El comando de la aplicación resultó en una activación precisa del servo y una dispensación adecuada de alimento, con una retroalimentación clara en la interfaz. La cantidad dispensada fue notablemente consistente, promediando 62.2 gramos por porción con un margen de error del 6.91%, considerado aceptable. La ausencia de anomalías destaca la estabilidad y la efectiva integración entre el hardware y el software de control local.

Demostración de Funcionamiento del Prototipo en Video

Para complementar la descripción detallada del diseño, implementación y pruebas de este proyecto, hemos preparado un video explicativo que ofrece una perspectiva visual y dinámica del prototipo de dispensador de comida para perros y gatos en funcionamiento. En este video, se podrá observar el ensamble del dispositivo, su operación automática y través de la aplicación móvil, así como la interacción entre sus componentes clave, brindando una comprensión más profunda de la solución desarrollada.

Ver Video: <https://youtu.be/XZ8Bo2QTIK0>

Prueba 2: Funcionamiento Remoto

Ahora se mostrarán las pruebas realizadas para verificar la funcionalidad del control remoto del dispensador mediante la aplicación móvil desde diversas ubicaciones geográficas. Durante la totalidad de estas pruebas de funcionamiento remoto, el prototipo se mantuvo constantemente conectado a la energía eléctrica en el lugar de desarrollo (Bogotá, Colombia). El evaluador compartió previamente la aplicación móvil con los voluntarios para facilitar la ejecución de estas pruebas.

Prueba 2.1: Control Remoto desde Bogotá (Ubicación Remota)

Esta prueba tiene como fin evaluar la funcionalidad del control remoto del dispensador cuando se opera desde una ubicación dentro de la misma ciudad de Bogotá donde se encuentra el prototipo, específicamente desde un punto geográficamente lejano. Se verificará la conectividad con el bróker, la capacidad de respuesta del dispositivo, la fiabilidad de la dispensación y la usabilidad de la aplicación móvil en un entorno de control remoto metropolitano.

Preparación Inicial. Para comenzar la prueba, el prototipo se encendió y se conectó a la red Wi-Fi local en el sitio de desarrollo. Simultáneamente, se verificó que el bróker MQTT estuviera operativo y accesible. Finalmente, la aplicación móvil se abrió en un dispositivo compatible, el cual fue operado por un voluntario desde un punto remoto dentro de Bogotá.

Activaciones programadas: Para cada sesión (mañana y tarde, aproximadamente a la misma hora cada día), el voluntario remoto realizó los siguientes pasos a través de la aplicación:

- Dentro de la aplicación, presionó el botón "Conectar".
- Verificó que el botón "Conectar" cambiara su estado a "Conectado", confirmando la conexión exitosa con el broker MQTT.
- Una vez conectado, presionó el botón "Alimentar" en la aplicación.

Observaciones en Cada Activación (Mañana y Tarde de cada día). Durante las pruebas, se implementó una videollamada constante con el lugar donde se encontraba el prototipo, lo que permitió confirmar visualmente la activación del servo y la dispensación del alimento en tiempo real. Simultáneamente, el voluntario remoto verificó en su dispositivo la aparición y posterior desaparición del mensaje "Dispensador Activado" en la aplicación móvil, confirmando así la recepción del comando. La cantidad de comida dispensada fue medida con precisión en Bogotá

utilizando una gramera, y se realizó una estimación conjunta de la latencia del sistema, calculando el tiempo transcurrido desde que se presionaba el botón "Alimentar" en la aplicación hasta que se observaba la respuesta del dispensador.

Entorno de Prueba. El prototipo completo se mantuvo en el lugar de desarrollo (Bogotá, Colombia), con conectividad Wi-Fi estable y acceso a internet de alta velocidad. La operación remota fue realizada desde una ubicación distinta dentro de Bogotá, también con conectividad a internet estable.

Datos de Entrada. Secuencia de interacción remota con la aplicación móvil (presionar "Conectar" y luego "Alimentar" dos veces al día durante 5 días) ejecutada por el voluntario desde Bogotá.

Tabla 6

Resultados prueba 2.1

| Día | Horario | Estado de conexión App | ¿Activación de Servo? | ¿Mensaje estado dispensador recibido? | Cantidad Dispensada | Anomalías |
|-----|---------|------------------------|-----------------------|---------------------------------------|---------------------|-----------|
| 1 | Mañana | Conectado | Si | Si | 59 | Ninguna |
| | Tarde | Conectado | Si | Si | 63 | Ninguna |
| 2 | Mañana | Conectado | Si | Si | 58 | Ninguna |
| | Tarde | Conectado | Si | Si | 62 | Ninguna |
| 3 | Mañana | Conectado | Si | Si | 62 | Ninguna |
| | Tarde | Conectado | Si | Si | 64 | Ninguna |
| 4 | Mañana | Conectado | Si | Si | 62 | Ninguna |
| | Tarde | Conectado | Si | Si | 60 | Ninguna |
| 5 | Mañana | Conectado | Si | Si | 59 | Ninguna |
| | Tarde | Conectado | Si | Si | 63 | Ninguna |

Nota. Elaboración propia

Resultados y análisis de Prueba 2.1. Esta prueba confirmó la eficacia y fiabilidad del dispensador de alimento, demostrando que el dispositivo puede ser manipulado desde cualquier punto de Bogotá. Se verificó una conexión exitosa y consistente vía MQTT, con el servo activándose de forma fiable y la retroalimentación visual funcionando correctamente desde la aplicación móvil a distancia.

Las cantidades dispensadas fueron consistentes (entre 58 y 63 gramos), presentando un porcentaje de error aproximado de $\pm 4.3\%$. Esto valida la capacidad del sistema para dosificar adecuadamente desde la distancia. En resumen, la prueba demostró que el prototipo puede ser controlado efectiva y establemente de forma remota, confirmando su viabilidad técnica sin anomalías significativas, incluso operando desde ubicaciones diversas dentro de la misma ciudad.

Prueba 2.2: Control Remoto desde Medellín (Ubicación Remota Nacional)

Esta prueba tiene como fin evaluar la funcionalidad del control remoto del dispensador cuando se opera desde una ubicación en otra ciudad de Colombia, específicamente Medellín, Antioquia, donde se encuentra el voluntario, mientras el prototipo permanece en Bogotá. Se verificará la conectividad con el bróker, la capacidad de respuesta del dispositivo, la fiabilidad de la dispensación y la usabilidad de la aplicación móvil en un entorno de control remoto a distancia nacional.

Preparación Inicial. Para comenzar la prueba, el prototipo se encendió y se conectó a la red Wi-Fi local en el sitio de desarrollo en Bogotá. Simultáneamente, se verificó que el bróker MQTT estuviera operativo y accesible. Finalmente, la aplicación móvil se abrió en un dispositivo compatible, el cual fue operado por un voluntario desde Medellín, Antioquia, Colombia.

Activaciones Programadas. En cada sesión de prueba, realizada en la mañana y en la tarde a horas similares cada día, el voluntario remoto interactuó con la aplicación móvil siguiendo una secuencia específica. Primero, presionó el botón "Conectar" dentro de la interfaz y verificó que su estado cambiara a "Conectado", lo cual indicaba que la conexión con el broker MQTT había sido establecida exitosamente. Una vez confirmada la conexión, procedió a presionar el botón "Alimentar" para enviar el comando de activación al dispensador.

Observaciones en Cada Activación (Mañana y Tarde de cada día). Durante cada activación, se mantuvo una videollamada con el lugar del prototipo en Bogotá, lo que permitió confirmar visualmente tanto la activación del servo como la dispensación del alimento. Simultáneamente, el voluntario remoto verificó en su dispositivo la aparición y posterior desaparición del mensaje "Dispensador Activado" en la aplicación, validando así la ejecución del comando. La cantidad de comida dispensada fue medida localmente en Bogotá con una gramera, asegurando precisión en los registros.

Entorno de Prueba: El prototipo completo se mantuvo en el lugar de desarrollo (Bogotá, Colombia), con conectividad Wi-Fi estable y acceso a internet de alta velocidad. La operación remota fue realizada desde Medellín, Antioquia, Colombia, también con conectividad a internet estable.

Tabla 7

Resultados Prueba 2.2

| Día | Horario | Estado de conexión App | ¿Activación de Servo? | ¿Mensaje estado dispensador recibido? | Cantidad Dispensada | Anomalías |
|-----|---------|------------------------|-----------------------|---------------------------------------|---------------------|-----------|
| 1 | Mañana | Conectado | Si | Si | 63 | Ninguna |

| | | | | | | |
|---|--------|-----------|----|----|----|---------|
| | Tarde | Conectado | Si | Si | 64 | Ninguna |
| 2 | Mañana | Conectado | Si | Si | 59 | Ninguna |
| | Tarde | Conectado | Si | Si | 60 | Ninguna |
| 3 | Mañana | Conectado | Si | Si | 62 | Ninguna |
| | Tarde | Conectado | Si | Si | 60 | Ninguna |
| 4 | Mañana | Conectado | Si | Si | 63 | Ninguna |
| | Tarde | Conectado | Si | Si | 59 | Ninguna |
| 5 | Mañana | Conectado | Si | Si | 64 | Ninguna |
| | Tarde | Conectado | Si | Si | 60 | Ninguna |

Nota. Elaboración propia

Conclusiones de la Prueba 2.2. La Prueba 2.2 confirmó la eficacia y fiabilidad del dispensador de alimento bajo control remoto nacional, demostrando que el dispositivo puede ser manipulado desde cualquier punto del país, con Medellín como punto de prueba. Se verificó una conexión exitosa y consistente vía MQTT, con el servo activándose de forma fiable y la retroalimentación visual funcionando correctamente desde la aplicación móvil a distancia.

Las cantidades dispensadas fueron consistentes (entre 59 y 64 gramos). Tomando el promedio de 61.4 gramos y la desviación máxima de 2.6 gramos (64 - 61.4), se obtuvo un porcentaje de error aproximado de $\pm 4.23\%$. Esto valida la capacidad del sistema para dosificar adecuadamente desde la distancia, a pesar de la separación geográfica significativa. En resumen, la prueba demostró que el prototipo puede ser controlado efectiva y establemente de forma remota, confirmando su viabilidad técnica sin anomalías significativas, incluso operando desde ubicaciones diversas a nivel nacional.

Prueba 2.3: Control Remoto desde Andalucía, España (Ubicación Remota Global)

Esta prueba tiene como fin evaluar la funcionalidad del control remoto del dispensador desde cualquier parte del mundo y determinar su funcionalidad y viabilidad a escala global. Para

ello, se utilizó Andalucía, España, como un caso de prueba de ubicación extremadamente distante del prototipo en Bogotá, Colombia. Se verificará la conectividad con el bróker, la capacidad de respuesta del dispositivo, la fiabilidad de la dispensación y la usabilidad de la aplicación móvil en un entorno de control remoto a escala global.

Preparación. Para comenzar la prueba, el prototipo se encendió y se conectó a la red Wi-Fi local en el sitio de desarrollo en Bogotá, Colombia. Simultáneamente, se verificó que el bróker MQTT estuviera operativo y accesible. Finalmente, la aplicación móvil se abrió en un dispositivo compatible, el cual fue operado por un voluntario desde Valencia, España.

Activaciones Programadas. Para cada sesión de prueba, realizada en la mañana y en la tarde (aproximadamente a la misma hora cada día, ajustando por la diferencia horaria), el voluntario remoto ejecutó una secuencia de acciones desde la aplicación móvil. Primero, presionó el botón "Conectar" dentro de la interfaz, verificando que este cambiara su estado a "Conectado", lo cual indicaba una conexión exitosa con el broker MQTT. Una vez establecida la conexión, presionó el botón "Alimentar" para enviar el comando de activación al dispensador.

Observaciones en Cada Activación. En cada activación, se mantuvo una videollamada constante con el lugar del prototipo en Bogotá para confirmar visualmente la activación del servo y la dispensación del alimento. Al mismo tiempo, el voluntario remoto verificó en su dispositivo la aparición y posterior desaparición del mensaje "Dispensador Activado" en la aplicación, lo que confirmaba la ejecución del comando. La cantidad de comida dispensada fue medida en Bogotá utilizando una gramera para asegurar precisión. Además, se realizó una estimación conjunta del tiempo de respuesta (latencia) desde que el voluntario presionaba el botón "Alimentar" en Valencia hasta que se observaba la activación efectiva del sistema en Bogotá.

Entorno de Prueba: El prototipo completo se mantuvo en el lugar de desarrollo (Bogotá, Colombia), con conectividad Wi-Fi estable y acceso a internet de alta velocidad. La operación remota fue realizada desde Valencia, España, también con conectividad a internet estable.

Tabla 8

Resultados Prueba 2.3

| Día | Horario | Estado de conexión App | ¿Activación de Servo? | ¿Mensaje estado dispensador recibido? | Cantidad Dispensada | Anomalías |
|-----|---------|------------------------|-----------------------|---------------------------------------|---------------------|-----------|
| 1 | Mañana | Conectado | Si | Si | 59 | Ninguna |
| | Tarde | Conectado | Si | Si | 63 | Ninguna |
| 2 | Mañana | Conectado | Si | Si | 62 | Ninguna |
| | Tarde | Conectado | Si | Si | 59 | Ninguna |
| 3 | Mañana | Conectado | Si | Si | 64 | Ninguna |
| | Tarde | Conectado | Si | Si | 62 | Ninguna |
| 4 | Mañana | Conectado | Si | Si | 62 | Ninguna |
| | Tarde | Conectado | Si | Si | 59 | Ninguna |
| 5 | Mañana | Conectado | Si | Si | 60 | Ninguna |
| | Tarde | Conectado | Si | Si | 60 | Ninguna |

Fuente: Elaboración propia

Conclusiones de la Prueba 2.3. La Prueba 2.3 confirmó la eficacia y fiabilidad del dispensador de alimento bajo control remoto global, demostrando que el dispositivo puede ser manipulado desde cualquier parte del mundo, con Valencia, España, como punto de prueba. Verificamos una conexión exitosa y consistente vía MQTT; el servo se activó de forma fiable y la retroalimentación visual funcionó correctamente desde la aplicación móvil a distancia.

Las cantidades dispensadas fueron consistentes (entre 59 y 64 gramos). Tomando el promedio de 61.0 gramos y la desviación máxima de 3.0 gramos (64 - 61.0), el porcentaje de

error fue aproximadamente de $\pm 4.92\%$. Esta variabilidad es aceptable y valida la capacidad del sistema para dosificar adecuadamente a pesar de la significativa separación geográfica intercontinental. En resumen, la prueba demostró que el prototipo puede controlarse efectiva y establemente de forma remota, confirmando su viabilidad técnica sin anomalías significativas, incluso operando desde ubicaciones diversas a nivel global.

Resultados

Las pruebas de funcionamiento del prototipo del dispensador remoto de comida para mascotas, realizadas tanto en control local como a distancia, permitieron verificar su desempeño y comportamiento. El análisis de los datos confirma un funcionamiento consistente y eficiente.

La cantidad de alimento dispensado se mantuvo cercana a los 60 gramos programados, con variaciones modestas, generalmente inferiores al 8.26 % respecto al promedio. Esto demuestra que el sistema de dosificación, compuesto por el servomotor y el motor vibratorio, garantiza una entrega uniforme y sin interrupciones.

Durante las 40 activaciones registradas a lo largo de 20 días, el prototipo mostró una operación estable, sin fallos o anomalías relevantes. En cada solicitud de dispensación, el servomotor se activó correctamente, el alimento fue entregado y la confirmación en la aplicación apareció de forma adecuada.

La capacidad de control remoto se validó con pruebas desde diferentes ubicaciones, incluyendo Bogotá, Medellín y Andalucía (España). La conexión a Internet entre la aplicación y el dispensador fue constante y confiable, permitiendo el envío de órdenes y la recepción de confirmaciones sin retrasos significativos. La retroalimentación de la activación en la aplicación funcionó correctamente en todo momento.

Conclusiones

El proyecto ha culminado con el desarrollo exitoso de un dispensador remoto de comida para mascotas, cumpliendo plenamente con su propósito principal: ofrecer una solución práctica y confiable para alimentar animales a distancia. Se ha logrado diseñar y construir un prototipo funcional que dispensa alimento de manera consistente y puede ser controlado remotamente, con una interfaz de usuario intuitiva. Las pruebas validaron su capacidad de operar a distancia, incluso desde ubicaciones geográficas diversas en cualquier parte del mundo, lo que confirma su efectividad para brindar tranquilidad a los dueños de mascotas.

Para el diseño, creación y funcionamiento de este dispensador, fue crucial considerar parámetros específicos que garantizaran su rendimiento óptimo. En la parte electrónica, el circuito creado con el ESP32 como procesador central, gestiona las funciones con eficiencia. Esta integración se verificó en pruebas, confirmando la fiabilidad en la entrega de porciones y una respuesta sólida a comandos remotos. Esta solución no solo aporta comodidad y bienestar animal, sino que también es una aplicación concreta y viable del Internet de las Cosas (IoT) en el entorno doméstico.

Sin embargo, el prototipo actual tiene limitaciones. Depende de una conexión Wi-Fi y energía fija, lo que reduce su autonomía. El diseño se enfocó solo en la dispensación básica, sin sensores de nivel de alimento, cámara ni programación avanzada. Esto define el alcance actual del proyecto.

Referencias Bibliográficas

- Arduino. (s.f.). millis (). Arduino. <https://docs.arduino.cc/language-reference/en/functions/time/millis/>
- Boylestad, R. L., & Nashelsky, L. (2013). *Dispositivos electrónicos y teoría de circuitos*. Pearson Educación.
- Brian. (2021, 30 junio). Choosing a Development Board for Your Project | Circuit Crush. Circuit Crush. <https://www.circuitcrush.com/choosing-development-board/>
- Cerón, R. (2024, 30 abril). ¿Qué es MQTT? El protocolo más utilizado para IoT. Pandora FMS. <https://pandorafms.com/es/it-topics/que-es-mqtt/>
- DANE. (2022, 1 julio). Encuesta Multipropósito 2021 [Comunicado de prensa]. https://www.dane.gov.co/files/investigaciones/multi/Comunicado_EM_2021.pdf
- Diez, B. (2024, 30 agosto). ¿Qué es un diodo LED y cómo funciona? La guía definitiva. B·LED - Blog. <https://www.barcelonaed.com/blog/informacion-led/que-es-un-diodo-led-y-como-funciona-la-guia-definitiva/?srsltid=AfmBOorC-PZms8YaIcUIIiYVV7hlfNHdJeAfnStq-Ktmhpz7DkxbjbV9>
- Erikaluna. (2024, 20 agosto). Qué es un IDE - Concepto, características y ejemplos. <https://platzi.com/blog/que-es-ide-editor-de-texto/>
- Fernández, J. M. E. (2021, 29 noviembre). IoT: Qué es, para qué sirve y cómo funciona. OpenWebinars.net. <https://openwebinars.net/blog/iot-que-es-para-que-sirve-y-como-funciona/>
- Hernández-Sampieri, R., Fernández-Collado, C., & Baptista, P. (2014). *Metodología de la investigación* (6ª ed.). McGraw-Hill Education.

IndustrySurfer, & IndustrySurfer. (2023, 11 marzo). Las 10 mejores placas de desarrollo

integradas. Industry Surfer. [https://industrysurfer.com/blog-](https://industrysurfer.com/blog-industrial/ingenieria/ingenieria-electrica-ingenieria/las-10-mejores-placas-de-desarrollo-integradas/)

[industrial/ingenieria/ingenieria-electrica-ingenieria/las-10-mejores-placas-de-desarrollo-integradas/](https://industrysurfer.com/blog-industrial/ingenieria/ingenieria-electrica-ingenieria/las-10-mejores-placas-de-desarrollo-integradas/)

Ineed Electronics. (2021, 16 de mayo). Motores de vibración de masa giratoria excéntrica y

actuadores de resonancia lineal. <https://www.ineed-motor.com/news/eccentric-rotating-mass-vibration-motors-and-l-46825132.html>

Llamas, L. (2019, 17 abril). ¿Qué es MQTT? Su importancia como protocolo IoT. Luis Llamas.

<https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>

María. (2024, 8 noviembre). Placas de Desarrollo: Empezar en el Mundo de la Electrónica y la

Programación. Phipsil. <https://phipsil.com/2024/11/08/placas-de-desarrollo-empezar-en-el-mundo-de-la-electronica-y-la-programacion/>

marvinroger. (2023). async-mqtt-client (Versión) [Repositorio de código]. GitHub. URL.

<https://github.com/marvinroger/async-mqtt-client>

Mirringo. (s.f.). ¿Cuánto debo alimentar a mi gato cachorro y a mi gato adulto? [Artículo en

página web]. Recuperado de [https://www.mirringo.com.co/actualidad-gatuna/cuanto-](https://www.mirringo.com.co/actualidad-gatuna/cuanto-debo-alimentar-a-mi-gato-cachorro-y-a-mi-gato-adulto#:~:text=GATOS%20ADULTOS,-A%20partir%20del&text=Llena%20su%20comedero%20dos%20veces,kilo%20de%20su%20peso%20corporal)

[debo-alimentar-a-mi-gato-cachorro-y-a-mi-gato-adulto#:~:text=GATOS%20ADULTOS,-A%20partir%20del&text=Llena%20su%20comedero%20dos%20veces,kilo%20de%20su%20peso%20corporal](https://www.mirringo.com.co/actualidad-gatuna/cuanto-debo-alimentar-a-mi-gato-cachorro-y-a-mi-gato-adulto#:~:text=GATOS%20ADULTOS,-A%20partir%20del&text=Llena%20su%20comedero%20dos%20veces,kilo%20de%20su%20peso%20corporal)

Moraguz, E. R. (2023, 28 febrero). Qué es el CAD (diseño asistido por ordenador):

¿cómo funciona y para qué sirve? | LovTechnology. LovTechnology.

<https://lovtechnology.com/que-es-el-cad-diseno-asistido-por-ordenador-como-funciona-y-para-que-sirve/>

madhepha. (2023). ESP32Servo (Versión) [Repositorio de código]. GitHub.

<https://github.com/madhephaestus/ESP32Servo>

Noguez, E. (2020, 31 octubre). 4 razones para que no dejes el alimento de tu perro todo el día en su plato - NUPEC. NUPEC. <https://nupec.com/blog-nupec/4-razones-para-que-no-dejes-el-alimento-de-tu-perro-todo-el-dia-en-su-plato/>

Ricardo. (2022, 25 enero). ¿Qué es un servomotor? ¿Cómo funciona? • Electronic Board.

Electronic Board. <https://www.electronicboard.es/que-es-un-servomotor-como-funciona/>

Serna C., L., Rodríguez de S., A., & Albán A., F. (2003). Ácido Poliláctico (PLA): Propiedades y Aplicaciones. *Ingeniería Y Competitividad*, 5(1), 16–26.

<https://doi.org/10.25100/iyc.v5i1.2301>

Ullis. (s. f.). Ullis Roboter Seite/AI2 MQTT. <https://ullisroboterseite.de/android-AI2-MQTT-en.html>