



UNIVERSIDAD TECNOLÓGICA ISRAEL
ESCUELA DE POSGRADOS "ESPOG"

MAESTRÍA EN
ELECTRÓNICA Y AUTOMATIZACIÓN
Resolución: RPC-SO-09-No.265-2021

PROYECTO DE TITULACIÓN EN OPCIÓN AL GRADO DE MAGÍSTER

Título del proyecto:
Dispensador automatizado de alimento para tilapias con tecnología IoT
Línea de Investigación:
Ciencias de la Ingeniería aplicadas a la producción, sociedad y desarrollo sustentable.
Campo amplio de conocimiento:
Ingeniería, Industria y construcción.
Autor/a:
Johnny Paúl Novillo Vicuña
Tutor/a:
René Ernesto Cortijo Leyva

Quito – Ecuador

2023

APROBACIÓN DEL TUTOR



Yo, **Rene Ernesto Cortijo Leyva** con C.I: **1719010108** en mi calidad de Tutor del proyecto de investigación titulado: **Dispensador automatizado de alimento para tilapias con tecnología IoT,**

Elaborado por: **Johnny Paúl Novillo Vicuña**, de C.I: **0702947409**, estudiante de la Maestría: **ELECTRÓNICA Y AUTOMATIZACIÓN**, resolución: **RPC-SO-09-No.265-2021**, de la **UNIVERSIDAD TECNOLÓGICA ISRAEL (UISRAEL)**, como parte de los requisitos sustanciales con fines de obtener el Título de Magister, me permito declarar que luego de haber orientado, analizado y revisado el trabajo de titulación, lo apruebo en todas sus partes.

Quito D.M., 1 de septiembre del 2023

RENE
ERNEST
O
CORTIJO
LEYVA

Firmado digitalmente por RENE ERNESTO CORTIJO LEYVA
Fecha: 2023.09.01 21:49:01 -05'00'

Firma

DECLARACIÓN DE AUTORIZACIÓN POR PARTE DEL ESTUDIANTE



Yo, Johnny Paúl Novillo Vicuña con C.I: 0702947409, autor/a del proyecto de titulación denominado: **Dispensador automatizado de alimento para tilapias con tecnología IoT**. Previo a la obtención del título de Magister en Electrónica y Automatización.

1. Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar el respectivo trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Manifiesto mi voluntad de ceder a la Universidad Tecnológica Israel los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor@ del trabajo de titulación, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital como parte del acervo bibliográfico de la Universidad Tecnológica Israel.
3. Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de prosperidad intelectual vigentes.

Quito D.M., 31 de agosto de 2023

JOHNNY
PAUL
NOVILLO
VICUNA

Firmado digitalmente
por JOHNNY PAUL
NOVILLO VICUNA
DN: cn=JOHNNY PAUL,
NOVILLO VICUNA c=EC,
o=SECURITY DATA S.A., 1
o=ENTIDAD DE
CERTIFICACION DE
INFORMACION
Motivo: JohnnyNovillo
Ubicación:
Fecha: 2023.08.01
19:54:05.00

Firma

Tabla de contenidos

APROBACIÓN DEL TUTOR	2
DECLARACIÓN DE AUTORIZACIÓN POR PARTE DEL ESTUDIANTE	3
INFORMACIÓN GENERAL	1
Contextualización del tema	1
Problema de investigación	2
Objetivo general	2
Objetivos específicos	2
CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO	5
1.1. Contextualización general del estado del arte	5
1.2. Proceso investigativo metodológico	10
CAPÍTULO II: PROPUESTA	13
2.1 Fundamentos teóricos aplicados	13
Introducción	13
Arquitectura IoT de cuatro niveles	13
Nivel de percepción	14
Nivel de conectividad	18
Nivel de servicio	19
Nivel de aplicación	20
2.2 Descripción de la propuesta	21
a. Estructura general	21
b. Explicación del aporte	25
c. Estrategias y/o técnicas	33
2.3 Validación de la propuesta	34
2.4 Matriz de articulación de la propuesta	38
2.5 Análisis de resultados. Presentación y discusión.	41
CONCLUSIONES	52
RECOMENDACIONES	53
BIBLIOGRAFÍA	54
ANEXOS	56

Índice de tablas

Tabla 1 Descripción de perfil de validadores	35
Tabla 2 Evaluación realizada por los especialistas	36
Tabla 3 Matriz de articulación	39
Tabla 4 Resultados y comparación de temperatura	45
Tabla 5 Resultados y comparación de peso	48
Tabla 6 Resultados de las pruebas de velocidad de dispensación	49

Índice de figuras

Figura 1 Arquitectura IoT de alto nivel	14
Figura 2 Placa de desarrollo NodeMCU-32	15
Figura 3 Módulo HX711 para manejo de celda de carga	15
Figura 4 Celda de carga de 20 Kg utilizada en la implementación del prototipo	16
Figura 5 Sensor de temperatura y humedad DHT11	17
Figura 6 Driver L289N utilizado para manejo de un motor reductor	17
Figura 7 Motor reductor DC de 12 V - 3500/100 rpm	18
Figura 8 Fuente conmutada 12 V / 5A	19
Figura 9 Estructura IoT del prototipo	22
Figura 10 Arquitectura del dispensador automatizado para tilapias	23
Figura 11 Diagrama de flujo para el dispensador automatizado para tilapias	24
Figura 12 Modelo 3D del dispensador automatizado para tilapias	25
Figura 13 Modelo 3D del tornillo sin fin	26
Figura 14 Diagrama de conexión del prototipo	27
Figura 15 Conexión física de los componentes electrónicos	28
Figura 16 Declaración de variables	29
Figura 17 Programación de MQTT	30
Figura 18 Publicación de los topics MQTT	31
Figura 19 Flujo en node-red para la comunicación MQTT	32
Figura 20 Configuración del nodo "mqtt in", primera parte:	32
Figura 21 Configuración del nodo "mqtt in", segunda parte:	33
Figura 22 Configuración del nodo MongoDB	33
Figura 25 Interfaz de la aplicación de escritorio	35
Figura 26 Comprobación de comunicación del dispositivo de borde con HiveMQ.	43
Figura 27 Visualización de los datos en el dashboard de NODE-RED.	43
Figura 28 Visualización de los datos en la aplicación de .NET MAUI	44
Figura 29 Prueba 1 de Temperatura proporcionada por el sistema de aire acondicionado y el sensor DTH11	45
Figura 30 Prueba 2 de Temperatura proporcionada por el sistema de aire acondicionado y el sensor DTH11	45
Figura 31 Prueba 1 balanza digital	47
Figura 32 Prueba 1 sensor HX711	47
Figura 33 Prueba 2 balanza digital	48
Figura 34 Prueba 2 Sensor de peso HX711	48

INFORMACIÓN GENERAL

Contextualización del tema

El Internet de las cosas (IoT, por sus siglas en inglés) es un concepto tecnológico que se refiere a la interconexión de objetos cotidianos con acceso a Internet y la capacidad de recopilar y compartir datos sin intervención humana directa. Su aplicación a la piscicultura se refiere a la implementación de tecnologías conectadas para mejorar la eficiencia, el monitoreo y la gestión de las operaciones en granjas acuícolas. El uso del IoT en la piscicultura ofrece una serie de beneficios significativos que pueden ayudar a los acuicultores a optimizar sus prácticas, garantizar la salud y el bienestar de los peces, y mejorar la producción general. Algunas aplicaciones del IoT en la piscicultura incluyen: monitoreo remoto de la calidad de agua (Delgado Tapia & Valencia Astudillo, 2021), optimización del proceso de piscicultura y toma de decisiones (Rivera Betancur, 2020), sistema de monitoreo IoT en la producción de alevinos (Piamba et al., 2021), entre otras.

El lugar escogido para la realización del presente trabajo, corresponde a una propiedad ubicada en el sitio Río Chico, perteneciente a la Parroquia Bella María, Cantón Santa Rosa de la provincia de El Oro, en donde su propietario cuenta con unos estanques para la crianza de la tilapia roja (*Oreochromis mossambicus*), también conocida como mojarra roja, pero que taxonómicamente no posee un único nombre científico, debido a que es un híbrido del cruce de tres de especies africanas y una israelí.

La propiedad denominada “Quinta Novillo’s”, posee una casa de campo de dos plantas, que brinda todas las comodidades necesarias para el descanso de sus ocupantes, ya que cuenta con servicio de energía eléctrica, agua potable, internet y servicio de televisión por cable. Además cuenta con cancha deportiva, piscina, jacuzzi, y área para eventos sociales.

En la actualidad Quinta Novillo’s, provee los espacios y recursos necesarios para la relajación y el deleite de los miembros familiares, así como también de sus invitados, para lo cual se cuenta con un trabajador que se encarga de la limpieza de las instalaciones, mantenimiento de áreas verdes, cuidado de la piscina y jacuzzi, así como también para el cuidado de los estanques en donde se siembra la tilapia roja. Eventualmente se contrata personal para preparación de comidas y asistencia para realización de eventos.

Todas las personas que han conocido Quinta Novillo’s, y han podido hacer uso de sus espacios e infraestructura, han emitido comentarios de elogio hacia el lugar, siendo uno de los aspectos que les ha llamado mucho la atención, la crianza de las tilapias rojas. Sea para los

amantes de la naturaleza, que se deleitan solamente con observar los peces en su ambiente casi natural, o para aquellos que también les gusta saborear la tilapia, aquí encontrarán lo que buscan, en un ambiente tropical.

Problema de investigación

Actualmente en Quinta Novillo's, no se puede controlar si el trabajador existente, alimenta adecuadamente en cantidad y horarios, a las tilapias que se encuentran en los estanques de la propiedad, lo que afecta directamente en el desarrollo de los cíclidos, y en sus costos de producción.

En este contexto, el problema de la crianza de tilapias en la propiedad, es la carencia de una automatización para su alimentación, en la cantidad y horarios adecuados.

El proceso de alimentación manual para las tilapias, involucra una organización estricta de los tiempos del trabajador, que mucha de las veces no se cumple, y se hace difícil controlar.

Esta situación provoca que las tilapias no se desarrollen adecuadamente y se incrementen los costos de producción para su crianza.

Por lo antes expuesto, se hace necesario la automatización del proceso de alimentación de las tilapias, que permitirá que los cíclidos alcancen su tamaño y peso óptimo en el tiempo justo, con la correspondiente disminución de los costos de producción, y permitiendo de esta manera continuar con la actividad de crianza de tilapias en la Quinta Novillo's.

Objetivo general

Desarrollar un dispensador automatizado de alimento para tilapias con tecnología IoT

Objetivos específicos

Determinar los materiales, sensores, actuadores y controlador para el desarrollo del dispensador de alimentos.

Diseñar el sistema electrónico de control automático del dispensador de alimento basado en ESP32.

Desarrollar una aplicación para la visualización de información de sensores y control del dispensador en tiempo real a través de un dashboard con tecnología IoT.

Validar el cumplimiento de la funcionalidad del prototipo mediante pruebas experimentales.

Vinculación con la sociedad y beneficiarios directos:

La implementación de un dispensador automatizado para alimentación de tilapias, está orientado hacia un ahorro sustancial del tiempo que dedica una persona, para la alimentación de tilapias en estanques, así como también el asegurar que esta alimentación sea proporcionada en la cantidad y horarios adecuados, logrando de esta manera una reducción en los costos de producción y un óptimo desarrollo de los peces.

Llevar un control de la cantidad exacta de alimento que se dispensa a una siembra de tilapias, así como también los horarios en los que se realiza su alimentación, permite validar las recomendaciones que los fabricantes de concentrados proporcionan respecto a la crianza de los cíclidos, y a su vez permitiría obtener resultados específicos de la crianza, ante situaciones particulares de alimentación y entornos climáticos, que servirían como experiencias previas para otros productores.

En la actualidad los productores de tilapias basan la alimentación de las mismas, sobre las recomendaciones que proporcionan los fabricantes de alimentos balanceados, acorde a las etapas de desarrollo de los peces. Aunque productores más experimentados han logrado establecer metodologías que les han dado mejores resultados, acorde a sus experiencias, como el caso de “Estación Piscícola Acuasol”, ubicada en la Parroquia Puerto Jelí, Cantón Santa Rosa de la Provincia de El Oro, que implementa una metodología que la han denominado “Ley de 5 minutos” (Barco, M., comunicación personal, 28 de septiembre de 2022) que consiste en proporcionar el alimento que puedan consumir las tilapias durante el tiempo máximo de 5 minutos, puesto que luego de este tiempo los cíclidos se encuentran saciados, y todo el alimento que se dispensara luego de este tiempo, ya no lo comen y flotaría sobre el estanque.

En nuestro país, la producción de tilapia roja ocupa el segundo lugar en participación económica en la acuicultura, después del cultivo de camarón, y se exporta principalmente hacia los Estados Unidos de Norteamérica. Estas exportaciones tuvieron su punto máximo en el año 2007, y han venido decreciendo paulatinamente hasta el año 2021 (*Estadísticas - Cámara Nacional de Acuicultura, 2022*), sin embargo, no ha dejado de ser una importante fuente de divisas para el país.

.

La producción de tilapia en el Ecuador va desde actividades para autoempleo, hasta procesos industriales de producción, liderados por empresas como: Aquamar, Indumar,

Marfrisco, El Rosado, El Garzal, Modercorp y Empagran (Luzón et al., 2021) . Produciendo de esta forma una alternativa para una economía doméstica, o también una actividad generadora de muchas fuentes de empleo directo e indirecto a través de las micro, medianas y grandes empresas dedicadas a la producción y comercialización de la tilapia.

Dada la importancia y potencial que tiene la producción de tilapias en nuestro país, la implementación exitosa de un prototipo para la alimentación automática de tilapias en la Quinta Novillo's, inicialmente permitiría reproducir éste tipo de tecnología en zonas aledañas, para posteriormente alcanzar una comercialización nacional. Lo que conlleva a una transferencia de tecnología y la posibilidad de realizar investigaciones para elevar el nivel productivo en la crianza de tilapias.

Como parte del proceso para divulgación científica del presente proyecto, se pretende participar en eventos de ferias tecnológicas o congresos que organizan las empresas privadas y/o universidades, con la finalidad de dar a conocer las bondades y la tecnología inmersa en el dispensador automatizado para alimento de tilapias desarrollado.

CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO

1.1. Contextualización general del estado del arte

La aplicación de la tecnología al campo agropecuario, es una tendencia actual que permite resolver muchos problemas, con los que tradicionalmente han venido lidiando los productores agropecuarios. Para contextualizar estos problemas al campo de la piscicultura, podemos mencionar los siguientes: falta de mediciones en tiempo real de las las condiciones físicas y químicas del agua, dependencia exclusiva de la mano del hombre para la alimentación de los peces en criaderos, falta de control para una adecuada dosificación del alimento, falta de sistemas de alerta temprana para evitar la mortalidad, etc.

El presente trabajo se centra en la implementación de un dispensador automático de alimento para tilapias, con la finalidad de disminuir considerablemente el tiempo de trabajo del personal encargado de la alimentación de los peces, así como también tener un control preciso de la dosificación del alimento, en correspondencia con el nivel de desarrollo de los cíclidos. Aunque el prototipo a desarrollarse puede ser utilizado para cualquier especie de peces, se ha escogido la tilapia y específicamente la tilapia roja, para la configuración del prototipo, por ser el pez que mayormente se produce para fines comerciales en nuestro país.

Para ubicar adecuadamente la documentación relevante en el proyecto de un dispensador automático de alimento para tilapias, se deben considerar diversos criterios. En cuanto a los tipos de documentos, se priorizarán investigaciones científicas, informes técnicos y estudios académicos que aborden la nutrición, el comportamiento y la cría de tilapias. Es esencial dar preferencia a documentos recientes, publicados en los últimos 5 años, para asegurar que la información esté actualizada y refleje los avances tecnológicos y científicos más recientes. Los objetivos de estudio se centrarán en aspectos nutricionales, bienestar animal y eficiencia de alimentación. Se buscarán líneas de investigación que exploren tanto la automatización como la sostenibilidad en la alimentación de peces.

Los resultados de esta investigación se utilizan directamente en el entorno del proyecto. La información recopilada será fundamental para diseñar un dispensador que cumpla con las necesidades específicas de las tilapias en términos de alimentación y comportamiento. Además, los hallazgos orientarán la programación y control del dispensador, optimizando la alimentación en función de las condiciones evolutivas de los cíclidos. La transferencia de conocimiento se extenderá a la mejora del bienestar de las tilapias, a la eficiencia de la producción y la adopción de prácticas más sostenibles en la acuicultura. En última instancia, los

resultados de esta investigación contribuirán a la creación de un sistema de alimentación automatizado que aumente la eficiencia de la cría de tilapias.

La construcción de un dispensador automático de alimento para tilapias con tecnología IoT, involucra una serie de conceptos interrelacionados que convergen en un sistema inteligente de alimentación para acuicultura. La base de esta tecnología radica en la interconexión de dispositivos cotidianos a través de internet, permitiéndoles comunicarse y compartir datos. Los sistemas de monitoreo y control se erigen como componentes esenciales, permitiendo supervisar las condiciones ambientales y la disponibilidad de alimento en tiempo real. Para lograr esto, se utilizan sensores ambientales que miden factores como la temperatura, el nivel y la calidad del agua. Estos datos son procesados y transmitidos a través de redes de comunicación como Wi-Fi o Bluetooth.

La eficacia del sistema radica en la capacidad de los actuadores para ejecutar acciones con base en las señales de control. En este caso, los actuadores son responsables de la liberación automatizada del alimento en el entorno acuático. La automatización es clave para garantizar que las tilapias reciban una alimentación consistente y en momentos adecuados. A su vez, se busca lograr eficiencia energética en el sistema, optimizando el consumo eléctrico y promoviendo un funcionamiento sostenible.

Dentro de los referentes en el campo de aplicación del presente proyecto, figuran investigadores como Kevin Ashton, acreditado por haber acuñado el término "Internet de las cosas", siendo un pionero en el campo de IoT, que ha contribuido significativamente a su desarrollo. También es importante citar a Daniel Benetti, profesor de acuicultura en la Universidad de Miami, que ha investigado ampliamente la aplicación de tecnologías IoT para mejorar la gestión y eficiencia en la acuicultura.

A nivel de plataformas tecnológicas, tecnologías como ESP8266, ESP32, Arduino y Raspberry Pi se destacan como pilares para el desarrollo de dispositivos IoT en la acuicultura. En conjunto, estos conceptos y autores delinean un panorama integral para la construcción exitosa de un dispensador automático de alimento para tilapias con tecnología IoT, impulsando la eficiencia y la sostenibilidad en la producción acuícola.

En el trabajo de Aguayo (2018), los investigadores abordaron el desarrollo de un dispositivo automático para suministrar alimento para tilapias, en el área de piscicultura ubicada en la finca experimental La María de la UTEQ. Específicamente se trata del diseño y construcción de un dispensador de balanceado para peces. Este dispositivo tiene la finalidad de contribuir a la alimentación de los peces en un estanque, reemplazando el manual de trabajo

que previamente realizó una persona encargada de esta tarea. Para lograr este objetivo, se plantearon dos propuestas de diseño de dispensadores automáticos, ambas simuladas a través del software SOLIDWORKS. Estas propuestas fueron evaluadas mediante un método de criterio ponderado, permitiendo así la selección de la opción más adecuada (Alternativa "C"), la misma que fue concebida de manera que se alinea con las necesidades del productor, destacando por su diseño simple y su facilidad de uso. Esta alternativa consta de: cuerpo de la estructura, motor, tornillo sin fin (vertical), taza centrífuga, tolva, cono, tapa, soporte, poleas, temporizador y bandas.

En el trabajo de grado de Rivera (2020), se propone un modelo de Internet de las Cosas (IoT) que permite dar soporte al proceso de piscicultura en el Centro de Desarrollo Agroalimentario El Limonal de la Universidad Santo Tomás. El objetivo es monitorear a distancia y en tiempo real diversas variables del proceso, mejorando así los índices de reproducción artificial de los peces. El sistema permite rastrear algunos peces desde su crianza hasta su venta. El proyecto se dividió en tres etapas: un análisis inicial de granjas inteligentes, la creación de un framework IoT para sensar y transmitir variables, y el diseño e implementación de la arquitectura IoT. La especie monitoreada fue la Tilapia; se caracterizó su cultivo y se analizó los costos de producción. Las variables críticas del agua (temperatura, oxígeno disuelto, pH, turbidez y nitrato) se midieron con el dispositivo ESP8266 y sensores, transmitiendo datos por WiFi. Los datos se analizaron con WEKA, un software de código abierto bajo GNU General Public License, utilizando el algoritmo Naive Bayes para predecir la mortalidad de los peces. Este proyecto hizo una clasificación precisa de más del 95% de las instancias, demostrando una predicción efectiva basada en las variables capturadas por el sistema IoT.

En González et al. (2021), el objetivo de la investigación fue establecer un sistema de monitoreo constante de los parámetros del agua en los estanques para la cría de peces, con alertas en caso de detectar valores anómalos que puedan perjudicar el cultivo, gracias a la utilización de tecnologías de IoT. El sistema se compone de un dispositivo Arduino que captura la temperatura y el pH del agua a través de sensores, y transmite los datos a una base de datos en línea mediante una aplicación web. La plataforma web permite visualizar y ajustar los parámetros. En caso de valores anómalos, se notificará al administrador a través de SMS y llamada telefónica, sin mensaje de voz en la llamada y con detalles precisos en el SMS. Esto permite al administrador tomar decisiones con respecto al estado del agua y la cría de peces.

El sistema propuesto en Rosaline (2019) utiliza sensores de temperatura, nivel de agua, salinidad, oxígeno disuelto y amoníaco para monitorear un sistema de cultivo en un estanque.

Estos sensores permanecen en el estanque y envían datos a la nube para su almacenamiento. Si alguno de los sensores detecta valores fuera del rango establecido, se envía una alerta inmediata. El sistema funciona a través de Arduino, que está conectado a los sensores. Las mediciones se toman cada 5 segundos, y la información se envía a una aplicación en la computadora. Esta solicitud recopila datos de cada sensor, permitiendo un monitoreo preciso del sistema de cultivo de manera remota a través de internet. La aplicación de esta tecnología en la acuicultura ofrece varios beneficios, como un mejor control ambiental, reducción de daños causados por grandes desastres, reducción de costos de gestión ambiental y un mejor crecimiento de los productos acuáticos.

En la investigación de Sousa et al. (2012), se analizó el comportamiento de tilapias del Nilo, que fueron criadas en jaulas de red con dosificadores automáticos de alimento, y alimentadas a diferentes frecuencias: una vez por hora y cada dos horas. Además, se variaron los períodos de alimentación, ya sea durante el día, la noche o ambos. Para llevar a cabo el estudio, se colocaron 18 tanques con un volumen neto de 1.0 m³ cada uno, en un tanque más grande de 2000 m² con 2 metros de profundidad y una renovación del 5% del agua. Se distribuyeron 170 tilapias en cada jaula de 1m³, con un peso inicial promedio de 16.0±4.9g. La tasa de alimentación se ajustó cada 21 días junto con las mediciones biométricas. Los resultados mostraron una diferencia significativa en el peso final entre los distintos tratamientos (P<0.05). La conversión alimenticia fue mejor en el caso de la alta frecuencia de alimentación, es decir, 24 veces al día. Esto podría traducirse en ahorros de hasta 360 kg de alimento por cada tonelada de pescado producido, lo que mejoraría la sostenibilidad económica de la producción de tilapia y sugeriría una menor contaminación ambiental.

En el proyecto de titulación desarrollado por Minta (2022) en opción al grado de Magíster en Electrónica y Automatización, en la Universidad Tecnológica Israel, que tiene por título “Aplicación IoT para el monitoreo de consumo eléctrico residencial utilizando software libre”, el autor se propuso visualizar los parámetros eléctricos del consumo de energía en tiempo real en un dashboard, mediante una arquitectura de alto nivel IoT. Para esto, utilizó en el nivel de percepción: un medidor de energía PZEM-004T-100 A, una Raspberry Pi 4 Model B y un NodeMCU ESP8266; en el nivel de conectividad: Message Queuing Telemetry Transport (MQTT); en el nivel de servicio: Mosquitto e InfluxDB; y en el nivel de aplicación hizo uso de Node-Red y Grafana. Con la implementación del prototipo denominado Smart_Energy, el autor logró realizar el monitoreo de la corriente, voltaje, frecuencia, potencia, factor de potencia, energía, consumo eléctrico y costo de una residencia con una muy buena precisión.

Gracias a la investigación del proyecto antes mencionado, se pudo establecer que una arquitectura de cuatro niveles, sería la más adecuada para el desarrollo del dispensador automático de alimento para tilapias con tecnología IoT, aplicando el protocolo MQTT a nivel de conectividad y Node-Red en el nivel de aplicación.

En el proyecto de Castillo (2022), cuyo título es “Sistema de detección predictiva de fallas de un motor de baja tensión mediante protocolo MQTT y aplicación IoT”, su autor se propuso implementar un sistema que pudiera predecir posibles fallas del motor de un elevador de cangilones, mediante la utilización de una placa de desarrollo NodeMCU ESP32, sensor de temperatura LM35, sensor de vibración ADXL335 y un sensor de corriente STC0123. Utilizó el protocolo MQTT, el broker Mosquitto, Node Red e InfluxDB, con la finalidad de almacenar las variables medidas en una base de datos en la nube, y de esta forma poder visualizar en un dashboard las alertas, en base al análisis de las variables monitoreadas. Este proyecto aportó a ratificar la arquitectura del prototipo, con la utilización del protocolo MQTT y el software de desarrollo libre Node Red.

En el trabajo de Enriquez (2022), para su titulación como Ingenieros de Sistemas por la Universidad Técnica de Machala, los autores plantearon su tema “Desarrollo de dispensador automático de alimento para tilapias utilizando ESP32 CAM”, con la finalidad de implementar un dispensador automático, a través de la utilización de un placa de desarrollo ESP32-CAM, un motor FD RD550, un driver para motor L298N y un panel solar POWOXI 12/1.7W. Sus resultados evidencian la construcción de un dispensador con estructura de acrílico, que utiliza un tornillo sin fin para dispensar el alimento cuatro veces por día. En cuanto a su aporte al presente trabajo, se ha considerado el uso de un motoreductor acoplado a un tornillo sin fin, como mecanismo para el dispensador de alimento, así como también la utilización de un driver L298BN.

Para concluir con este apartado, se puede mencionar que los trabajos revisados han permitido identificar al protocolo MQTT como el más utilizado para proyectos IoT en piscicultura, conjuntamente con la herramienta para programación en hardware, conocida como Node-Red. La existencia de gestores de bases de datos gratuitos en la nube, también han permitido que muchos proyectos puedan hacer uso de estos gestores para almacenar datos que se recopilan de sensores, y una vez que estos datos son procesados a través de una analítica adecuada, pueden servir para la toma de decisiones, que permitan controlar adecuadamente cualquier sistema productivo, y en particular los sistemas relacionados con la piscicultura.

1.2. Proceso investigativo metodológico

El tipo de investigación considerada para la implementación de un dispensador automático de alimento para tilapias con tecnología IoT, es una investigación de tipo "Investigación y Desarrollo" (I+D) enfocada en la ingeniería y la aplicación práctica de tecnologías para resolver un problema específico en un campo particular, en este caso, la acuicultura y la alimentación de peces. La fase de desarrollo y diseño del dispensador de alimentos, implica la identificación de las necesidades específicas de las tilapias en términos de alimentación, la selección de los componentes adecuados para el dispensador, el diseño del sistema de control, la programación del software y la integración de la IoT.

La implementación de un dispensador automático de alimento para tilapias involucra una combinación de métodos teóricos y prácticos que abarcan la ingeniería, la tecnología de sensores y la automatización. Desde un enfoque teórico, se debe realizar un análisis exhaustivo de las necesidades nutricionales de las tilapias, considerando factores como su tamaño, etapa de crecimiento y la cantidad adecuada de alimento. Esto implica la revisión de estudios científicos y recomendaciones de expertos en acuicultura para asegurar que el diseño del dispensador proporcione una alimentación equilibrada y adecuada.

Desde una perspectiva práctica, se emplea la ingeniería de diseño para crear el propio dispensador automático. Esto involucra la selección de componentes de calidad que garantizan la precisión y la confiabilidad de la dispensación de alimentos. Los métodos de ingeniería también se aplican para diseñar el sistema de dosificación que garantiza que el alimento se libere en las cantidades correctas y en los intervalos apropiados. Asimismo, la tecnología de sensores es crucial para monitorear factores ambientales como la temperatura y humedad, ya que estos elementos también afectan el comportamiento alimenticio de las tilapias.

En el ámbito práctico, la integración de la tecnología IoT es esencial para el funcionamiento remoto y la gestión eficiente del dispensador. Esto implica la programación de algoritmos de control que permiten ajustar el horario y la cantidad de alimentación en función del desarrollo cronológico de las tilapias y sus preferencias establecidas. Además, se emplean conocimientos de desarrollo de software para crear una interfaz de usuario que permita a los operadores supervisar y modificar la configuración del dispensador a distancia.

La fase de pruebas y validación es otro componente práctico fundamental. Se realizaron pruebas en condiciones reales, utilizando un estanque de cultivo con tilapias para evaluar la precisión y efectividad del dispensador. Los datos recopilados durante estas pruebas permitirán ajustar los algoritmos de control para optimizar el rendimiento del dispositivo.

Además, la retroalimentación de los operadores y los resultados observados en las tilapias serán esenciales para refinar el diseño y el funcionamiento del dispensador.

La técnica de observación se utiliza en la implementación de un dispensador automático de alimento para tilapias para comprender y analizar el comportamiento alimenticio de las tilapias en respuesta al dispositivo y su entorno. La recopilación de datos detallados a través de la observación contribuye significativamente a la optimización y mejora continua del sistema de alimentación automatizado, asegurando un crecimiento saludable y eficiente de las tilapias en instalaciones de acuicultura.

Con la finalidad de partir con un valor más preciso sobre el peso actual de las tilapias en el estanque de la Quinta Novillo's, se procedió a realizar el pesaje de 68 tilapias, de un total aproximado de 100 existentes, lo que permitió obtener un peso promedio de 87 gramos, que de acuerdo a Baltazar (2004), ubicaría el desarrollo actual de las tilapias en la fase de crecimiento (50 a 150 gr.). El peso promedio obtenido en el proceso de muestreo, sirvió para la configuración inicial de la cantidad de alimento que debe dispensar el prototipo.

La implementación exitosa del dispensador automático de alimento para tilapias con tecnología IoT involucró una metodología de trabajo estructurada y colaborativa. El proceso se dividió en varias etapas interconectadas, que abarcan desde la planificación hasta la fase de prueba y mejora continua. En la etapa inicial, se llevó a cabo una investigación exhaustiva para comprender las necesidades nutricionales de las tilapias en diferentes etapas de crecimiento y los requisitos específicos de la acuicultura. Esta fase teórica proporcionó la base para diseñar un sistema de alimentación equilibrado y eficiente. Simultáneamente, se realizó una revisión de las tecnologías de sensores y actuadores compatibles con la tecnología IoT para monitorear y controlar la alimentación.

Posteriormente, la metodología implicó la fase de diseño y desarrollo del dispensador automático. Se aplicaron principios de mecánica y electrónica para diseñar el mecanismo de dispensación y seleccionar componentes electrónicos adecuados. Paralelamente, se programaron algoritmos de control que permitieron ajustar la cantidad y el horario de alimentación, acorde a las recomendaciones proporcionadas en Baltazar (2004) y (*Manual de Crianza de Tilapia Nicovita – DIGEPESCA | SAG HONDURAS, s. f.*).

La integración de la tecnología IoT fue una etapa clave. Se desarrolló una red de comunicación para conectar el dispensador con una plataforma en la nube, lo que permitiría la transmisión de datos en tiempo real y el acceso remoto para configuración y monitoreo. Se desarrolló una interfaz de usuario amigable para operadores y cuidadores, lo que facilitó la

interacción con el dispensador y la revisión de estadísticas de alimentación y comportamiento. La metodología también incluyó una fase de pruebas y validación. Se instaló el prototipo del dispensador en un entorno de cultivo de tilapias reales. Los datos obtenidos mediante sensores y observación directa se compararon con las pautas nutricionales y los patrones de alimentación esperados. Los ajustes se realizaron en función de los resultados, lo que garantiza una dispensación precisa y eficaz del alimento.

La metodología se articula con una fase de mejora continua. Los datos recopilados a lo largo del tiempo se podrán utilizar para afinar los algoritmos de control y optimizar la operación del dispensador. Los comentarios de los operadores y los resultados de crecimiento de las tilapias también influirán en la evolución constante del sistema.

En resumen, la metodología de trabajo para la implementación de un dispensador automático de alimento para tilapias con tecnología IoT se centra en una combinación de investigación, diseño ingenieril, programación de software y pruebas en entornos reales. La colaboración multidisciplinaria y la adaptación continua en función de los datos recopilados serán esenciales para lograr un sistema de alimentación automatizado y efectivo que beneficiará al cultivo saludable de las tilapias.

CAPÍTULO II: PROPUESTA

2.1 Fundamentos teóricos aplicados

Introducción

La industria de la acuicultura ha experimentado una evolución constante, impulsada por la búsqueda de soluciones eficientes y sostenibles para la crianza de especies acuáticas. La Quinta Novillo's, dentro de su proyecto de desarrollo, ha considerado la crianza de tilapias, como un objetivo importante dentro de su planificación, por lo que la alimentación adecuada y en horarios precisos para sus peces, es un aspecto crucial.

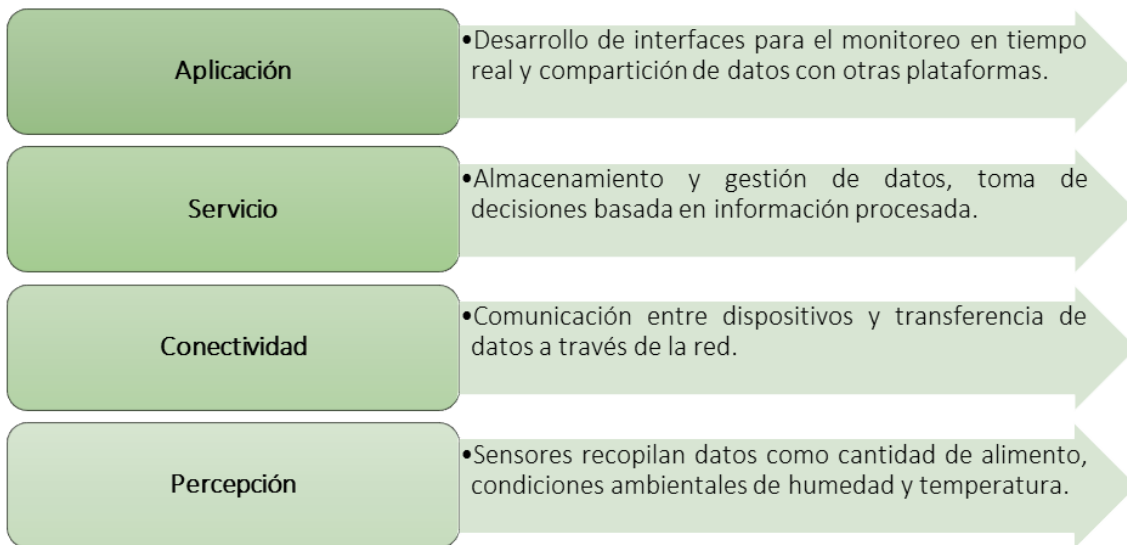
En esta sección, se examinarán los principios teóricos esenciales necesarios para la implementación de un dispensador automatizado de alimento para tilapias mediante tecnología IoT. Por lo cual, se partirá de una estructura arquitectónica de cuatro niveles que facilite la interconexión y colaboración efectiva entre todos los componentes, incluyendo los dispositivos físicos, las plataformas de procesamiento de datos y los sistemas de visualización. Esta configuración arquitectónica será el cimiento sobre el cual se construirá el prototipo central de este trabajo, demostrando la manera en que la tecnología IoT puede converger de manera eficiente con el proceso de crianza de tilapias en Quinta Novillo's.

Arquitectura IoT de cuatro niveles

El punto de partida esencial para la exitosa concepción y desarrollo de prototipos IoT radica en la adopción de una arquitectura orientadora. Esto adquiere especial relevancia debido a las características particulares de los dispositivos involucrados, ya que es crucial contar con funcionalidades específicas que se adapten a las necesidades del proyecto en cuestión, como la automatización precisa y ajustada de la alimentación de tilapias. En la figura 1 se puede observar la arquitectura IoT que ha sido adoptada para el desarrollo del prototipo.

Figura 1

Arquitectura IoT de alto nivel



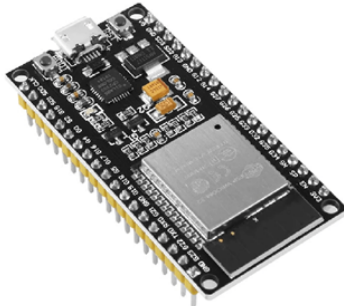
Nivel de percepción

La fase inicial de la estructura arquitectónica establecida se concentra primordialmente en los elementos de Internet de las Cosas (IoT), que engloban tanto los sensores como los actuadores.

NodeMCU-32. La placa de desarrollo NodeMCU-32, ilustrada en la Figura 2, incorpora en una sola placa el módulo SoM ESP-WROOM-32, que se basa en el SoC (System on a Chip) ESP32, junto con el convertidor USB-serial CP2102 necesario para programar el ESP32 a través de USB, así como reguladores de voltaje y LEDs indicadores. Ofrece una combinación de características económicas y un consumo de energía reducido, además de integrar tecnología Wi-Fi y Bluetooth de doble modo. Diseñado para una amplia gama de aplicaciones en el ámbito de la Internet de las cosas (IoT) y la electrónica, el ESP32 utiliza un microprocesador Tensilica Xtensa LX6 con uno o dos núcleos, dependiendo de la variante. Además de su capacidad de conectividad inalámbrica, el chip incluye componentes como interruptores de antena, balun de radiofrecuencia y amplificador de potencia. Todo esto lo convierte en una opción adecuada para proyectos que requieran comunicación inalámbrica y capacidad de procesamiento, como dispositivos inteligentes, sistemas de automatización y más.

Figura 2

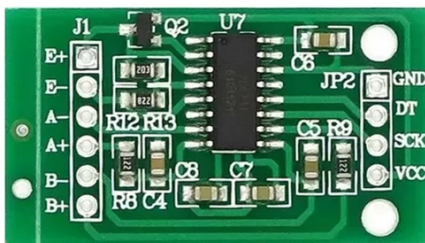
Placa de desarrollo NodeMCU-32



Módulo HX711. El módulo HX711, mostrado en la Figura 3, es un componente utilizado para medir la carga o la presión en aplicaciones como balanzas y sistemas de medición de fuerza. Es compatible con sensores de peso de diferentes capacidades y características. Este módulo incluye un chip HX711 que permite una alta precisión en la medición. Opera a 5V DC con bajo consumo de corriente y admite voltajes de entrada diferenciales pequeños. La resolución de conversión A/D es de 24 bits, lo que permite una detección precisa de cambios en la carga. Además, ofrece una frecuencia de refresco adecuada para una lectura estable y precisa de los datos. El sensor de peso HX711 es ampliamente utilizado en proyectos de electrónica y sistemas de medición que requieren precisión en la detección de carga y presión.

Figura 3

Módulo HX711 para manejo de celda de carga



Celda de carga de 20kg. Una celda de carga de 20kg, indicada en la Figura 4, es un dispositivo utilizado para medir fuerzas y cargas en aplicaciones industriales y de medición. Está diseñada para soportar cargas de hasta 20 kilogramos. Normalmente, consiste en una barra de aluminio con agujeros roscados para montaje y galgas extensiométricas que permiten

medir deformaciones bajo carga. Las celdas de carga de 20kg son utilizadas en aplicaciones como balanzas, sistemas de pesaje industrial y equipos de medición que requieren precisión en la detección de fuerzas. Estas celdas generan una señal eléctrica proporcional a la carga aplicada, lo que permite su integración con sistemas electrónicos y microcontroladores para obtener mediciones precisas.

Figura 4

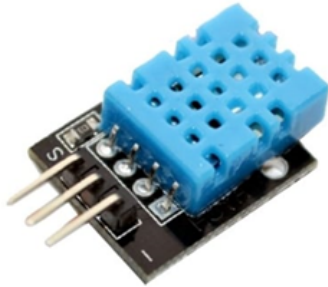
Celda de carga de 20 Kg utilizada en la implementación del prototipo



Sensor de temperatura y humedad DHT11. El sensor de temperatura y humedad DHT11, indicado en la Figura 5, es un dispositivo utilizado para medir tanto la temperatura como la humedad relativa del ambiente en el que se encuentra. Funciona con una fuente de alimentación de 3.3-5V DC y proporciona una señal digital a través de un único bus. Está compuesto por un elemento sensor que utiliza un condensador de polímero para medir la humedad y un termistor para medir la temperatura. El rango de funcionamiento de este sensor es una humedad del 20 al 90% HR y una temperatura de 0 ~ 50 grados Celsius. Además, algunos modelos incluyen un transductor interno de temperatura del tipo NTC para mejorar la precisión. El DHT11 es ampliamente utilizado en aplicaciones de monitoreo ambiental, sistemas climatológicos y control de temperatura y humedad.

Figura 5

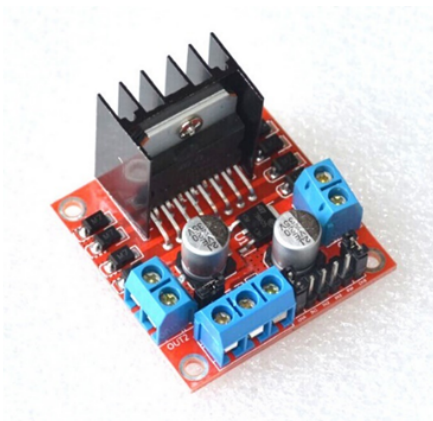
Sensor de temperatura y humedad DHT11



Driver para motor DC L289N. El L289N mostrado en la Figura 6, es un controlador de doble puente en H fabricado por STMicroelectronics. Se utiliza comúnmente para controlar motores de corriente continua y otras cargas en varios proyectos electrónicos. El componente ofrece la capacidad de controlar la dirección y velocidad de los motores. A menudo se emplea en aplicaciones donde se requiere un control preciso de los movimientos del motor. Las discusiones y diseños que involucran al L289N incluyen configuraciones con microcontroladores como ESP32 y STM32 para el control y automatización de motores, así como la integración con plataformas como Home Assistant. La hoja de datos proporciona información técnica detallada sobre el L289N para su uso e interfaz adecuados en proyectos.

Figura 6

Driver L289N utilizado para manejo de un motor reductor



Motor reductor DC de 12V – 3500/100 rpm. Un motor reductor DC de 12 V, con una reducción de velocidad de 3500 a 100 rpm, como el mostrado en la Figura 7, es un dispositivo

que combina un motor de corriente continua con una reducción de velocidad, lo que permite lograr un par de torsión más alto a costa de una velocidad más lenta. Se utiliza para reducir significativamente la velocidad de los motores, siendo común en maquinaria industrial y procesos mecánicos. El valor de 3500 r/min es la velocidad del motor DC a 12 V, y 100 r/min es la velocidad de salida sin carga. La reducción de velocidad se logra mediante un sistema de engranajes. Estos dispositivos son ampliamente usados en diferentes sectores productivos debido a su capacidad para proporcionar mayor par de torsión en aplicaciones que requieren control preciso y velocidad controlada.

Figura 7

Motor reductor DC de 12 V - 3500/100 rpm



Fuente conmutada 12V / 5 A. Una fuente conmutada de 12V y 5A, como la indicada en la Figura 8, es un dispositivo que proporciona una salida de corriente continua de 12 voltios y una corriente máxima de 5 amperios. Estas fuentes son utilizadas para alimentar una variedad de dispositivos electrónicos y proyectos, incluyendo equipos electrónicos, circuitos integrados, tiras de LED, equipos de comunicación y más. Las fuentes conmutadas son eficientes y compactas, ya que utilizan tecnología de conmutación para regular la tensión de salida de manera eficiente. Son ideales para aplicaciones donde se requiere una alimentación estable y precisa. Estas fuentes a menudo se utilizan en proyectos de electrónica, robótica y automatización, así como en sistemas de iluminación y electrónica de consumo.

Figura 8

Fuente conmutada 12 V / 5A



Nivel de conectividad

Este nivel se encarga de establecer vínculos con otros dispositivos físicos, servidores y componentes de redes.

En esta sección, se explorará el protocolo MQTT, que desempeñará un papel fundamental en la implementación de la aplicación planteada.

Message Queuing Telemetry Transport (MQTT). Es un protocolo de conectividad para el Internet de las Cosas (IoT), opera en un enfoque de mensajería de publicación y suscripción extremadamente ligero. Destacado por su eficiencia en situaciones de baja capacidad de ancho de banda, este protocolo encuentra su utilidad en contextos remotos. Su sistema de publicación y suscripción permite a los dispositivos intercambiar mensajes como clientes, promoviendo la comunicación entre múltiples dispositivos. Diseñado específicamente para dispositivos con recursos limitados y conexiones restringidas, MQTT se erige como una solución ideal para escenarios variados de IoT.

Los componentes esenciales en la comunicación MQTT incluyen:

Cliente: dispositivos que utilizan MQTT para comunicarse a través de una pila TCP/IP.

Broker: encargado de gestionar las suscripciones de los mensajes y su posterior envío.

Conexión: se establece entre el cliente y el broker, mediando la comunicación entre ambos.

Dentro de MQTT, el cliente realiza dos acciones fundamentales:

Publicar: Esta operación ocurre cuando el cliente envía información al servidor, y se conoce como "publicación".

Suscribirse: Este proceso se efectúa cuando el cliente recibe datos provenientes del servidor, y se denomina "suscripción".

Nivel de servicio

En este escenario, se incluyen el HiveMQ y MongoDB, encargados de la administración de datos, almacenamiento y procesamiento de información.

HiveMQ. Ofrece un broker MQTT empresarial que está optimizado para implementaciones nativas en la nube, lo que permite el uso eficiente de recursos en la nube y reduce el ancho de banda requerido para mover datos. HiveMQ Cloud Edition es una versión personalizada de MQTT.fx, una herramienta de prueba y depuración, que se integra con HiveMQ Cloud, una plataforma MQTT administrada en la nube. Además, HiveMQ Edge es un gateway de software de código abierto que incluye un broker MQTT optimizado para aplicaciones en el borde, lo que ayuda a simplificar y modernizar la infraestructura de IoT industrial al convertir protocolos propietarios en protocolos estándar. HiveMQ es ampliamente utilizado para la comunicación eficiente en soluciones de IoT y automatización industrial.

Mongo DB. Es un sistema de base de datos NoSQL de código abierto que almacena los datos en estructuras de documentos BSON (similar a JSON) en lugar de tablas. Su esquema es dinámico y puede manejar datos semiestructurados y no estructurados, lo que lo hace adecuado para aplicaciones modernas y escalables. MongoDB es conocido por su capacidad de escalado horizontal, alta disponibilidad y distribución geográfica incorporadas. Ofrece soporte para transacciones multi-documento ACID desde su versión 4.0. MongoDB es ampliamente utilizado en aplicaciones web, móviles e IoT debido a su flexibilidad y rendimiento en la gestión de datos.

Nivel de aplicación

En este nivel se engloban las aplicaciones y servicios de software dirigidos a los usuarios, donde se logra una utilización efectiva de la información que ha sido previamente procesada.

Node -RED. Es una herramienta de desarrollo de flujo basada en gráficos y de bajo código, originalmente desarrollada por IBM. Se utiliza para conectar dispositivos de hardware,

APIs y servicios en línea en el contexto de Internet de las Cosas (IoT). Node-RED ofrece un editor de flujo basado en navegador web, lo que permite a los usuarios crear funciones JavaScript y construir aplicaciones conectando varios elementos. El entorno de ejecución se basa en Node.js. Es una herramienta versátil que se puede instalar localmente, en dispositivos como NodeMCU-32, o en la nube. Node-RED simplifica el proceso de construcción de aplicaciones IoT al proporcionar una interfaz intuitiva para crear, compartir y reutilizar flujos de datos y lógica. Puedes encontrar guías de instalación y uso en el sitio web oficial de Node-RED y documentación para comenzar rápidamente.

PlatformIO. Es un entorno de desarrollo integrado (IDE) amigable y extensible diseñado para el desarrollo de software empujado. Ofrece un conjunto completo de herramientas de desarrollo profesional para agilizar la creación y entrega de productos empujados. Esta plataforma proporciona características modernas y potentes que simplifican el proceso de desarrollo de software para sistemas empujados. PlatformIO es compatible con múltiples plataformas y arquitecturas, lo que lo hace versátil para varios proyectos de microcontroladores y placas de desarrollo. Puedes encontrar más información y recursos en el sitio web de PlatformIO, acceder a su repositorio en GitHub y participar en la comunidad en el foro de Comunidad de PlatformIO para discutir diversos temas relacionados con el desarrollo empujado y prácticas de codificación.

.NET MAUI. Siglas provenientes de “.NET Multi-platform App UI” es un marco de trabajo que utiliza las últimas tecnologías para construir aplicaciones nativas en Windows, macOS, iOS y Android. Abstrae estas plataformas en un marco común basado en .NET, lo que permite crear una única base de código y sistema de proyectos para todos los objetivos de dispositivos. Esto permite desarrollar aplicaciones que se vean y sientan como nativas en las distintas plataformas. “.NET MAUI” utiliza C# y XAML como lenguajes de programación y marcado para la creación de interfaces de usuario. Permite desarrollar aplicaciones para Windows, MacOS, Web y dispositivos móviles (iOS y Android) a partir de un único proyecto.

2.2 Descripción de la propuesta

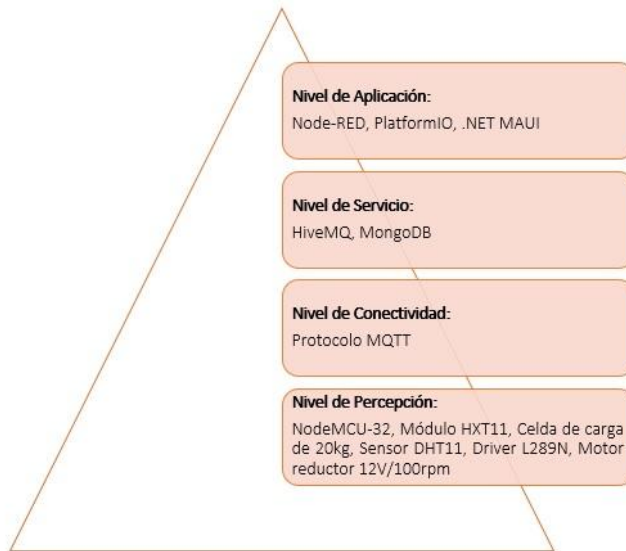
a. Estructura general

La base de este proyecto se fundamenta en la concepción y desarrollo de una aplicación IoT destinada al monitoreo del suministro de alimento en criaderos de tilapias. La efectiva implementación de proyectos IoT está íntimamente vinculada a la elección de una arquitectura de referencia, como se ha planteado previamente. Por esta razón, se ha adoptado

un enfoque de cuatro niveles, como se representa en la Figura 9, donde además se visualiza la relación entre los componentes, el software y los servidores que serán empleados en este proyecto.

Figura 9

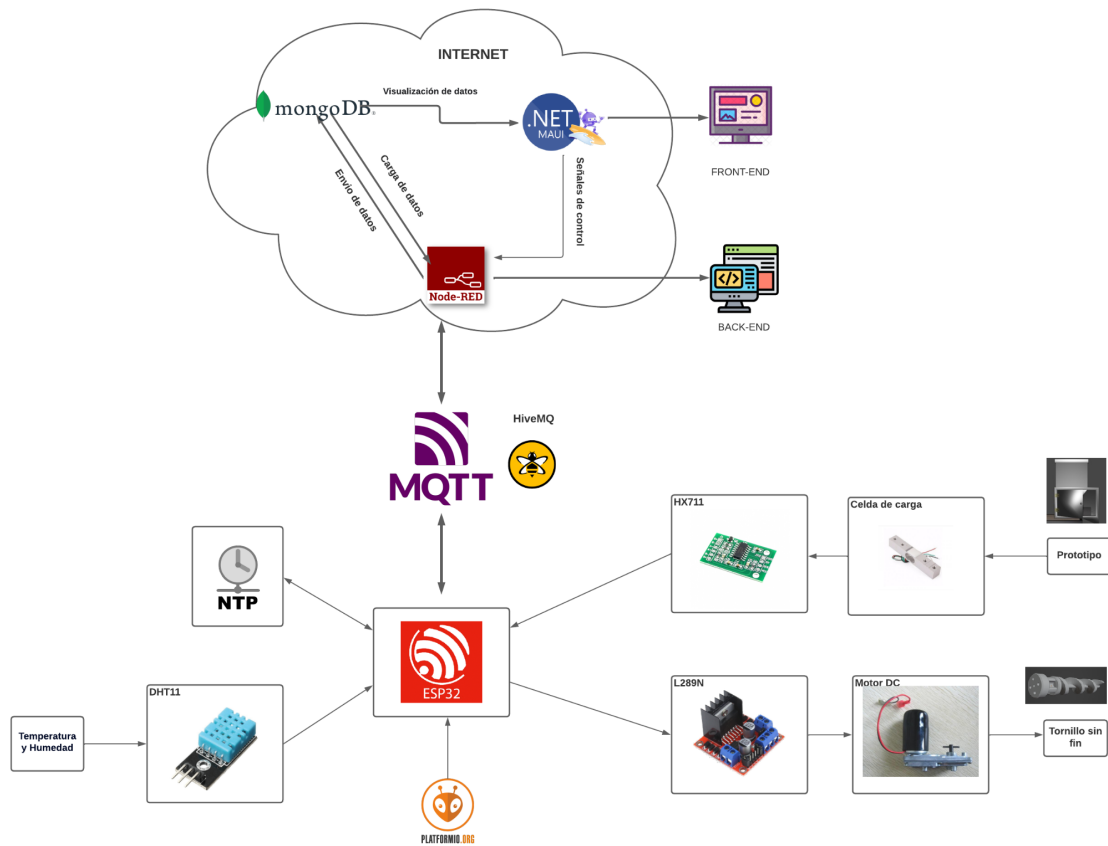
Estructura IoT del prototipo



Siguiendo la arquitectura concebida para el dispensador automatizado de alimento dirigido a las tilapias, en la Figura 10 se evidencia una interacción estratégica del ESP32 con diversos componentes esenciales. Entre estos componentes, se distingue el entorno de desarrollo PlatformIO, ejerciendo su papel fundamental en el backend, junto con componentes como NTP, el sensor DHT11, el módulo HX711 (encargado de la administración de la celda de carga) y el módulo L289N (responsable del control del Motor DC). El esp32 desempeña un rol versátil al efectuar la transferencia de datos hacia la aplicación desarrollada con .NET MAUI, además de su habilidad para suscribirse y publicar información a través del broker HiveMQ. Tras la publicación, los datos son encaminados a Node-RED, una entidad intermediaria que redirige la información a la base de datos MongoDB. El clímax del proceso reside en la visualización de datos, donde .NET MAUI actúa como la frontend, consolidando un sistema integral que abarca captura, procesamiento, transferencia y presentación de datos, fundamentando la tecnología en el dispensador automatizado de alimento para tilapias.

Figura 10

Arquitectura del dispensador automatizado para tilapias



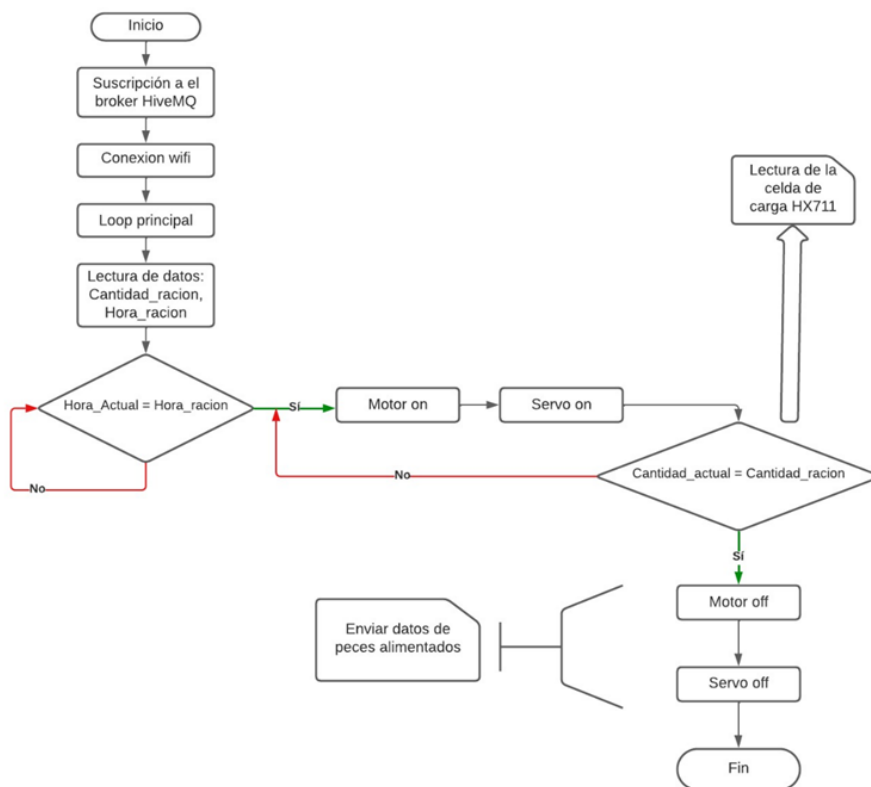
Igualmente, se desarrolló un diagrama de flujo que muestra la lógica aplicada para la implementación de la aplicación IoT (véase Figura 11). Inicialmente, se ejecutan tres acciones fundamentales: la conexión a la red, la suscripción al bróker y la actualización horaria a través de un servidor NTP. Una vez que estos pasos iniciales han sido completados, el sistema entra en un ciclo continuo donde se verifica constantemente la presencia de datos pendientes provenientes del bróker MQTT hacia el dispositivo. Estos datos incluyen información como la cantidad y el momento de la dosificación del alimento para las tilapias. Al detectar un dato pendiente, el sistema procede a procesarla. Posteriormente, se verifica si la hora indicada por el RTC coincide con alguna de las horas predefinidas por el usuario. Si se produce una coincidencia, se activa el motor que impulsa el tornillo sinfín, responsable de transportar el alimento desde el final de la tolva del prototipo hacia el estanque.

El módulo HX711 lleva a cabo mediciones en curso hasta que el peso del alimento alcance o exceda el peso deseado. En ese momento, se envía una señal al motor para detener la rotación del tornillo sinfín. Tras un breve intervalo, se realiza una última medición del peso

del alimento contenido en la tolva, para asegurar que el peso no sea menor al establecido por el usuario. Si la condición sigue cumpliéndose, se dispensa el alimento hacia el estanque de las tilapias. Si la condición no se cumple, se reactiva el motor y se repite el proceso anterior. Una vez concluido este proceso, el dispositivo publica los datos al bróker.

Figura 11

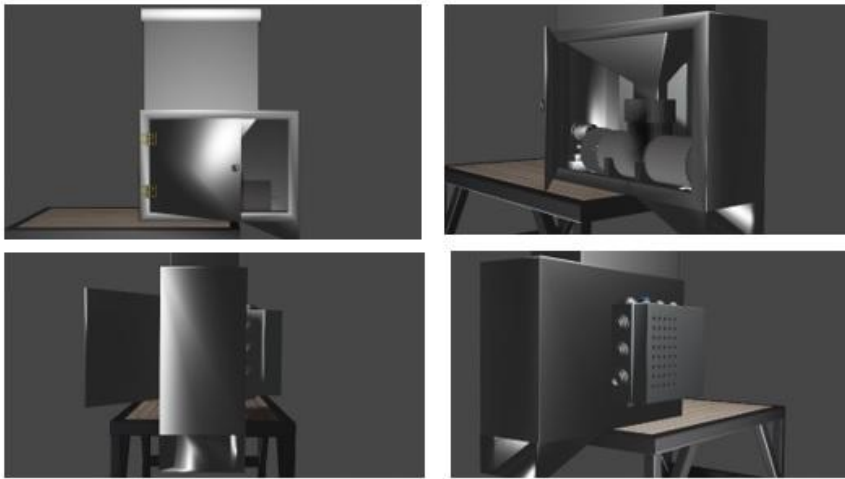
Diagrama de flujo para el dispensador automatizado para tilapias



Se elaboró un diseño a través de la herramienta Blender, para la modelación del prototipo, tal como se puede apreciar en la Figura 12. Blender destaca por su versatilidad y por ser una plataforma de código abierto que permite a los usuarios generar y editar objetos tridimensionales de forma eficiente y creativa.

Figura 12

Modelo 3D del dispensador automatizado para tilapias

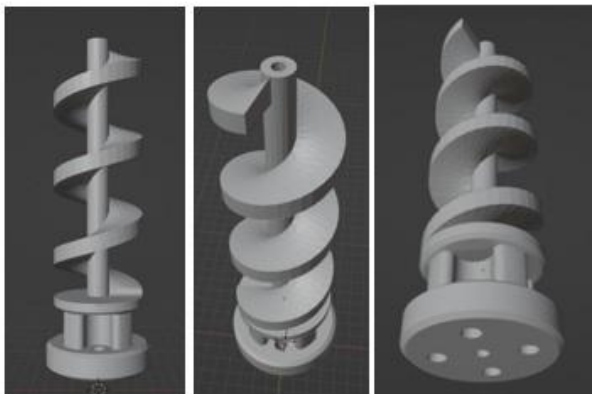


Para observar imágenes reales del prototipo implementado se puede revisar el Anexo 4.

También se abordó la tarea de trabajar en un modelo de tornillo infinito en 3D (véase Figura 13), obtenido de la plataforma "Thingiverse", reconocida por ser un espacio en línea donde se comparten y alojan archivos de diseño en 3D sin costo alguno. Una vez que se realizaron las adecuaciones a una altura de 240 mm, y un diámetro de 762 mm, se procedió a imprimir el tornillo sin fin en una impresora "Creality, Modelo Ender-3 S1 Pro", para su posterior montaje en el prototipo.

Figura 13

Modelo 3D del tornillo sin fin



b. Explicación del aporte

Para dar inicio al desarrollo de la aplicación IoT destinada al monitoreo del consumo de alimento en el dispensador automatizado para tilapias con tecnología IoT, se emprendieron los siguientes pasos:

Lectura de sensores

Uno de los componentes de vital importancia en la aplicación IoT, es el módulo de celda de carga HX711, que tiene la responsabilidad de capturar y cuantificar con precisión el peso del alimento presente en el dispensador automatizado de alimento para tilapias. En la figura 14 se observa su conexión al ESP32.

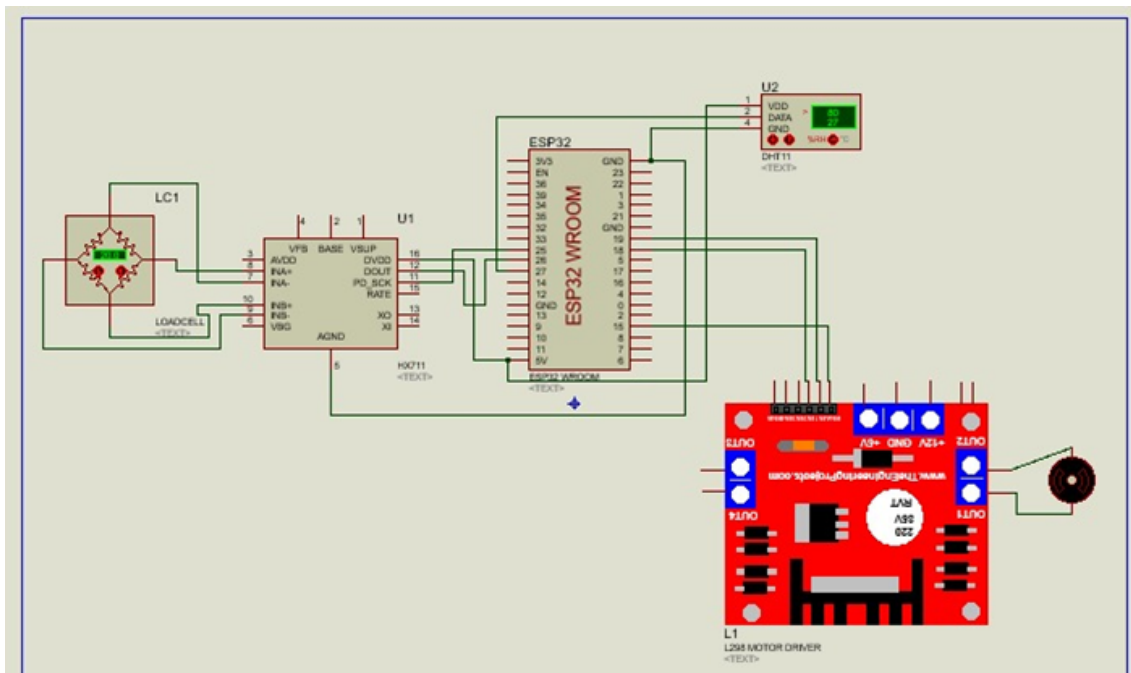
Otro de los sensores es el DHT11, que se encarga de medir y proporcionar datos precisos sobre la temperatura y la humedad del entorno en el que se encuentra el dispensador y las tilapias. En la Figura 14 también se observa su conexión al ESP32

Actuador del prototipo

El driver que permite controlar al motor reductor de 12 Voltios DC, es el L289N. Este driver facilita la interfaz entre la placa de desarrollo ESP32 y el motorreductor que se utiliza para dispensar el alimento en el sistema, como se puede observar en la Figura 14. Las señales para activación del motorreductor parten desde el ESP32, hasta las señales de activación correspondientes en el L298N, para de esta forma gracias a la polaridad adecuada de las señales de activación, el motorreductor gire en el sentido adecuado para producir la salida del alimento.

Figura 14

Diagrama de conexión del prototipo



Fuente de alimentación para el prototipo

Se utilizó una fuente conmutada de 12V/5A, para a través del driver L298N alimentar al motor reductor de 12 V, que se encarga de mover el tornillo sin fin para producir la dispensación del alimento.

Gracias a la arquitectura del driver L298N, una vez que se alimenta con 12 V, se obtiene una salida de alimentación de 5V, que sirve para energizar al NodeMCU-32, al sensor DHT11 y al driver de la celda de carga, que es el módulo HX711.

Montaje de componentes electrónicos

Se utilizó un contenedor plástico para el montaje de la electrónica del prototipo, con la finalidad de independizar el componente electrónico, de la parte mecánica del mismo. En la Figura 15 se puede observar las conexiones físicas de los sensores, drivers, y demás componentes del circuito electrónico, con la placa NodeMCU-32.

Figura 15

Conexión física de los componentes electrónicos



Revisar Anexo 2, para observar diseño de placa de circuito impreso.

Captura de datos y programación del ESP32

En cuanto a la programación del ESP32, es muy importante la instalación de ciertas librerías la cual permitirá la comunicación mediante la interfaz, para lo cual se declaró las variables para los distintos parámetros a ser medidos (ver Figura 16)

Figura 16

Declaración de variables

```
//PIN BALANZA
HX711 balanza;
//CONEXION MQTT PUBLICACION
WiFiClient espClient;
PubSubClient client(espClient);
//CONEXION MQTT ESCUCHA
WiFiClient espClient2;
PubSubClient client2(espClient2);
// PIN DE TEMPERATURA Y HUMEDAD
DHT dht;
//VARIABLES DE HORA
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");
//CONTROLADOR DE HILOS
Scheduler runner;
//LISTA DE HORARIOS
String horarios[10] = {
  "08:00",
  "12:00",
  "16:00"
};
```

Programación del ESP32 para la comunicación.

Para establecer la comunicación, se configuró el ESP32 utilizando la plataforma Platform.io. Esta configuración involucra la instalación de la biblioteca PubSubClient.h y otras bibliotecas necesarias que permiten la transmisión y recepción de mensajes MQTT. Para llevar a cabo este proceso, se empleó el broker HiveMQ como intermediario, tal como se representa en la Figura 17.

Figura 17

Programación de MQTT

```
#include <Arduino.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include "HX711.h"
#include "DHT.h"
#include <EEPROM.h>
#include <TaskScheduler.h>
#include <WiFiUdp.h>
#include <TimeLib.h>
#include <NTPClient.h>
#include <ThingSpeak.h>

void conectar_mqtt() {
  Serial.println("Conectando a broker MQTT... (hilo de publicacion)");
  if (client.connect("ESP32Client")) {
  } else {
    Serial.print("MQTT connection failed, rc=");
    Serial.println(client.state());
  }
}
```

Se utilizó el puerto 1883 para enviar los datos al broker, enfocándose en los topics. El cliente envía datos al topic "tesisM/datos sensores", como se muestra en la Figura 18.

Figura 18

Publicación de los topics MQTT

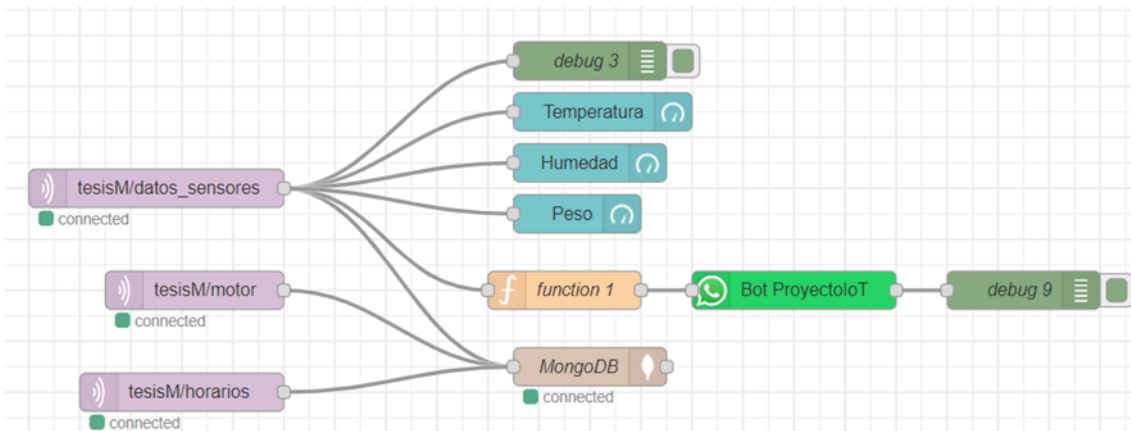
```
//Publicar Humedad, Temperatura, Peso
float h = dht.getHumidity();
float t = dht.getTemperature();
Serial.println("Humedad: " + String(h) + "% Temperatura: " + String(t) + "°C Peso:" + String(peso));
String datos = String(h) + ";" + String(t) + ";" + String(peso);
client.publish("tesisM/datos_sensores", datos.c_str());
// Publicar estado del motor
if (digitalRead(2) == HIGH) {
  client.publish("tesisM/motor", "ON");
} else {
  client.publish("tesisM/motor", "OFF");
}
// Publicar horarios
String horariosStr = "";
for (int i = 0; i < cantidadElementos; i++) {
  horariosStr += horarios[i];
  if (i < cantidadElementos - 1) {
    horariosStr += ";";
  }
}
client.publish("tesisM/horarios", horariosStr.c_str());
}
```

El código completo para la programación del NodeMCU-32 se puede observar en el Anexo 3.

Como se puede observar en la Figura 19, el “mqtt-in” tiene la finalidad de transmitir los datos generados por la programación del ESP32 a través de Platform.io, al flujo dentro de Node-RED. El nodo “Mqtt in” (tesisM/datos_sensores, tesisM/motor, tesis/horarios) permite enviar datos al gauge, function 1 y al mongoDB. El nodo “Gauge” (Temperatura, Humedad, Peso) sirve para visualizar los datos de los sensores. El nodo “Function” (function 1) sirve para el envío de datos al bot de WhatsApp. El nodo “Mongodb-aleph-ou”t , permite la conexión al mongo db. Finalmente, el nodo “send message” permite configurar el Bot en WhatsApp.

Figura 19

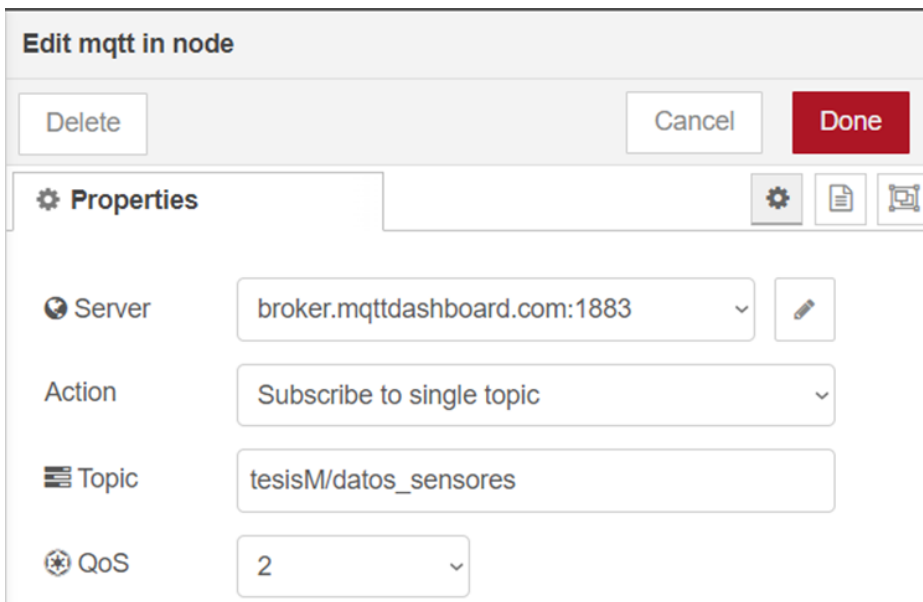
Flujo en node-red para la comunicación MQTT



El procedimiento arranca con la configuración del nodo "mqtt in", el cual habilita la suscripción a mensajes asociados con un tema particular. En la Figura 20 se observa el procedimiento que permite seleccionar el servidor y el topic al cual se desea suscribirse.

Figura 20

Configuración del nodo "mqtt in", primera parte:



En la segunda parte de la configuración del nodo "mqtt in", es indispensable definir la dirección del servidor (Server) y seleccionar el protocolo de comunicación (Protocol), tal como se puede apreciar en la representación visual de la Figura 21.

Figura 21

Configuración del nodo "mqtt in", segunda parte:

The screenshot shows a configuration window with three tabs: "Connection", "Security", and "Messages". The "Connection" tab is selected. It contains the following fields and options:

- Server:** A text input field containing "broker.mqttdashboard.com".
- Port:** A text input field containing "1883".
- Connect automatically:** A checked checkbox.
- Use TLS:** An unchecked checkbox.
- Protocol:** A dropdown menu showing "MQTT V3.1.1".

Base de datos en MongoDB

Para acceder a la base de datos que es MongoDB se debe llenar ciertos campos, como lo son el host, el nombre de la base de datos, su username y password, como se ve en la figura 22.

Figura 22

Configuración del nodo MongoDB

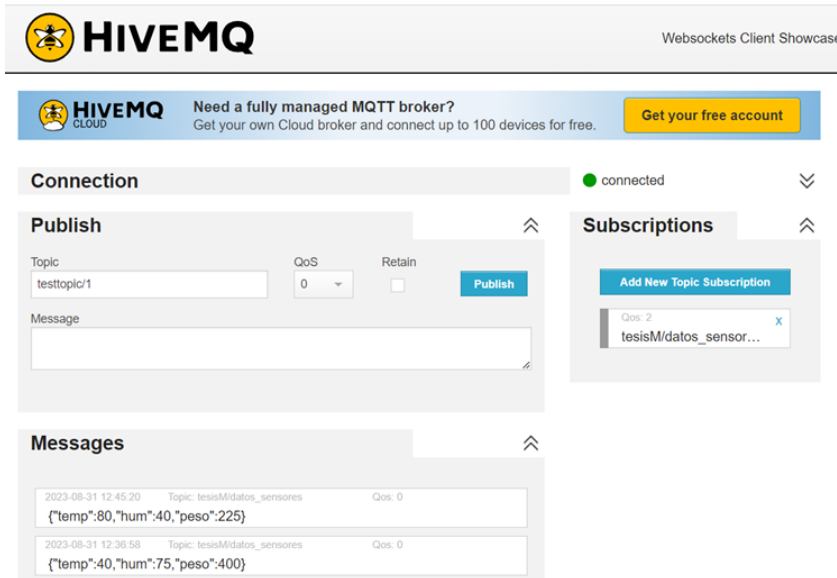
The screenshot shows a configuration window titled "Edit mongodb-aleph-out node > Edit mongodb-aleph-config node". It has three buttons at the top: "Delete", "Cancel", and "Update". Below the buttons is a "Properties" section with the following fields:

- Host:** A text input field containing "cluster0.obyewwj.mongodb.net".
- Connection topology:** A dropdown menu showing "DNS Cluster (mongodb+srv://", with a downward arrow.
- Connect options:** An empty text input field.
- Database:** A text input field containing "Sensores".
- Username:** A text input field containing "JohnnyNovillo".
- Password:** A text input field containing ".....".

Cuando el código de la figura 21 este en ejecución, los datos obtenidos se enviarán al HiveMQ como se visualiza en la figura 23

Figura 23

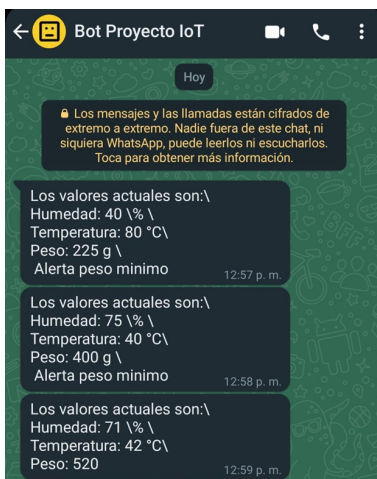
Visualización de datos en HiveMQ



Por otro lado, cuando se esté ejecutando la aplicación de escritorio desarrollada, se enviarán datos al node-red, que a su vez generará una alerta a través del bot de WhatsApp, cuando se alcance la condición de mínima cantidad de alimento aceptable en el dispensador, tal como se puede observar en la figura 24.

Figura 24

Visualización de datos en el bot de WhatsApp

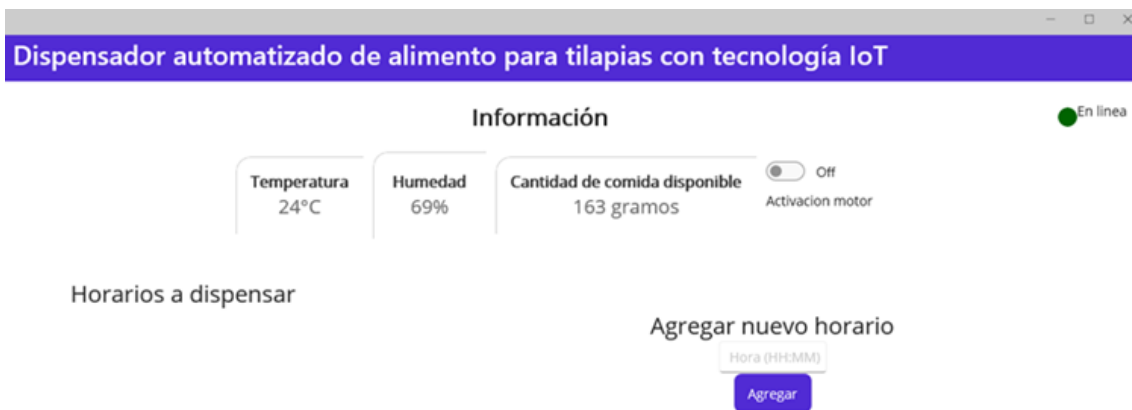


Presentación de parámetros en .NET MAUI

Se realizó una aplicación de escritorio usando .NET MAUI, para visualizar la temperatura, humedad y la cantidad de alimento disponible. También se pueden asignar horarios, activar y apagar el motor, tal como se puede apreciar en la figura 25.

Figura 25

Interfaz de la aplicación de escritorio



c. Estrategias y/o técnicas

El desarrollo del prototipo de dispensador automatizado de alimento para tilapia utilizando tecnología IoT se realizó utilizando un enfoque metodológico que priorizó la adaptabilidad y la ordenación sistemática de tareas. El proyecto se dividió en etapas lógicas, comenzando con la programación de ESP32, seguido del desarrollo de la aplicación del controlador con Visual Studio 2022 y .NET MAUI, y las pruebas, aunque no se adoptó por completo una metodología ágil como Scrum. Se mantuvo el equilibrio entre flexibilidad y estructura, y al mismo tiempo se permitieron ajustes iterativos y mejoras continuas en respuesta a nuevos hallazgos y dificultades.

Para maximizar la eficiencia y la calidad, se utilizaron importantes herramientas tecnológicas durante todo el proceso. El ESP32 se programó utilizando Visual Studio Code y la extensión PlatformIO, que aprovechó la interfaz fácil de usar y la adaptabilidad del hardware de PlatformIO. Se seleccionó Visual Studio 2022 con .NET MAUI para el desarrollo de la aplicación del controlador, con el fin de crear de manera efectiva una interfaz de usuario consistente en

varias plataformas. Se utilizó el protocolo MQTT para establecer comunicación entre ESP32 y la aplicación, lo que permitió una comunicación eficiente en tiempo real entre dispositivos y aplicaciones de IoT. Al garantizar una programación sólida, una interfaz de usuario útil y una comunicación fluida entre los componentes, estas herramientas ayudaron a la ejecución exitosa del proyecto.

2.3 Validación de la propuesta

La selección de expertos se basó en la evaluación de candidatos que cumplen con los siguientes requisitos: una sólida formación académica en el área de investigación pertinente, experiencia tanto académica como profesional en la gestión administrativa, y una clara disposición para participar activamente. A continuación, se proporciona una tabla que detalla la información de los individuos elegidos para validar el modelo.

Tabla 1

Descripción de perfil de validadores

Nombres y apellidos	Años de experiencia	Titulación Académica	Cargo
Dixys Hernández Rojas	30	Doctor en Tecnología Información de Comunicaciones Radio móviles	Docente titular, investigador
Eduardo Tusa Jumbo	13	Doctorado en Señales, Imágenes, Voz, Telecomunicaciones	Docente/Investigador
Iván Ramírez Morales	16	Doctor en Tecnologías de la Información y las Comunicaciones	Profesor titular de la Universidad Técnica de Machala

La evaluación realizada por cada uno de los especialistas se presenta en la Tabla 2, y las valoraciones completas se pueden revisar en el Anexo 7.

Tabla 2

Evaluación realizada por los especialistas

EVALUACIÓN SEGÚN IMPORTANCIA Y REPRESENTATIVIDAD						
Especialista: Ivan Eduardo Ramirez Morales						
CRITERIOS	En Total	En	Ni de Acuerdo Ni en	De	Totalmente de	
	desacuerdo	Desacuer	desacuerdo	acuerdo	acuerdo	
		do				
Impacto						X
Aplicabilidad						X
Conceptualización						X
Actualidad						X
Calidad						
Técnica						X
Factibilidad						X
Pertinencia						X
EVALUACIÓN SEGÚN IMPORTANCIA Y REPRESENTATIVIDAD						
Especialista: Dixys Leonardo Hernández Rojas						
CRITERIOS	En Total	En	Ni de Acuerdo Ni en	De	Totalmente de	
	desacuerdo	Desacuer	desacuerdo	acuerdo	acuerdo	
		do				

Impacto	X
Aplicabilidad	X
Conceptualización	X
Actualidad	X
Calidad Técnica	X
Factibilidad	X
Pertinencia	X

EVALUACIÓN SEGÚN IMPORTANCIA Y REPRESENTATIVIDAD

Especialista: Eduardo Alejandro Tusa Jumbo

CRITERIOS	En Total desacuerdo	En Desacuerdo	Ni de Acuerdo Ni en desacuerdo	De acuerdo	Totalmente de acuerdo
Impacto					X
Aplicabilidad					X
Conceptualización					X
Actualidad					X
Calidad Técnica					X
Factibilidad					X

Pertinencia

X

2.4 Matriz de articulación de la propuesta

En la matriz siguiente se resume la conexión del producto desarrollado con las bases teóricas, enfoques metodológicos, estrategias técnicas y tecnológicas utilizadas.

Tabla 3

Matriz de articulación

<i>Ejes o partes principales del proyecto</i>	<i>Breve descripción de los resultados de cada parte</i>	<i>Sustento teórico que se aplicó en la construcción del proyecto</i>	<i>Metodologías, herramientas técnicas y tecnológicas que se emplearon</i>
<i>Definición de 1 arquitectura IoT de cuatro niveles</i>	<i>1.1. Determinación de software y de hardware basadas en consideraciones de interoperabilidad y costo.</i>	<i>Se basa en los principios de la Industria 4.0, el Internet de las Cosas y las comunicaciones inalámbricas.</i>	<i>La investigación bibliográfica y el análisis de fuentes de empresas desarrolladoras de hardware y software fueron empleados para tomar decisiones informadas.</i>
	<i>1.2. Indagación de proformas de dispositivos.</i>		

1.3. Adquisición de dispositivos

Diseño y programación de arquitectura IoT

2.1. Esquema de bloques para registro de variables.

2.2. Instalación de HiveMQ, Node-Red, MongoDB y .NET MAUI.

2.3. Programación del ESP32.

2.4. Creación de base de datos en MongoDB.

2.5. Programación de la aplicación en .NET MAUI.

Se aprovechan protocolos de comunicación MQTT y tecnologías de Software Defined Networks.

La instalación de servidores HiveMQ y Node-RED, así como la creación de la base de datos en MongoDB, se completaron. La plataforma .NET MAUI se utilizó para construir la aplicación de visualización

3 Implementación de aplicación IoT

3.1. Conexiones físicas de dispositivos.

3.2. Integración de software y hardware de la aplicación.

Conceptos clave involucran Sensores y Actuadores, además de Protocolos de Comunicación.

Los manuales técnicos y servicios de internet fueron aprovechados. La programación y las herramientas de software del paso anterior se reutilizaron. Excel y Google Drive también se emplearon.

3.3. Panel de monitoreo de variables.

3.4. Base de datos de variables monitoreadas.

2.5 Análisis de resultados. Presentación y discusión.

La aplicación IoT asignada con el nombre de *Dispensador automatizado de alimento para tilapia con tecnología IoT* fue implementada en la Quinta Novillo's, ubicado en el sitio Río Chico, parroquia Bellamaría del Cantón Santa Rosa. El prototipo desarrollado se instaló en un estanque de tilapias y mediante NODE-RED se comprobó el establecimiento de la comunicación con el topic (ver Figura 26 y Figura 27).

Figura 26

Comprobación de comunicación del dispositivo de borde con HiveMQ.

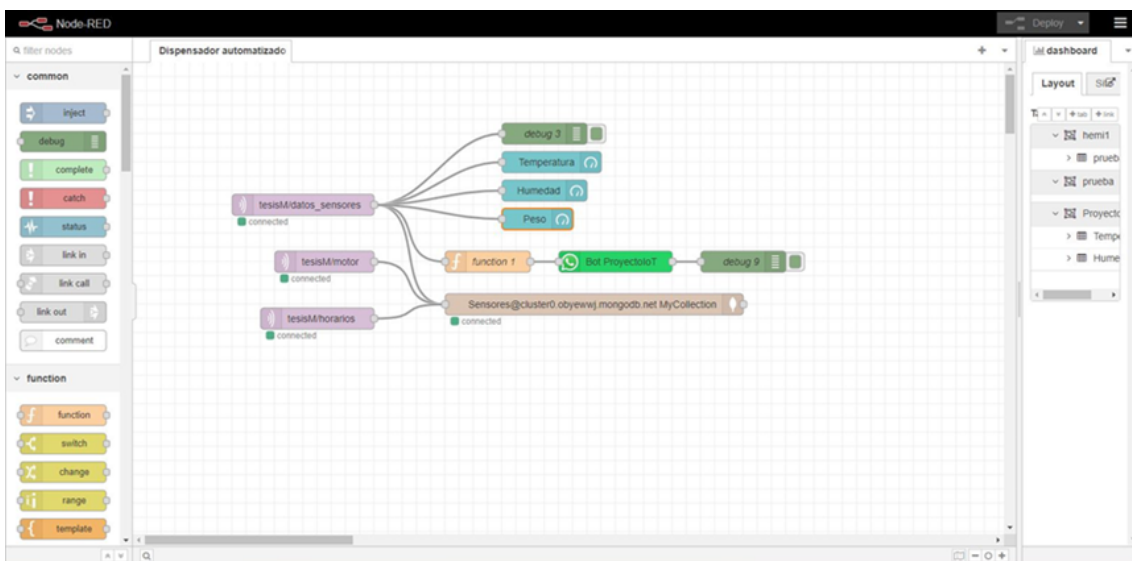
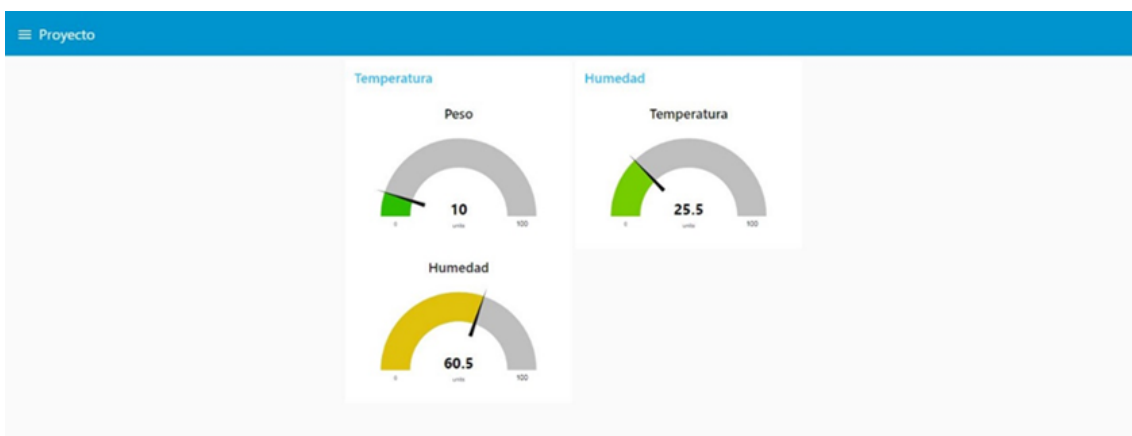


Figura 27

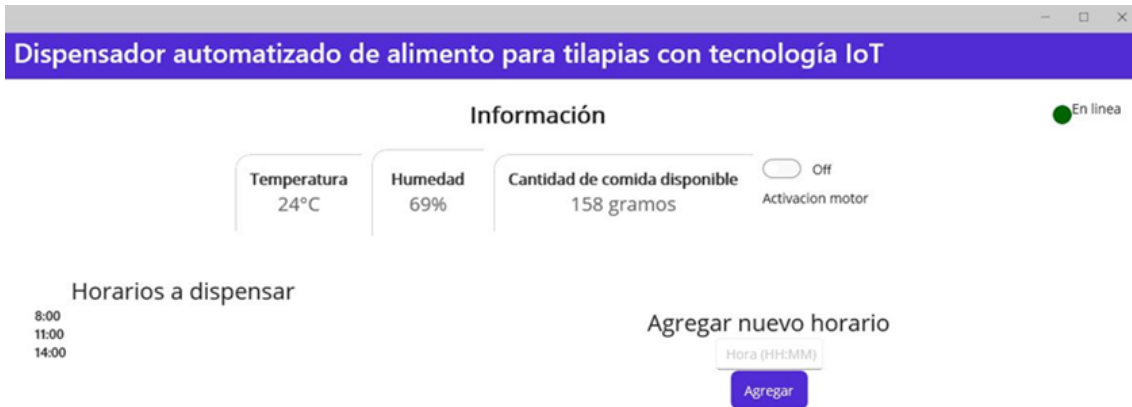
Visualización de los datos en el dashboard de NODE-RED.



Seguidamente se efectuó el monitoreo en la aplicación de .NET MAUI de los siguientes datos: temperatura y humedad, peso en gramos del alimento disponible, horarios de alimentación y activación manual del motor (ver Figura 28).

Figura 28

Visualización de los datos en la aplicación de .NET MAUI



Prueba 1: temperatura

En esta sección, se presentan los resultados de la prueba realizada para evaluar la precisión y fiabilidad del sensor de temperatura y humedad implementado en el sistema. El objetivo principal de esta prueba fue verificar si los valores medidos por el sensor se asemejan significativamente a las condiciones reales de temperatura y humedad proporcionadas por el sistema de aire acondicionado en una sala de laboratorio.

La prueba se llevó a cabo en una sala de laboratorio cerrada, donde el sistema de aire acondicionado estaba activo y controlando la temperatura del entorno. El sensor de temperatura y humedad se colocó en una ubicación estratégica dentro de la sala, garantizando su exposición a las mismas condiciones que el termostato del aire acondicionado monitoreaba y regulaba.

Los valores medidos por el sensor de temperatura y humedad se recopilaron y visualizaron en la aplicación desarrollada en .NET MAUI. Esta aplicación permitió la observación en tiempo real de las lecturas del sensor, lo que facilitó la comparación con los valores

proporcionados por el sistema de aire acondicionado, tal como se puede apreciar en la Figura 29 y Figura 30.

Figura 29

Prueba 1 de temperatura proporcionada por el sistema de aire acondicionado y el sensor DTH11

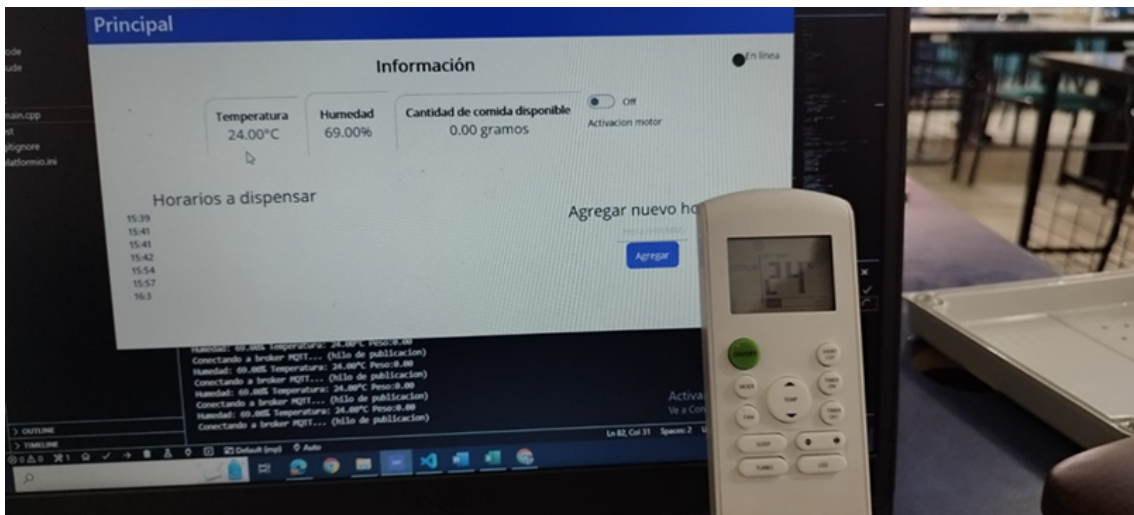
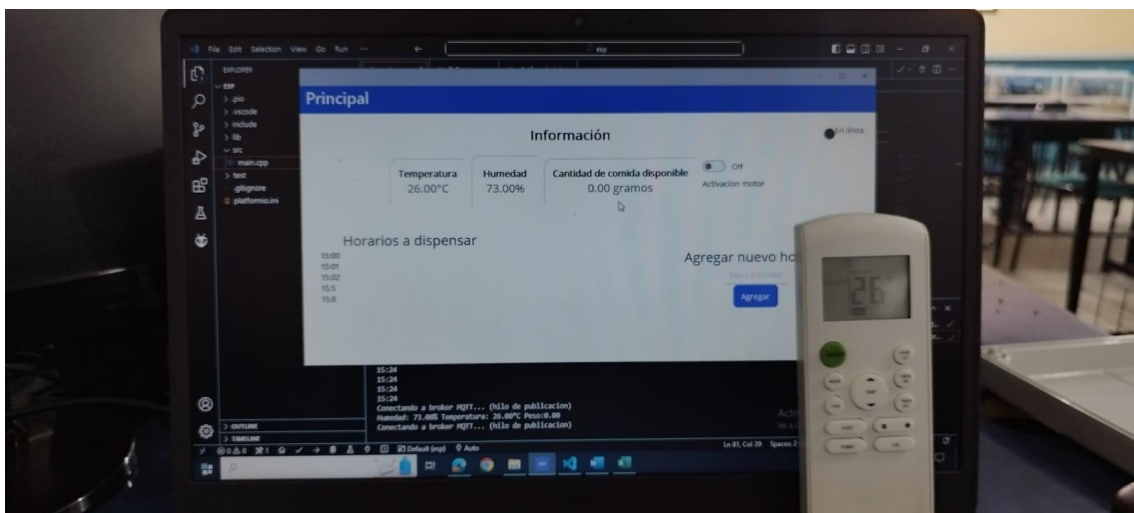


Figura 30

Prueba 2 de temperatura proporcionada por el sistema de aire acondicionado y el sensor DTH11



Durante las pruebas, se registraron las lecturas de temperatura y humedad, tanto del sensor de temperatura del prototipo, como del sistema de aire acondicionado. En la Tabla 4, se resumen los resultados de las pruebas efectuadas.

Tabla 4*Resultados y comparación de temperatura*

Tiempo	Sensor prototipo (Temperatura)	Aire Acondicionado (Temperatura)	Error absoluto %
15:24	26°C	26°C	0.00
16:03	24°C	24°C	0.00

Los resultados obtenidos de la prueba indican una igualdad entre las lecturas proporcionadas por el sensor de temperatura y humedad y las lecturas del sistema de aire acondicionado. Esto sugiere que el sensor implementado es capaz de medir con precisión la temperatura y humedad del entorno, ofreciendo resultados consistentes con las condiciones reales.

La aplicación desarrollada en .NET MAUI proporcionó una interfaz intuitiva para visualizar las lecturas del sensor en tiempo real, lo que facilitó la comparación y el análisis de los datos. La capacidad de observar las lecturas simultáneamente permitió una validación efectiva de la precisión del sensor.

En resumen, la prueba demostró que el sensor de temperatura y humedad implementado en el sistema es confiable y ofrece mediciones coherentes con las condiciones proporcionadas por el sistema de aire acondicionado. Los resultados validan la capacidad del sensor para proporcionar información precisa y útil sobre las condiciones ambientales en la habitación.

Prueba 2: Peso

En esta sección, se exponen los resultados obtenidos de la prueba realizada para evaluar la precisión y confiabilidad del sensor de peso, que se encuentra conformado por una celda de carga de 20 Kg y el módulo HX711. El propósito central de esta prueba fue verificar si los valores medidos por el sensor de peso se correlacionaban de manera significativa con los valores obtenidos de una balanza digital.

La prueba se efectuó utilizando una balanza digital calibrada como punto de referencia. Se llevaron a cabo dos pruebas distintas utilizando cantidades de alimento para peces. Durante

cada prueba, se colocó la misma cantidad de alimento en la balanza digital y en el sensor de peso HX711, garantizando la comparación de las mediciones bajo las mismas condiciones.

Los valores medidos por el sensor de peso del prototipo se obtuvieron y visualizaron en la aplicación creada en .NET MAUI. Esta aplicación permitió una observación en tiempo real de las lecturas del sensor, facilitando la comparación con los valores proporcionados por la balanza digital CAMRY con precisión de 1 gramo.

En la Figura 31 y Figura 32 se pueden observar los valores obtenidos en la primera prueba de peso realizada.

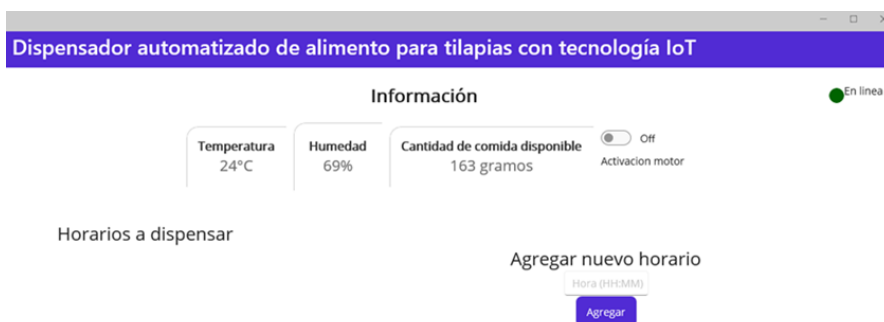
Figura 31

Prueba 1 balanza digital



Figura 32

Prueba 1 sensor HX711



En la Figura 33 y Figura 34 se pueden observar los valores obtenidos en la segunda prueba realizada.

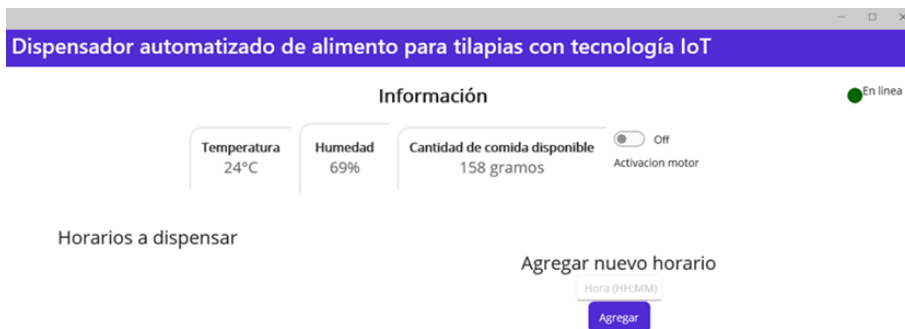
Figura 33

Prueba 2 balanza digital



Figura 34

Prueba 2 Sensor de peso HX711



En la Tabla 5, se presentan los resultados de las dos pruebas realizadas y la comparación entre las lecturas del sensor de peso del prototipo y la balanza digital:

Tabla 5

Resultados y comparación de peso

Prueba	Balanza Digital (g)	Sensor de prototipo (g)	Error absoluto %
Primera	162	163	0.62
Segunda	156	158	1.28

Los resultados obtenidos de las pruebas indican que el sensor de peso del prototipo proporciona mediciones muy similares a las de la balanza digital. Las diferencias en los valores de peso entre el sensor y la balanza son mínimas y se encuentran dentro de un rango de variación aceptable. Estas discrepancias podrían deberse a la precisión propia de la balanza digital o a pequeñas variaciones en la colocación del alimento sobre las plataformas de medición.

La aplicación desarrollada en .NET MAUI permitió una visualización en tiempo real de las lecturas del sensor de peso del prototipo, lo que simplificó la comparación y el análisis de los datos. Esta visualización simultánea facilitó una evaluación efectiva de la precisión del sensor.

En resumen, los resultados de las pruebas confirman la capacidad del sensor de peso del prototipo para medir con precisión el peso del alimento para tilapias. Las mediciones proporcionadas por el sensor son coherentes con las de la balanza digital, lo que valida su confiabilidad en la aplicación propuesta. El sistema implementado ofrece una manera efectiva de monitorear el peso del alimento y tomar decisiones informadas en la alimentación de los peces.

Prueba 3: velocidad de dispensación de alimento

Esta sección presenta los resultados de las pruebas realizadas para determinar la velocidad con la que el dispensador automatizado entrega el alimento. El objetivo principal de estas pruebas fue evaluar la eficiencia del proceso de dispensación y determinar el tiempo promedio necesario para dispensar una cantidad específica de alimento.

Se realizaron un total de cinco pruebas individuales para pesar el alimento que dispensaba el dispensador en cierto tiempo, tal como se puede apreciar en el Anexo 5.

Los resultados de tiempo obtenidos se registraron y se contrastaron con las cantidades de alimento dispensadas en cada prueba. Estos datos se encuentran tabulados en la Tabla 6, y

proporcionaron información esencial para determinar la velocidad promedio en la dispensación de alimento concentrado para tilapias.

Tabla 6

Resultados de las pruebas de velocidad de dispensación

Prueba	Cantidad de Alimento (g)	Tiempo de Dispensación (s)	Velocidad (g/s)
1	151	3.37	44.865
2	150	3.26	46.012
3	150	3.33	44.844
4	156	3.65	42.739
5	154	3.32	46.396
Promedio	153.4	3.39	45.163

Las pruebas de velocidad de dispensación del dispensador automatizado, demostró que el sistema es capaz de dispensar cantidades específicas de alimento de manera eficiente y confiable. Los resultados indican que el sistema es consistente en su velocidad de dispensación, lo que es esencial para garantizar una alimentación precisa y controlada en aplicaciones como la acuicultura.

Resultados de pruebas

En esta sección, se presentan los resultados conjuntos de las pruebas realizadas para evaluar la precisión y eficiencia de los sensores y del sistema de dispensación automatizado. Además, se calculará el tiempo necesario para que el dispensador arroje la cantidad correcta de alimento en base a la velocidad promedio de dispensación.

Sensor de peso: Las pruebas del sensor de peso del prototipo, demostraron que el sensor proporciona mediciones precisas y coherentes de peso, validando su capacidad para medir de manera confiable el peso del alimento dispensado.

Sensor de temperatura y humedad: las pruebas del sensor de temperatura y humedad verificaron que el sensor proporciona mediciones precisas y cercanas a las condiciones ambientales reales, lo que valida su utilidad en el monitoreo de condiciones ambientales.

Velocidad de dispensación: las pruebas realizadas con el objetivo de determinar la velocidad de dispensación del prototipo, mostraron que el sistema automatizado de dispensación es consistente y eficiente en la entrega del alimento. Los tiempos de dispensación se mantuvieron muy consistentes a lo largo de las pruebas, validando la eficacia del sistema.

Cálculo del tiempo de dispensación: para calcular cuántos segundos de activación del motor reductor se necesitan, para que el dispensador arroje la cantidad correcta de alimento, se utiliza la siguiente fórmula:

$$\text{Tiempo (s)} = \text{Cantidad de alimento deseada (g)} \div \text{Velocidad promedio (g/s)}$$

Cálculo de la cantidad de alimento a dispensar en base al estado actual de las tilapias: el pesaje de las tilapias se llevó a cabo una semana antes de la instalación del prototipo, obteniendo un peso promedio de 87 gramos (Ver Anexo 6), que de acuerdo al documento "Manual Crianza de Tilapia" de Nicovita, las ubicaría en un fase de crecimiento acorde a una semana 13, por lo que en la semana 14 (peso promedio de 100 gr) se recomienda que las tilapias sean alimentadas con una cantidad de alimento equivalente al 4% de su peso corporal.

Para calcular el alimento diario necesario para alimentar 100 tilapias con un peso promedio de 100 g, considerando una alimentación del 4% de su peso y dividida en 4 raciones diarias, se aplican las siguientes fórmulas:

$$\text{Alimento diario por tilapia (g)} = \text{Peso promedio por tilapia (g)} \times \text{Porcentaje de alimentación}$$

$$\text{Alimento total diario (g)} = \text{Alimento diario por tilapia (g)} \times \text{Número de tilapias}$$

Luego, para calcular el alimento necesario en cada ración diaria:

$$\text{Alimento por ración diario (g)} = \frac{\text{Alimento total diario (g)}}{\text{Número de raciones diarias}}$$

Dado que se dispensarán 4 raciones diarias, se debe dividir el alimento total diario entre 4, para obtener el alimento necesario en cada ración.

A continuación se realizan los cálculos:

1. Cálculo del alimento diario por tilapia:

$$\text{Alimento diario por tilapia (g)} = 100g \times 0.04 = 4g$$

2. Cálculo del alimento total diario:

$$\text{Alimento total diario (g)} = 4g \times 100 = 400g$$

3. Cálculo del alimento por ración diaria:

$$\text{Alimento por ración diaria (g)} = \frac{400g}{4} = 100g$$

En conclusión, cada una de las 100 tilapias requerirá una alimentación diaria de 4g, y esta cantidad se dividirá en 4 raciones diarias de 100 g cada una. Estos cálculos garantizarán que las tilapias reciban una alimentación adecuada para su crecimiento y salud.

Dado que la cantidad de alimento por ración es de 100 g, y la velocidad promedio de dispensación calculada fue de 45.163 g/s, el tiempo necesario para activación del motor del prototipo en las pruebas de funcionamiento fue de aproximadamente 2210 milisegundos.

CONCLUSIONES

Una vez culminado el presente trabajo, se puede manifestar que se logró alcanzar con el objetivo general del mismo, puesto que se implementó exitosamente un dispensador automatizado de alimento para tilapias con tecnología IoT, en el estanque para crianza de tilapias en la Quinta Novillo's.

Se logró modelar e implementar la parte mecánica del prototipo, consistente en una tolva metálica, con una capacidad aproximada para 3 Kg de alimento tipo concentrado para tilapias. La tolva en su base inferior se conecta a un tornillo sin fin motorizado, el mismo que también fue modelado y construido con una impresora 3D, marca Creality, Modelo Ender-3 S1 Pro

Se consiguieron en el mercado local el sensor de temperatura y humedad (DHT11), la celda de carga de 20 kg con su respectivo driver (HX711), el motor reductor de 12 V que permite el accionamiento del tornillo sin fin, con su respectivo driver (L298N); y desde luego el cerebro del circuito de control que corresponde a un NodeMCU-32.

Con la ayuda del software Proteus se procedió a diseñar y simular el sistema electrónico para el control automático del dispensador de alimento basado en ESP32. También se procedió al diseño de la tarjeta de circuito impreso que permitió montar adecuadamente el NodeMCU-32 y los demás componentes electrónicos en un contenedor independiente para el montaje de la electrónica del prototipo.

RECOMENDACIONES

El uso de sensores adicionales que permitan monitorear las condiciones físicas y químicas del agua, serían de gran ayuda para generar alertas tempranas, ante condiciones que pudieran afectar el desarrollo o vida de las tilapias dentro de los estanques de cría. Por lo que, incorporar estos sensores al diseño de un dispensador automatizado para tilapias, podría mejorar considerablemente su funcionalidad.

Sería muy importante, una vez que se haya validado satisfactoriamente la operación del prototipo, la incorporación de un equipo para regulación y respaldo de energía (UPS), para protección de la electrónica del mismo.

Si se desea continuar con mejoras para el prototipo implementado en el presente trabajo, se recomienda la incorporación de un sistema de generación fotovoltaico, con la finalidad de darle una autonomía energética al prototipo. Pudiendo alcanzar de esta forma, una adecuada utilización del prototipo en lugares donde no se cuente con suministro de energía eléctrica pública.

Se recomienda para futuros trabajos relacionados, la implementación de un dashboard con la plataforma Thingspeak, que según investigaciones realizadas, permitiría no solo monitorear variables físicas, como lo permite Grafana, sino que también se podrían realizar acciones de control, como las que se han utilizado en el software desarrollado con la herramienta .NET MAUI en el presente trabajo.

Es indispensable contar con una adecuada cobertura de red WIFI, para garantizar el normal funcionamiento del prototipo, y en caso tener una señal débil, se debería utilizar repetidores de señal, con la finalidad de alcanzar una potencia de transmisión óptima en el lugar de instalación del prototipo.

BIBLIOGRAFÍA

Aguayo Litardo, J. M., & Palma Macay, C. L. (2018). *Construcción de un dispensador de alimento automatizado para estanque de criadero de tilapias en el área de piscicultura ubicada en la finca experimental La María de la UTEQ.*

<http://repositorio.uteq.edu.ec/handle/43000/3769>

Baltazar Guerrero, P. (2004). *Manual del Cultivo de Tilapias.*

Castillo Badillo, B. B. (2022). *SISTEMA DE DETECCIÓN PREDICTIVA DE FALLAS DE UN MOTOR DE BAJA TENSIÓN MEDIANTE PROTOCOLO MQTT Y APLICACIÓN IOT.* [MasterThesis, Quito, Ecuador: Universidad Tecnológica Israel].

<http://repositorio.uisrael.edu.ec/handle/47000/3317>

Delgado Tapia, C. E., & Valencia Astudillo, W. G. (2021). *Diseño e implementación de prototipo IOT para el monitoreo remoto de la calidad del agua para la crianza de tilapias en estanques* [BachelorThesis]. <http://dspace.ups.edu.ec/handle/123456789/21427>

Enríquez Córdova, B. A., & Romero Ochoa, M. R. (2022). *Desarrollo de dispensador automático de alimento para tilapias utilizando esp32 cam.*

<http://repositorio.utmachala.edu.ec/handle/48000/19932>

Estadísticas—Cámara Nacional de Acuacultura. (2022, marzo 31).

<https://www.cna-ecuador.com/estadisticas/>

González Lugo, J. P., Montiel Arrieta, D. A., & Gómez Gómez, J. (2021). Desarrollo de un sistema de adquisición de parámetros ambientales en los estanques piscícolas basado en IoT. *Revista de Ingenierías Interfaces*, 4(2), 1-10.

Luzón, E. B. A., Alava, A. R. C., Carpio, E. P., & Montealegre, V. J. G. (2021). Gestión de agronegocios de la tilapia roja (*Oreochromis Spp. O*) y su comercialización. *Revista Metropolitana de Ciencias Aplicadas*, 4(2), Article 2.

Manual de Crianza de Tilapia Nicovita – DIGEPESCA | SAG HONDURAS. (s. f.).

Recuperado 23 de agosto de 2023, de

<https://www.digepesca.sag.gob.hn/2022/09/09/manual-de-crianza-de-tilapia-nicovita/>

Minta Toapanta, A. R. (2022). *APLICACIÓN IoT PARA EL MONITOREO DE CONSUMO ELÉCTRICO RESIDENCIAL UTILIZANDO SOFTWARE LIBRE*. [MasterThesis, Quito, Ecuador: Universidad Tecnológica Israel]. <http://repositorio.uisrael.edu.ec/handle/47000/3327>

Piamba, T., Zambrano, L. E., Montaña Rúaes, L. A., & Rojas Gonzales, F. A. (2021). Implementación de un sistema de monitoreo IoT aplicado a una piscicultura de trucha. *Informador técnico*, 85(1), 3-19.

Rivera Betancur, L. H. (2020). *Solución IoT para la optimización del proceso de piscicultura en el Centro de Desarrollo Agroalimentario El Limonal* [Master thesis]. <https://repository.usta.edu.co/handle/11634/23368>

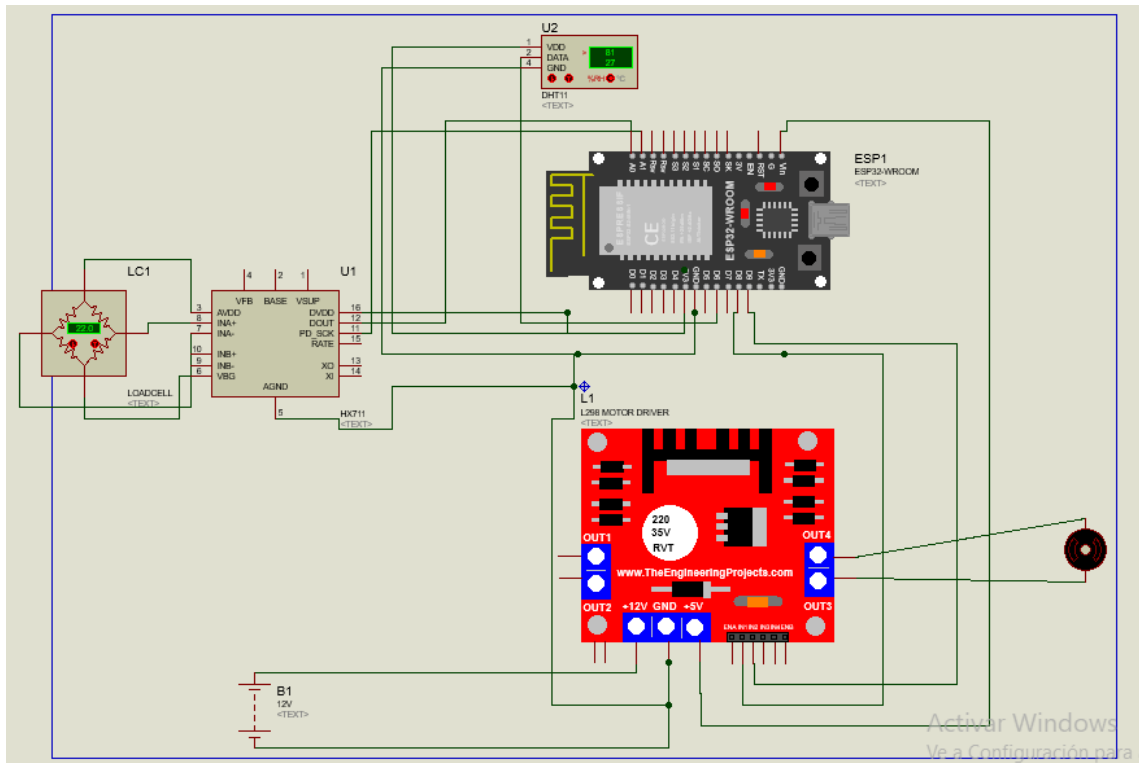
Rosaline, N., & Sathyalakshimi, S. (2019). IoT Based Aquaculture Monitoring and Control System. *Journal of Physics: Conference Series*, 1362, 012071. <https://doi.org/10.1088/1742-6596/1362/1/012071>

Sousa, R. M. R., Agostinho, C. A., Oliveira, F. A., Argentim, D., Novelli, P. K., & Agostinho, S. M. M. (2012). Productive performance of Nile tilapia (*Oreochromis niloticus*) fed at different frequencies and periods with automatic dispenser. *Arquivo Brasileiro de Medicina Veterinária e Zootecnia*, 64, 192-197. <https://doi.org/10.1590/S0102-09352012000100027>

ANEXOS

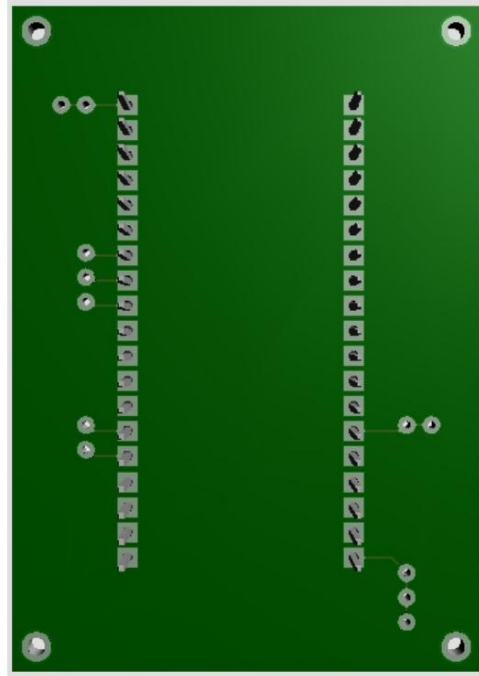
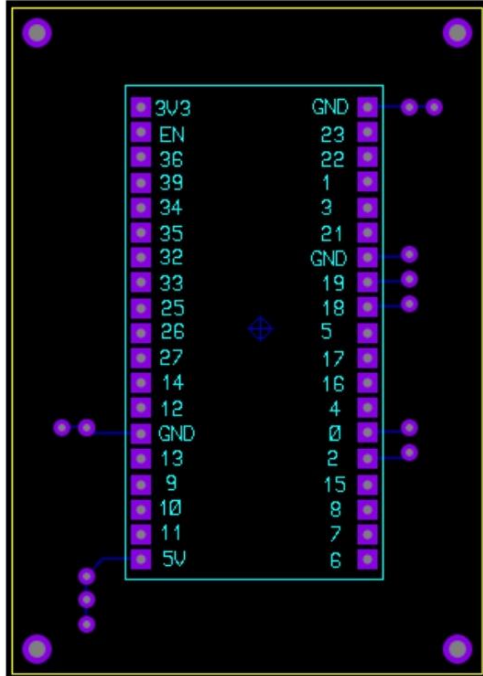
ANEXO 1

DIAGRAMA ELECTRÓNICO COMPLETO DE LA TARJETA DE CONTROL



ANEXO 2

DISEÑO DE LA TARJETA PARA CIRCUITO IMPRESO



ANEXO 3

CÓDIGO PARA PROGRAMACIÓN DE NODEMCU-32

```
#include <Arduino.h>
```

```
#include <WiFi.h>
```

```
#include <PubSubClient.h>
```

```
#include "HX711.h"
```

```
#include "DHT.h"
```

```
#include <EEPROM.h>
```

```
#include <TaskScheduler.h>
```

```
#include <WiFiUdp.h>
```

```
#include <TimeLib.h>
```

```
#include <NTPClient.h>
```

```
//PIN BALANZA
```

```
HX711 balanza;
```

```
//CONEXION MQTT PUBLICACION
```

```
WiFiClient espClient;
```

```
PubSubClient client(espClient);
```

```
//CONEXION MQTT ESCUCHA
```

```
WiFiClient espClient2;
```

```
PubSubClient client2(espClient2);

// PIN DE TEMPERATURA Y HUMEDAD

DHT dht;

//VARIABLES DE HORA

WiFiUDP ntpUDP;

NTPClient timeClient(ntpUDP, "pool.ntp.org");

//CONTROLADOR DE HILOS

Scheduler runner;

//LISTA DE HORARIOS

String horarios[10] = {

    "08:00",

    "12:00",

    "16:00"

};

int cantidadElementos = sizeof(horarios) / sizeof(horarios[0]);

bool ejecutado = false;

bool ejecucion_manual = false;

unsigned long tiempoInicioMotor = 0;
```

```
unsigned int duracionMotor = 3000;
```

```
void conectar_mqtt() {
```

```
Serial.println("Conectando a broker MQTT... (hilo de publicacion)");
```

```
if (client.connect("ESP32Client")) {
```

```
  } else {
```

```
    Serial.print("MQTT connection failed, rc=");
```

```
    Serial.println(client.state());
```

```
  }
```

```
}
```

```
void conectar_mqtt2() {
```

```
Serial.println("Conectando a broker MQTT... (hilo de escucha)");
```

```
if (client2.connect("ESP32ClientEscucha")) {
```

```
  client2.subscribe("tesisM/motor");
```

```
  client2.subscribe("tesisM/horarios");
```

```
  } else {
```

```
    Serial.print("MQTT connection failed, rc=");
```

```
    Serial.println(client2.state());
```

```
  }
```

```
}
```

```

bool CompararHoras(String horaActual){

for (int i = 0; i < cantidadElementos; i++) {

    if (horarios[i] == horaActual) {

        return true;

    }

}

return false;

}

```

```

void loopEscucha(){

if(!client2.connected()){

    conectar_mqtt2();

}

}

```

```

time_t currentEpochTime = timeClient.getEpochTime();

int currentHour = hour(currentEpochTime);

int currentMinute = minute(currentEpochTime);

//int currentDay = day(currentEpochTime);

//int currentMonth = month(currentEpochTime);

String horaActual = String(currentHour) + ":" + String(currentMinute);

if(CompararHoras(horaActual)){

```

```

if(!ejecutado){

    Serial.println("MOTOR ACTIVO-----");

    digitalWrite(2, HIGH);

    digitalWrite(0, LOW);

    ejecutado = true;

    tiempoInicioMotor = millis();

}

}else{

    ejecutado = false;

}

if (millis() - tiempoInicioMotor >= duracionMotor) {

    digitalWrite(2, LOW);

    digitalWrite(0, LOW);

    tiempoInicioMotor = 0;

    Serial.println("MOTOR INACTIVO-----");

}

}

void publicarDatos() {

```

```

conectar_mqtt();

if (client.connected()) {

    // Obtener peso

    float peso;

    peso = balanza.get_units(10);

    // HUMEDAD

    float h = dht.getHumidity();

    float t = dht.getTemperature();

    Serial.println("Humedad: " + String(h) + "% Temperatura: " + String(t) + "°C Peso:" +
String(peso));

    String datos = String(h) + ";" + String(t) + ";" + String(peso);

    client.publish("tesisM/datos_sensores", datos.c_str());

}

}

```

```

void escuchar_broker(char* topic, byte* payload, unsigned int length) {

    if (strcmp(topic, "tesisM/motor") == 0) {

        Serial.println("Mensaje recibido");

        String mensaje = "";

        for (int i = 0; i < length; i++) {

```

```

    mensaje += (char)payload[i];

}

Serial.println(mensaje);

if (mensaje == "ON") {

    digitalWrite(2, HIGH);

    digitalWrite(0, LOW);

} else {

    digitalWrite(2, LOW);

    digitalWrite(0, LOW);

}

}

if (strcmp(topic, "tesisM/horarios") == 0) {

    Serial.println("Mensaje recibido");

    String mensaje = "";

    for (int i = 0; i < length; i++) {

        mensaje += (char)payload[i];

    }

    char mensajeBuffer[length + 1];

    memcpy(mensajeBuffer, mensaje.c_str(), length);

    mensajeBuffer[length] = '\0';

```



```

char* lista = strtok(mensajeBuffer, ";");

int i = 0;

while (lista != nullptr && i < 10) {

    horarios[i++] = lista;

    lista = strtok(nullptr, ";");

}

}

}

void iniciar_conexion_mqtt() {

    delay(100);

    WiFi.begin("QUINTA NOVILLOS_CBVISION", "Novillos@2022@");

    Serial.print("Conectando a la red WiFi");

    while (WiFi.status() != WL_CONNECTED) {

        delay(1000);

        Serial.print(".");

    }

    Serial.println("\nHecho");

    client.setServer("mqtt-dashboard.com", 1883);

    client2.setServer("mqtt-dashboard.com", 1883);

    client2.setCallback(escuchar_broker);

```

```
}
```

```
void iniciar_balanza() {
```

```
    long escala = -249;
```

```
    bool calibrado = false;
```

```
    balanza.begin(19, 18);
```

```
    EEPROM.get(0, escala);
```

```
    if (escala != 0) {
```

```
        balanza.tare(20);
```

```
        balanza.set_scale(escala);
```

```
        calibrado = true;
```

```
    }
```

```
}
```

```
// Definir tareas
```

```
Task publicarDatosTask(1000, TASK_FOREVER, &publicarDatos);
```

```
Task conectarMqttTask(5000, TASK_FOREVER, &loopEscucha);
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    iniciar_balanza();
```

```
iniciar_conexion_mqtt();

dht.setup(27);

pinMode(2, OUTPUT);

pinMode(0, OUTPUT);

timeClient.setTimeOffset(-18000);

timeClient.begin();

// Agregar tareas al planificador

runner.addTask(publicarDatosTask);

runner.addTask(conectarMqttTask);

// Iniciar tareas

publicarDatosTask.enable();

conectarMqttTask.enable();

}

void loop() {

runner.execute();

client.loop();

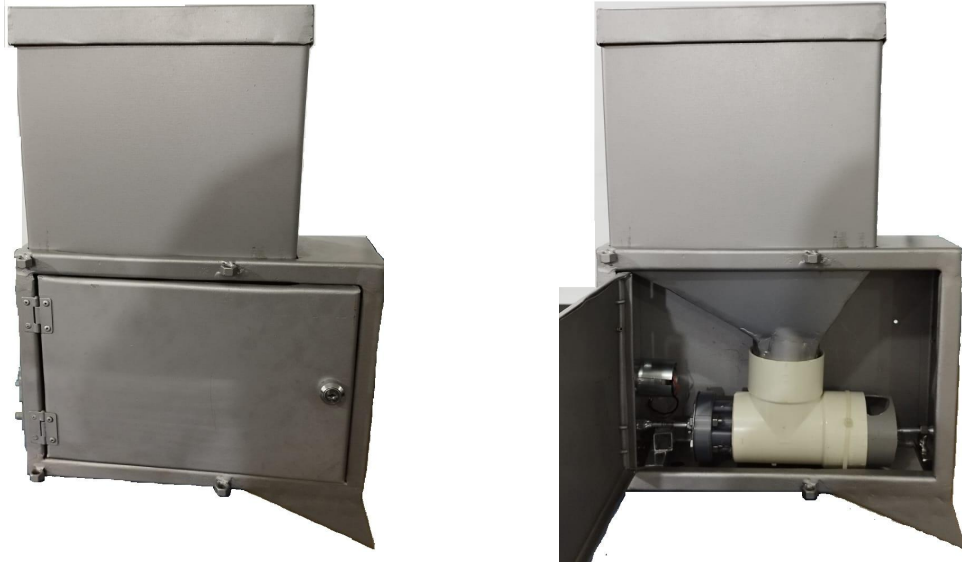
client2.loop();
```

```
timeClient.update();
```

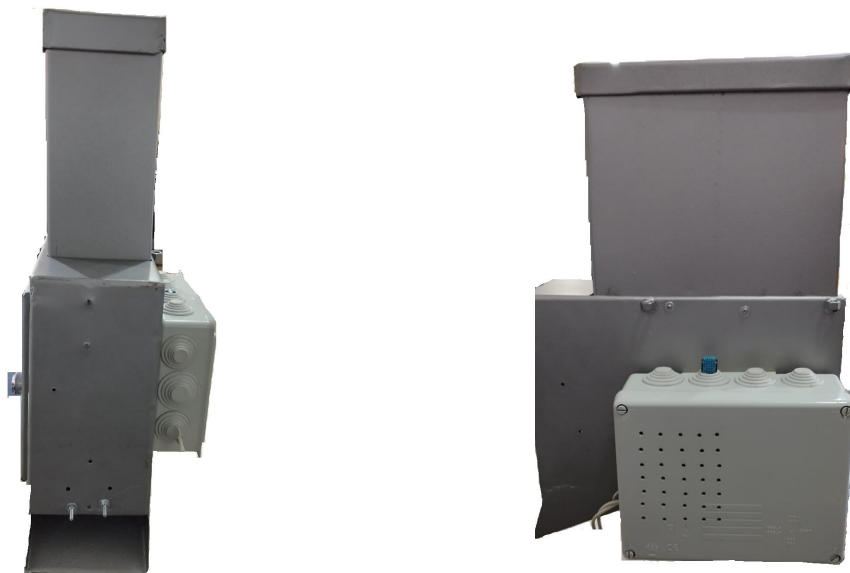
```
}
```

ANEXO 4

VISTAS DE PROTOTIPO IMPLEMENTADO



Vistas frontales



Vistas laterales

ANEXO 5

INTEGRACIÓN DE PROTOTIPO EN ESTANQUE DE TILAPIAS

Montaje de estructura para soporte y protección del prototipo



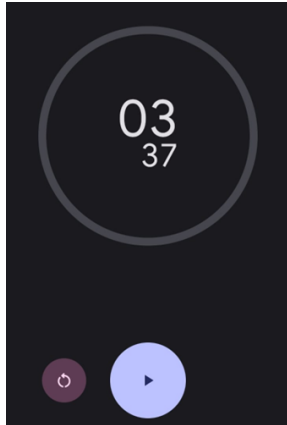
Montaje final del prototipo en estructura para soporte y protección



ANEXO 5

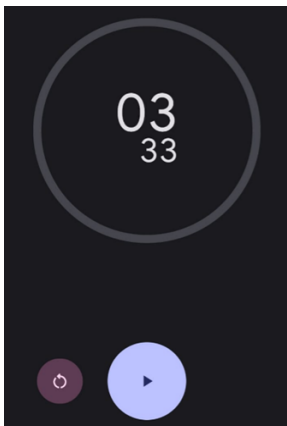
PRUEBAS DE VELOCIDAD DE DISPENSACIÓN DEL ALIMENTO

Prueba 1



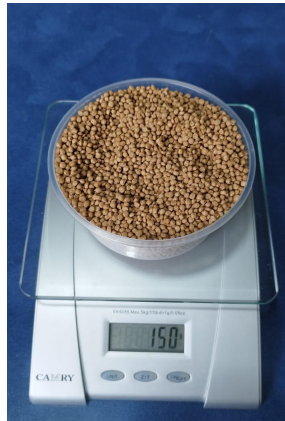
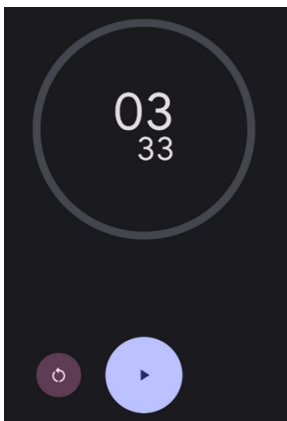
En la primera prueba para determinar la velocidad promedio de dispensación de alimento se obtuvo en un tiempo de 3.37 sg, una dispensación de 151 g de alimento concentrado para tilapias.

Prueba 2



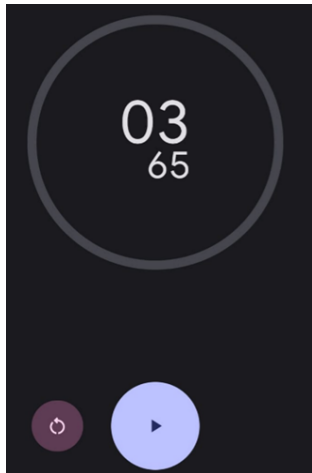
En la segunda prueba, en un tiempo de 3.33 sg, se obtuvo una dispensación de 150 g de alimento.

Prueba 3



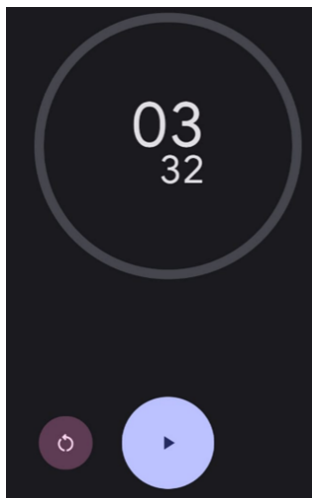
En la tercera prueba, en un tiempo de 3.33 sg, se dispensó 150 g de alimento.

Prueba 4



En la cuarta prueba, en un tiempo de 3.65 sg, se alcanzó una dispensación de 156 g de alimento.

Prueba 5



En la quinta y última prueba, en un tiempo de 3.32 sg, se tuvo una dispensación de 154 g de alimento concentrado para tilapias.

ANEXO 6

TABULACIÓN DE PESOS DE TILAPIAS EN QUINTA NOVILLO'S

Fecha:	19 de agosto de 2023					
Cantidad aproximada de tilapias en estanque:			100			
Datos de Tilapias en estanque de Quinta Novillo's			Datos de Tilapias en estanque de Quinta Novillo's			
Numero De Tilapia	Peso(KG)	Peso(g)	Numero De Tilapia	Peso(KG)	Peso(g)	
1	0,09	90	38	0,065	65	
2	0,115	115	39	0,05	50	
3	0,09	90	40	0,075	75	
4	0,105	105	41	0,09	90	
5	0,115	115	42	0,055	55	
6	0,12	120	43	0,05	50	
7	0,075	75	44	0,11	110	
8	0,08	80	45	0,105	105	
9	0,04	40	46	0,065	65	
10	0,12	120	47	0,11	110	
11	0,075	75	48	0,1	100	
13	0,1	100	49	0,11	110	
14	0,105	105	50	0,125	125	
15	0,045	45	51	0,135	135	
16	0,055	55	52	0,105	105	
17	0,045	45	53	0,08	80	
18	0,125	125	54	0,105	105	
19	0,065	65	55	0,045	45	
20	0,08	80	56	0,095	95	
21	0,14	140	57	0,095	95	
22	0,115	115	58	0,04	40	
23	0,1	100	59	0,035	35	
24	0,065	65	60	0,095	95	
25	0,135	135	61	0,065	65	
26	0,065	65	62	0,065	65	
27	0,115	115	63	0,09	90	
28	0,08	80	64	0,045	45	
29	0,09	90	65	0,045	45	
30	0,075	75	66	0,085	85	
31	0,065	65	67	0,075	75	
32	0,14	140	68	0,065	65	

12			
PROMEDIO:	87,09	MODA:	65
MAXIMO:	140	MEDIANA:	90
MINIMO:	35		

ANEXO 7

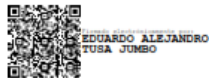
EVALUACIONES DE LOS ESPECIALISTAS



Yo, **Eduardo Alejandro Tusa Jumbo**, con C.I **0704323427**, en mi calidad de validador de la propuesta del proyecto titulado: **DISPENSADOR AUTOMATIZADO DE ALIMENTO PARA TILAPIAS CON TECNOLOGÍA IoT**.

Elaborado por el Ing. **Johnny Paúl Novillo Vicuña**, con C.I **0702947409**, estudiante de la Maestría en Electrónica y Automatización de la Universidad Tecnológica Israel (UISRAEL), como parte de los requisitos para obtener el Título de Magister, me permito declarar haber revisado el proyecto y realizado la evaluación de criterios.

Quito D.M., 31 de agosto de 2023



Eduardo Alejandro Tusa Jumbo

C.I 0704323427

Registro SENESCYT: 2501177762



Yo, **Iván Eduardo Ramírez Morales**, con C.I **0704065903**, en mi calidad de validador de la propuesta del proyecto titulado: **Dispensador automatizado de alimento para tilapias con tecnología IoT**.

Elaborado por el Ing. **Johnny Paúl Novillo Vicuña**, con C.I **0702947409**, estudiante de la Maestría en Electrónica y Automatización de la Universidad Tecnológica Israel (UISRAEL), como parte de los requisitos para obtener el Título de Magister, me permito declarar haber revisado el proyecto y realizado la evaluación de criterios.

Quito D.M., 31 de agosto de 2023



Iván Eduardo Ramírez Morales

CI: 0704065903

Registro SENESCYT : 7241123006



Yo, **Dixys Leonardo Hernández Rojas**, con C.I **0923026298**, en mi calidad de validador de la propuesta del proyecto titulado: **DISPENSADOR AUTOMATIZADO DE ALIMENTO PARA TILAPIAS CON TECNOLOGÍA IoT.**

Elaborado por el Ing. **Johnny Paúl Novillo Vicuña**, con C.I **0702947409**, estudiante de la Maestría en Electrónica y Automatización de la Universidad Tecnológica Israel (UISRAEL), como parte de los requisitos para obtener el Título de Magister, me permito declarar haber revisado el proyecto y realizado la evaluación de criterios.

Quito D.M., 31 de agosto de 2023



DIXYS LEONARDO
HERNANDEZ ROJAS

Dixys Leonardo Hernández Rojas

C.I 0923026298

Registro SENESCYT 7241138300

