UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO

TÍTULO

AUTOMATIZACIÓN DEL SISTEMA DE AIREACIÓN EN SATUKIN PISCINA 10 A TRAVÉS DE UN SISTEMA DE MONITOREO SCADA PARA LA CRIANZA Y ENGORDE DE CAMARÓN.

AUTOR

SANTOS SUÁREZ BRAULIO DAMIAN

TRABAJO DE TITULACIÓN

Previo a la obtención del grado académico en
MAGÍSTER EN ELECTRÓNICA Y AUTOMATIZACIÓN

TUTOR

Ing. RICARDO ALFREDO CAJO DÍAZ, Ph.D.

Santa Elena, Ecuador

Año 2024

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO

TRIBUNAL DE SUSTENTACIÓN

Firmado electrónicamente por:
ALICIA GERMANIA ANDRADE VERA

Ing. Alicia Andrade Vera, Mgtr.
COORDINADORA DEL PROGRAMA

Firmado electrónicamente por:
RICARDO ALFREDO CAJO DIAZ

Ing. Ricardo Cajo Díaz, Ph.D.
TUTOR

FRANCISCO VICENTE NOVILLO PARALES
Firmado digitalmente por FRANCISCO VICENTE NOVILLO PARALES
Fecha: 2024.09.09 12:02:42 -05'00'

Ing. Francisco Novillo Parrales, Ph.D.
DOCENTE ESPECIALISTA

Firmado electrónicamente por:
MANUEL ASDRUVAL MONTANO BLACIO

Ing. Manuel Montaño Blacio. MSc.
DOCENTE ESPECIALISTA

Firmado electrónicamente por:
MARIA MARGARITA RIVERA GONZALEZ

Abg. María Rivera, MSc.
SECRETARIA GENERAL
UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por BRAULIO DAMIAN SANTOS SUÁREZ, como requerimiento para la obtención del título de Magíster en Electrónica y Automatización.

TUTOR

Firmado electrónicamente por:
RICARDO ALFREDO CAJO DIAZ

Ing. Ricardo Cajo Díaz, Ph.D.

Santa Elena, 22 de junio de 2024

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO
DECLARACIÓN DE RESPONSABILIDAD

Yo, **Braulio Damian Santos Suárez**

DECLARO QUE:

El trabajo de Titulación, Automatización del sistema de aireación en Satukin piscina 10 a través de un sistema de monitore SCADA para la crianza y engorde de camarón previo a la obtención del título en Magister en electrónica y automatización, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Santa Elena, 22 de junio de 2024

EL AUTOR

Firmado electrónicamente por:
BRAULIO DAMIAN SANTOS SUAREZ

Braulio Damian Santos Suárez

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO
CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado Automatización del sistema de aireación en Satukin piscina 10 a través de un sistema SCADA para la crianza y engorde de camarón, presentado por el estudiante, Braulio Damian Santos Suárez, fue enviado al Sistema Antiplagio COMPILATIO, presentando un porcentaje de similitud correspondiente al 6%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.


CERTIFICADO DE ANÁLISIS
magister

BRAULIO SANTOS_TESIS_MAESTRIA

6% Textos sospechosos

5% Similitudes
0% similitudes entre comillas
< 1% entre las fuentes mencionadas
< 1% Idiomas no reconocidos

Nombre del documento: BRAULIO SANTOS_TESIS_MAESTRIA.docx
ID del documento: 5ffed8ed9ceb52c1bad0d45a15e8b034030a9f83
Tamaño del documento original: 9.41 MB

Depositante: RICARDO ALFREDO CAJO DÍAZ
Fecha de depósito: 22/6/2024
Tipo de carga: interface
fecha de fin de análisis: 22/6/2024

Número de palabras: 17.575
Número de caracteres: 115.752

TUTOR

Firmado electrónicamente por:
RICARDO ALFREDO CAJO DIAZ

Ing. Ricardo Cajo Díaz, Ph.D.

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO
AUTORIZACIÓN

Yo, **Braulio Damian Santos Suárez**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de mi Proyecto de titulación con componentes de investigación aplicada y/o de desarrollo con fines de difusión pública, además apruebo la reproducción de este trabajo académico dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, 22 de junio de 2024

EL AUTOR

Firmado electrónicamente por:
BRAULIO DAMIAN SANTOS SUAREZ

Braulio Damian Santos Suárez

AGRADECIMIENTO

Quisiera expresar mi más profundo agradecimiento a Dios y a mis padres por su amor brindado. A mis hermanos y novia les agradezco profundamente por su apoyo incondicional. Su aliento y compañía me han dado la fuerza necesaria para superar los obstáculos y seguir adelante.

Un sincero agradecimiento a mi tutor por su paciencia, dedicación y conocimiento que ha sido fundamental para el desarrollo y realización del proyecto.

Gracias a todos por creer en mí y por estar a mi lado en este camino. Este logro es tanto mío como de ustedes.

Braulio Damian, Santos Suárez

DEDICATORIA

A mis padres, con todo amor dedico este trabajo a ustedes. Sus sacrificios, consejos y enseñanzas han sido el cimiento sobre cual he construido mis sueños. Este logro es un reflejo de todo lo que me han dado. A ustedes, que siempre han creído en mí, incluso cuando yo mismo dudaba, este logro se los dedico con el más profundo agradecimiento y amor.

Con todo mi corazón

Braulio Damian, Santos Suárez

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente trabajo destaca la importancia de mantener parámetros óptimos para la cría y engorde del camarón, especialmente en lo que respecta al oxígeno disuelto en el agua. Se propone un sistema autónomo que monitoree continuamente el oxígeno en una piscina y ajuste la aireación mediante IA y tecnologías de IoT para mejorar la salud de los camarones y reducir costos. Se describen herramientas como Thonny, MQTT Explorer, InfluxDB y KepServer utilizadas para la adquisición, procesamiento y visualización de datos. Se detallan tecnologías como Modbus TCP/IP y protocolo MQTT para la transmisión de datos en sistemas de acuicultura. Se mencionan métodos de recolección de datos, diseño de investigación y sistemas de adquisición de datos como el sensor KNF-103b. Se discuten sistemas SCADA y aplicaciones de IA, incluyendo máquinas de vectores de soporte (SVM) y árboles de decisión. El estudio de caso se centra en la Granja Camaronera Satukin, destacando la implementación de un sistema SCADA para controlar niveles de oxígeno y la aplicación de IA para optimizar la aireación. Se detallan pruebas y resultados que demuestran la efectividad de aplicar IA en la gestión de la aireación, permitiendo una mejor optimización de recursos y control de motores.

Palabras claves: Oxígeno, IOT, SCADA

ABSTRACT

This work highlights the importance of maintaining optimal parameters for shrimp farming and fattening, especially with respect to dissolved oxygen in the water. An autonomous system that continuously monitors oxygen in a pool and adjusts aeration using AI and IoT technologies is proposed to improve shrimp health and reduce costs. Tools such as Thonny, MQTT Explorer, InfluxDB and KepServer used for data acquisition, processing and visualization are described. Technologies such as Modbus TCP/IP and MQTT protocol for data transmission in aquaculture systems are detailed. Data collection methods, research design and data acquisition systems such as the KNF-103b sensor are mentioned. SCADA systems and AI applications are discussed, including support vector machines (SVM) and decision trees. The case study focuses on the Satukin Shrimp Farm, highlighting the implementation of a SCADA system to monitor oxygen levels and the application of AI to optimize aeration. Tests and results are detailed that demonstrate the effectiveness of applying AI in aeration management, allowing for better resource optimization and motor control.

Keywords: Oxigen, IOT, SCADA

INTRODUCCIÓN

La importancia de los parámetros adecuados para la crianza y engorde del camarón, los técnicos acuicultores de producción regularmente tratan de tomar medidas adecuadas, las cuales contemplan mantener la salud del animal en buen estado, esta juega un papel muy importante para la supervivencia y crecimiento. Uno de los parámetros más importantes que se debe tomar en cuenta es el oxígeno, es decir, al encontrarse bajo el oxígeno disuelto en el agua existen muchas posibilidades de que el camarón se pueda morir y así alterar los precios de producción (López, 2021).

En sus inicios el oxígeno disuelto en el agua era regulado por el ingreso y salida de agua de los canales principales hacia las piscinas, esta acción generaba oxigenación en el agua, con el tiempo y expansión de la producción ahora con mayores densidades de organismos sembrados, se vio la necesidad de generar oxígeno disuelto por lo que se optó utilizar aireación mecánica (Lombeida Vásconez & Samaniego Reyes, 2022).

La empresa Satukin está conformada por 21 piscinas donde todas ellas cuentan con aireación mecánica, cada piscina cuenta con 1 aireador por hectárea. Los parámetros de oxígeno son tomados en horario nocturno donde por lo general el oxígeno baja, al no tener un control constante de este parámetro puede llegar a generar problemas como presentar bajones de oxígeno; llevando esto a ocasionar mortalidad por falta de oxígeno en los organismos (Carbajal Hernández & Sánchez Fernández, 2013).

La aireación mecánica es encendida en horarios establecidos de 4pm hasta las 6am, sin llevar un registro del nivel de oxígeno disuelto en el agua que tiene cada piscina, sin embargo, en ocasiones no es necesario tener encendido los aireadores con ello se genera gastos innecesarios.

Dado estos antecedentes, se plantea implementar un sistema autónomo, el cual nos ayudará a monitorear las lecturas de oxígeno de la piscina 10 durante las 24 horas del día, al tener los valores de este parámetro se plantea mediante un algoritmo matemático calcular el requerimiento necesario de encendido de aireadores y horarios que realmente sean necesarios, para así evitar el consumo innecesario de estos motores que permanecen encendidos por un horario establecido.

Dicha implementación, se realizará entre los meses de enero a junio de 2024 con una duración de 6 meses aproximadamente.

Una inadecuada generación de oxígeno en la piscina puede llevar a generar riesgos, como obtener un parámetro del nivel de oxígeno inadecuado por la cantidad de organismos sembrados en la piscina. Por lo cual, mediante cálculos matemáticos se plantea presentar en el sistema la cantidad necesaria de motores que se debería encender por bajas de oxígeno significativas, de la misma forma cuando el oxígeno este en su umbral adecuado mantener los aireadores inactivos de esta forma se reducirían gastos.

Al tener ya horarios establecidos para la activación de los motores de aireación se ha presentado problemas de bajones de oxígeno fuera del horario que estos permanecen encendidos. Por ello, se presenta llevar un registro y monitoreo de los datos, en el cual se implementará adaptaciones tecnológicas de Inteligencia Artificial (IA), donde se analizará y entrenará la data para predicciones para la activación de los motores, esto mediante un análisis de entrenamiento y así evitar que el nivel del oxígeno en la piscina sea menor a los estándares establecidos.

La implementación del proyecto proporciona garantizar de manera segura y confiable el funcionamiento del sistema Scada, en el cual intervienen tecnologías como: la Inteligencia Artificial (IA), Internet de las cosas (IOT) para el análisis y enfoque del funcionamiento del sistema; al implementar este tipo de tecnología nos permite garantizar un sistema predictivo, el cual nos ayudará a la toma de decisiones y así tener un sistema que nos pueda alertar algún tipo de anomalías al estar recibiendo los datos en tiempo real.

La lectura del Oxígeno disuelto en el agua se realizará a través de un sensor Knf-103b, donde la información será procesada por una ESP32 por el protocolo de comunicación Modbus R485. Para la transmisión de los datos procederemos con el protocolo MQTT para enlazar dicha información con las plataformas de monitoreo y almacenamiento de datos. Con ello poder presentar los datos que se receptaran por medio de un Scada local y la nube.

Planteamiento de la investigación (Fundamentación de la investigación)

Con las constantes mejoras que se vienen realizando en la automatización de la producción del camarón, se emplea garantizar con este tipo de tecnología poder mantener un monitoreo constante; el cual nos ayude a prevenir riesgos de mortalidad de los organismos sembrados en las piscinas.

La implementación de este tipo de tecnologías como: la IA y IOT; se vuelve crucial para garantizar el monitoreo y prevención de riesgo en la producción de camarón, también nos ayuda a analizar la data en tiempo real, predecir situaciones adversas y la toma de decisiones para mantener un entorno óptimo para los organismos sembrados; con ello se asegura no solo la eficiencia sino también la salud y productividad del cultivo de camarón a largo plazo. En los campos acuícolas de producción de camarón se ha implementado la aireación eléctrica y mecánica, sin mantener un monitoreo del nivel de oxígeno de las piscinas, al no identificar los niveles adecuados de oxígeno, se puede dar paso a perdidas de producción por falta de oxígeno disuelto en el agua (Carchipulla Leal, 2018).

La baja de Oxígeno Disuelto en el agua (OD), genera un inconveniente relevante para los crecimientos y supervivencia adecuada, lo que nos da un indicio para poder implementar un sistema automatizado incorporando a la IA, esta nos ayudará a evaluar patrones y comportamientos del parámetro, y con ello poder mantener umbrales adecuados para mejores resultados en la producción. Un ejemplo de ello es cuando los parámetros básicos se encuentran en los rangos adecuados, la producción del camarón tiene supervivencias superiores al 90% (Arcia, s. f.).

Formulación del problema de investigación

¿Cuál es la contribución de las tecnologías emergentes, como la IA y Scada, en la producción de camarón para garantizar un entorno adecuado y eficaz?

OBJETIVOS

Objetivo General:

Diseñar un sistema Scada que transmita en tiempo real parámetro de oxígeno a través del sensor de Oxígeno disuelto KNF-103B, para la automatización y eficiencia energética en los aireadores de la piscina 10.

Objetivos Específicos:

1. Implementar un sistema de lectura de oxígeno disuelto, que será monitoreado permanentemente (mg/L), registrando dicha información en una de base de datos local y nube.

2. Diseñar la red de comunicación de los equipos electrónicos para la integración de compatibilidad entre sensor, controladores y actuadores.

3. Desarrollar un modelo matemático que determine el rendimiento energético de los aireadores el cual permita realizar comparaciones de consumos eléctricos y funcionamiento con respecto a otras piscinas no automatizadas.

4. Implementar un sistema Scada para supervisar y controlar de manera eficiente los niveles de oxígeno.

Planteamiento hipotético

¿El sistema autónomo basado en inteligencia artificial, permitirá la detección de bajas de oxígeno y toma de decisiones de mejora para la piscina 10 de Satukin?

CAPÍTULO 1. MARCO TEÓRICO REFERENCIAL

1.1. Revisión de literatura

El artículo presentado por (Z Dong and Y Wang, 2023), demuestra que tras varias investigaciones el consumo eléctrico en los terrenos acuícolas sobrepasa aproximadamente 40% veces más que el consumo generado por la aireación mecánica, debido a su gran consumo hecho por la aireación eléctrica por ello se restringe el desarrollo. A través de la investigación, se encontró que el consumo de energía de la carga del aireador representaba la mayor proporción del consumo de energía del sistema, la proporción del tiempo de uso era la más alta y había un gran espacio para el ahorro de energía. Las cargas se funcionan a una potencia constante, el cual representa un 38.5% del consumo total del sistema, estos mismos manejados a través de válvulas. Por lo tanto, el método basado en PID difuso se utiliza para controlar el motor interno del aireador con voltaje variable, frecuencia variable y regulación de velocidad. Al analizar diferentes condiciones de trabajo, el motor se controla razonablemente permitiendo calcular el consumo de energía del aireador en cada momento; los resultados muestran que el uso de la regulación de velocidad por conversión de frecuencia puede reducir efectivamente el consumo de energía y mejorar los beneficios económicos del consumo de energía en las fábricas.

(Sierra & Maroso, 2019), en sus investigaciones resaltan lo importante de controlar los parámetros de un sistema de cultivo. El oxígeno disuelto en el agua es un parámetro de vital importancia en un ecosistema acuícola, siendo la variable más crítica para la calidad del agua de un estanque, sin una buena provisión de oxígeno los organismos pueden ser vulnerables a enfermedades, o morir. Los acuicultores deben conocer bien los factores que afectan la concentración de oxígeno disuelto; y cómo influye una baja concentración de este parámetro en los camarones. Para un manejo adecuado de la producción del camarón debe estar entre los 4-5 mg/L.

En el artículo publicado por (Bajaña Araujo & Junqui Sudario, 2018), hace referencia a la implementación de un dispositivo, el cual plantea llevar el control de varios parámetros importantes para mantener los umbrales adecuados para la crianza de camarones desde un peso de (0.008 a 13) gramos, a través de una placa raspberry con tecnología ZigBee;

junto con otros sensores conectados registrando valores para una bitácora para mantener reportes sobre los parámetros y estados de los umbrales. El sistema busca la mejora y eficiencia en el protocolo de la crianza de camarón.

El artículo presentado por (Delgado Tapia & Valencia Astudillo, 2021), presentan la tecnificación para el crecimiento de las tilapias mediante un monitoreo constante a través de Iot, utilizando sensores y hardware como el ESP8266 y Raspberry Pi, la cual permite el monitoreo constante de parámetros ambientales y la calidad del agua para mantener un control adecuado del crecimiento de tilapia, estos datos serán guardados en el servidor local de la Raspberry y así también serán enviados a un servidor web. La contribución de este prototipo IoT, se realiza utilizando hardware de bajo costo y el software ThingSpeak con una licencia gratuita de pruebas. Es importante destacar que se requiere una conexión a internet estable en el sitio donde se encuentra la piscina con los alevines para el funcionamiento adecuado del Sistema.

El artículo de (Lombeida Vásconez & Samaniego Reyes, 2022), se centra en la implementación de un tablero eléctrico demostrativo; el cual simula un sistema de aireación y alimentación para industrias camaroneras. Este sistema es supervisado y controlado mediante un sistema SCADA con un enlace de comunicación por radiofrecuencia. El objetivo es crear un sistema eficiente que permita convertir los accionamientos de los motores de forma manual a remota; lo que facilita el monitoreo desde un cuarto de control principal sin necesidad de dirigirse a cada piscina. Se pueden controlar motores simulados de alimentación y aireación, ajustando la velocidad del motor de alimentación para controlar el radio de expansión del alimento.

1.2. Desarrollo teórico y conceptual

1.2.1 Parámetros de calidad del agua para el cultivo de camarón

La calidad del agua es fundamental en la acuacultura para el desarrollo del sistema de producción, ya que influye en la salud y el crecimiento de los organismos cultivados. Problemas como la contaminación, los cambios en la temperatura, el pH y la concentración de oxígeno pueden afectar negativamente a los cultivos acuáticos. La

sobrealimentación y la acumulación de desechos también pueden deteriorar la calidad del agua. Por ello, los acuicultores emplean diversas estrategias para mantener la calidad del agua en niveles óptimos, como la filtración, la aireación y el monitoreo regular de parámetros clave (Carbajal Hernández & Sánchez Fernández, 2013).

La calidad del agua se pierde debido a la administración inadecuada de las materias primas utilizadas para producir larvas de camarón o debido al flujo natural del agua, la limpieza de los tanques receptores o el diseño inadecuado de aireación del estanque, nutrición larvaria inadecuada u otros factores que se presentan durante la producción (Chuya Zhangallimbay & Landívar Zambrano, 2021).

1.2.1.1 Oxígeno

La concentración mínima de oxígeno requerida para mantener una supervivencia aceptable variará según la duración de la exposición. El florecimiento algal, el aumento de temperatura y la sobrealimentación que puede aumentar el contenido de materia orgánica en el agua, son unos de los factores que llevan a obtener concentraciones bajas de oxígeno disuelto (*Folleto del agua - trabajo de investgacion - Folleto Informativo Oxígeno Disuelto (OD) ¿Qué es el - Studocu*, s. f.).

Bajo la influencia de la luz solar, las algas y el plancton desarrollan la fotosíntesis y esta acción aumenta la concentración de oxígeno disuelto en el agua. Sin embargo, por la noche, en días lluviosos o nublados, es posible que el agua no proporcione suficiente oxígeno a los camarones (Chuya Zhangallimbay & Landívar Zambrano, 2021).

Cuando los niveles de oxígeno disuelto son bajos, los camarones son más susceptibles a las enfermedades, debido a que se trasladan hacia la superficie, lo que puede producir un estado letárgico generando la muerte en organismos adultos y juveniles. Sin embargo, cuando el agua esta sobresaturada de oxígeno, el aire entra al sistema circulatorio del camarón, generando el inicio de la enfermedad denominada "burbuja de gas" provocando la muerte del ejemplar. Cuando la tasa de saturación se mantiene por debajo de 5 mg/l, los camarones tienen bajas tasas de alimentación y crecimiento. Los cambios en el oxígeno disuelto en los estanques durante el día parecen tener poco efecto sobre la

nutrición y el crecimiento de algunas especies hasta que las concentraciones de oxígeno disuelto caen por debajo de 2 mg/l por la mañana. El valor mínimo recomendado por los expertos es de 4 a 5 mg/l (Carbajal Hernández & Sánchez Fernández, 2013).

Tabla 1

Parámetros de calidad de agua en cultivos de camarón (Oxígeno)

Valor de oxígeno disuelto	Efecto sobre los camarones
0.3 mg/l	Camarones mueren
2.0 mg/l	Camarones con ritmo de crecimiento bajo
4.0 mg/l	Crecimiento normal
>7.0 mg/l	Riesgo de enfermedad “burbuja de agua”

1.2.1.2 Temperatura

La temperatura es un parámetro controlador de los diferentes procesos químicos y biológicos en la piscina hasta tal punto que un aumento de 10°C en la temperatura provocará un aumento de dos a tres veces en la velocidad de las reacciones químicas y biológicas. La temperatura controla la solubilidad de los gases en el agua, la velocidad de las reacciones químicas y la toxicidad del amonio. Se considera que el rango óptimo de temperatura se sitúa entre los 25 y 32°C (Carbajal Hernández & Sánchez Fernández, 2013).

En el caso del cultivo de camarón, el valor óptimo permite un buen desarrollo durante el ciclo de crecimiento. La constancia en el mantenimiento de las condiciones ideales de temperatura suele estar relacionada con las estaciones (invierno y verano). La temperatura es una de las variables más importantes ya que impide un crecimiento rápido dentro del rango óptimo de la especie, lo que permite reducir los tiempos de producción y mejorar las operaciones de alimentación mientras se mantiene el equilibrio (Chuya Zhangallimbay & Landívar Zambrano, 2021).

Tabla 2

Parámetros de calidad de agua en cultivos de camarón (temperatura)

Valores de temperatura	Efecto sobre los camarones
0 – 25	Bajo crecimiento
25 – 32	Crecimiento normal
> 32	Crecimientos altos y poco propenso a enfermedades

1.2.1.3 Salinidad

Los organismos que viven en el agua deben mantener una determinada concentración de iones (salinidad) en sus fluidos vitales para poder funcionar correctamente. A medida que la salinidad ambiental fluctúa, los organismos deben implementar estrategias específicas para mantenerla en niveles favorables, incluido el gasto de energía que puede conducir a tasas de crecimiento más bajas. Las altas concentraciones de salinidad reducen la cantidad de oxígeno disuelto en el agua del estanque. Las concentraciones óptimas de sal oscilan entre 15 y 23 mg/l (Carbajal Hernández & Sánchez Fernández, 2013).

La salinidad en los estanques de cultivo de camarones puede variar. Puede aumentar con la evaporación o disminuir con la lluvia. Sin embargo, aunque los camarones pueden vivir en aguas de diferente salinidad, no pueden soportar cambios repentinos de salinidad de 0 a 50 ppm. El proceso de adaptación del camarón al nuevo nivel de salinidad debe realizarse lentamente (Chuya Zhangallimbay & Landívar Zambrano, 2021).

Tabla 3

Parámetros de calidad de agua en cultivos de camarón (Salinidad)

Valor de pH	Efectos sobre los camarones
Menos de pH 5	Los camarones pueden morir
pH 7.5 a pH 9.0	pH optimo

1.2.1.4 pH

Cuando el pH es bajo o alto, causa estrés a los organismos cultivados, así como el incremento de sensibilidad para contraer enfermedades, lo cual conlleva a reducir los niveles de producción y ocasionar un desarrollo defectuoso ocasionando inclusive la muerte. El agua con un pH entre 6,5 y 9,0 es la mejor para la acuicultura. La reproducción disminuye en valores inferiores a 6,5 o superiores a 9,0. Por debajo de 4,0 se produce la muerte ácida y por encima de 11,0 se produce la muerte alcalina. Aunque los organismos pueden sobrevivir, la productividad de los estanques con valores de pH matutinos de 4 a 6 o de 9 a 10 es baja. En muchos sistemas de cultivo se puede elevar el pH hasta un valor de 9 o 10 durante cortos periodos de tiempo sin provocar efectos nocivos para el organismo. Los niveles de pH extremadamente bajos y altos pueden provocar la formación de una cáscara dura y una mala supervivencia (Carbajal Hernández & Sánchez Fernández, 2013).

Por lo general, en el cultivo de camarón se observan valores bajos de pH durante la noche debido a la respiración y la producción de CO_2 por parte de los organismos que viven en el estanque, mientras que durante el día el pH aumenta, estas fluctuaciones no deben exceder 0,5. Mantener un rango estable es esencial porque las variaciones pueden afectar el metabolismo del camarón y otros procesos fisiológicos (Chuya Zhangallimbay & Landívar Zambrano, 2021).

1.2.2 Sistema de aireación

1.2.2.1 Mecánica

Figura 1

Sistema de aireación

La aireación mecánica es un factor importante en el consumo de energía en el cultivo de camarón. Alrededor de 4 millones de toneladas métricas (TM) de los aproximadamente 5 millones de toneladas métricas (TM) de camarones peneidos cultivados en granjas provienen de estanques con aireados mecánicos. La cantidad de aireación utilizada en los estanques de camarón no se ha investigado exhaustivamente, pero los administradores de las granjas camaroneras suelen utilizar una "regla general" de que por cada aumento en la producción de camarón de 300 a 500 kg se requiere 1 hp de aireación (de 3,33 a 2,0 CV/TM de camarón). Sin aireación, se pueden cultivar hasta 1,5 toneladas de camarones por hectárea, pero esta cantidad generalmente no se resta del rendimiento objetivo al calcular el factor de aireación para calcular el factor de seguridad (Chuya Zhangallimbay & Landívar Zambrano, 2021).

Con la referencia del cálculo que se tiene para la generación de OD, procedemos a transformar los 400kg como medida promedia de camarón a libras, para poder realizar el cálculo necesario de los Hp que necesarios para la generación de oxígeno.

Ecuación (1)

Cálculos de hp necesario para generación de OD por libras estimadas

$$x = \frac{(1hp) * (libras\ estimadas\ )}{881lb} \qquad (1)$$

Teniendo las libras estimadas que se tienen en la piscina podemos obtener el valor aproximado de los hp que se necesitará, con ello cabe recalcar que la generación de OD puede verse alterado con cambios de los parámetros ya antes expuesto. Por lo cual, no se tiene un cálculo exacto de generación de oxígeno que los motores pueden generar.

Para obtener cálculos de consumo de un motor mecánico a diésel se requiere de sus especificaciones en la cual se requiere saber el consumo especifico por hora caballo de fuerza.

Ecuación (2)

Cálculo del consumo de diésel por tiempo de operación

$$consumo\ total(gal) = (consumo\ especifico\ gph) * (tiempo\ de\ operacion) \qquad (2)$$

1.2.2.2 Eléctrica

Figura 2

Sistema de aireación eléctrica

Los motores eléctricos pequeños consumen alrededor de 1 kWh (un kilovatio hora es una unidad de energía equivalente a 3,6 megajulios) de electricidad por caballo de fuerza-hora de funcionamiento. La cantidad de energía consumida por tonelada métrica de camarón a una tasa de aireación de 2,5 hp/TM de camarón usando un aireador para operar el estanque durante un promedio de 16 horas en el día durante el período de crecimiento de 80 días es bastante alta. La cantidad de electricidad (y energía) por hectárea seguirá aumentando a medida que aumente el objetivo de producción en toneladas métricas por hectárea, pero la cantidad de energía utilizada por tonelada métrica de camarón para la aireación es de 11,5 GJ/TM (2,5 kW/TM × 16 horas/día × 80 días/cosecha × 0,0036 GJ / kWh = gigajulio, una unidad de medida de energía) y es constante en todos los niveles de producción (Chuya Zhangallimbay & Landívar Zambrano, 2021).

Para el cálculo del consumo eléctrico necesitamos tener las especificaciones técnicas del motor como:

- Potencia del motor: (HP)
- Eficiencia del motor: ($\eta\%$)
- Factor de potencia: (PF)
- Horas de operación: (h)

A continuación, procedemos a convertir de HP a kW:

Ecuación (3)

Potencia de HP a Kw

$$potencia\ (kw) = HP * 0.746 \qquad (3)$$

Luego debemos considerar la eficiencia y el factor de potencia del motor:

Ecuación (4)

Potencia real

$$potencia\ real(kw) = \frac{potencia\ (kw)}{eficiencia * factor\ de\ potencia} \quad (4)$$

Con ello llegamos al cálculo de consumo energético por hora:

Ecuación (5)

Cálculo del consumo energético por tiempo de operación

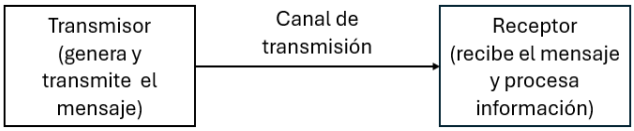
$$consumo\ de\ energia\ (kWh) = potencia\ real\ (kw) * horas\ de\ operacion\ (h) \quad (5)$$

1.2.3 Sistemas de comunicación y compatibilidad de sensores

Un sistema de comunicación se denomina a un conjunto de dispositivo el cual nos permitirá la transmisión y recepción de datos ya sean estos analógicos o digitales. Por lo general, está constituido de un elemento el cual se encarga de realizar las lecturas de datos ya sea un microcontrolador o un controlador lógico programable, estos se encargan del procesamiento de la información para luego ser enviados a través de un transmisor.

Figura 3

Sistema de comunicación



La transmisión de datos puede darse de forma alámbrica o inalámbrica, esto se adecuará según la data que se quiera procesar y las condiciones adecuadas donde se implementará el sistema. Para la interpretación de datos, procesamiento y recepción de este se debe tomar en cuenta los protocolos de comunicación de cada equipo con el que se estará realizando la comunicación; se debe tener en cuenta que, al tratar de gestionar la lectura

de los datos, se debe conocer si son compatibles sensores, actuadores y microcontroladores para el tipo de estructura que se va a implementar.

1.2.3.1 Protocolo de comunicación Modbus TCP/IP

El protocolo Modbus se comenzó utilizando en fabricación industrial, en la actualidad la mayoría de las empresas cuentan también con este tipo de servicio en telecomunicaciones, agricultura, seguridad e industrias dedicadas a series de procesos de producción. Los sistemas que se utilizan en este protocolo son habilitantes para RTU o TCP los mismos que nos permitirán un monitoreo constante de los dispositivos conectados al sistema de comunicación que se implementarán.

La definición de TCP/IP es el Protocolo de control de trasmisión/Protocolo de internet el cual nos servirá para la intercomunicación de los dispositivos que estarán conectadas a nuestra red, este protocolo se lo interpreta como la capa de conmutación y enrutamiento entre las aplicaciones de internet, donde se produce la trasmisión de datos de un punto a otro, aquí se produce la división de paquetes, direccionamiento y el enrutamiento para la recepción de datos. Este tipo de redes es muy confiable y tienen la capacidad de recuperar de manera inmediata ante cualquier falla de algún dispositivo que está conectada a la red (*AIX 7.1*, 2023).

Beneficios de TCP/IP:

Este protocolo es confiable al garantizar de manera ordenada los paquetes, al ser un protocolo de bajo nivel facilita la comunicación a través de internet. Al contener un encabezado IP que contiene la información por donde debe enrutarse da origen para que la transmisión de datos sea más rápida y eficaz; también consta de una puerta de enlace que ayuda la verificación de la red conectada.

A su vez, TCP se encarga que los datos antes de ser enviados se ensamblen en paquetes más pequeños para su envió y luego al recibirlo se encarga de ordenarlos para la recepción del destinatario, TCP trabaja en la capa 4, capa de transporte para la intercomunicación de los sistemas OSI, tanto que IP en la capa 3 que es por donde se accede la red, por lo cual se podría definir que este protocolo se encargará del transporte y acceso a la red de la información que se generará(*TCP (Transmission Control Protocol)*, 2020).

1.2.3.2 Recepción de datos Modbus TCP/IP

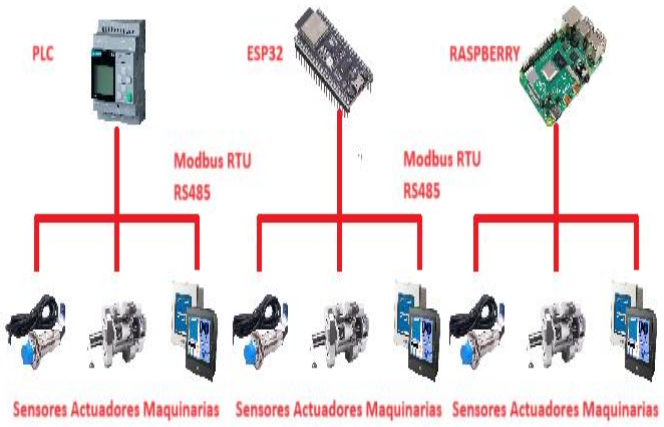
El protocolo Modbus TCP/IP mejor conocido como el protocolo universal de comunicación, se ha incorporado a la mayoría de los sensores que están ligados a la producción en industrias uno de sus mayores beneficios es poder majar información desde una red a internet de bajo nivel y es muy confiable, estos sensores ofrecen compatibilidad con equipos como PLC, microcontroladores o software que tienen este tipo de protocolos(*Modbus*, 2020).

Estos sensores que tienen protocolos Modbus TCP/IP permiten que múltiples controladores puedan realizar consultas de información a cada Slave, los mismos podrían estar conectados en la misma línea de transmisión.

1.2.3.3 Modbus RTU

Figura 4

Topología Modbus RTU



El protocolo RTU es una arquitectura maestra/esclavo, la estructura de sus mensajes de 16 bits se puede utilizar para empaquetar textos ASCII, tablas, colas y otros datos relacionados, este protocolo de comunicación utiliza interfaces seriales RS-485 o RS-432 ya que estos son compatibles con la mayoría software; sirven para monitoreos y adquisiciones de datos(*Modbus*, 2020).

1.2.3.3.1 Mapa de memoria y codificación de datos Modbus RTU

El mecanismo de codificación debe ser el mismo tanto del transmisor como del receptor para la comprensión del contenido de los datos, estos pueden ser modbus: ASCII y RTU. La codificación RTU da paso a la codificación binaria big-endemian el cual da el valor de 16 bits del valor más significativo hasta el menos significativo.

El tipo de memoria Modbus RTU es importante para permitir una comunicación estandarizada entre dispositivos, donde se establecen las direcciones de la memoria de la siguiente forma según su funcionamiento de los datos:

Tabla 4

Direcciones de memoria protocolo RTU

Tipo de datos Modbus RTU	Nombre	Dirección
Bobina	Valores binarios, Bits, banderas	00001 – 10000
Entrada Digital	Entradas binarias	10001 – 20000
Entrada analógica	Entradas binarias	30001 – 40000
Registro de entrada	Valores analógicos	40001- 50000

1.2.4 Protocolo MQTT

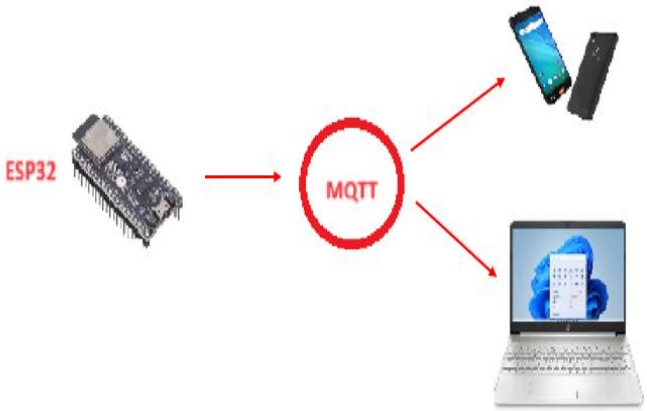
MQTT (Message Queuing Telemetry Transport) por sus siglas en inglés, es un protocolo diseñado para mensajería ligero y redes de ancho de banda restringido, el mismo fue desarrollado para comunicación Internet de las cosas (IOT) y de maquina a máquina (M2M). El protocolo está compuesto por un modelo de publicación/suscripción donde un dispositivo envía mensajes, el suscritor se encarga de recibir el mensaje, además de ello también de un Broker que es el servidor que administra la red del mensaje. MQTT utiliza Topics que se encarga de crear cadenas jerárquicas de la ruta del directorio, los suscritores

pueden adquirir información de varios Topics para la recepción de varios mensajes(comercial, 2020).

Algunas de las ventajas de MQTT es que se puede manejar una gran cantidad de Topics, alta fiabilidad para la entrega de mensajes, bajo consumo de energía y fácil implementación al tener una estructura clara. Este tipo de protocolo es instalado en sistemas de control, vigilancia y monitoreo, también en dispositivos domésticos inteligente y sensores.

Figura 5

Estructura física MQTT



1.2.5 Aplicaciones IOT

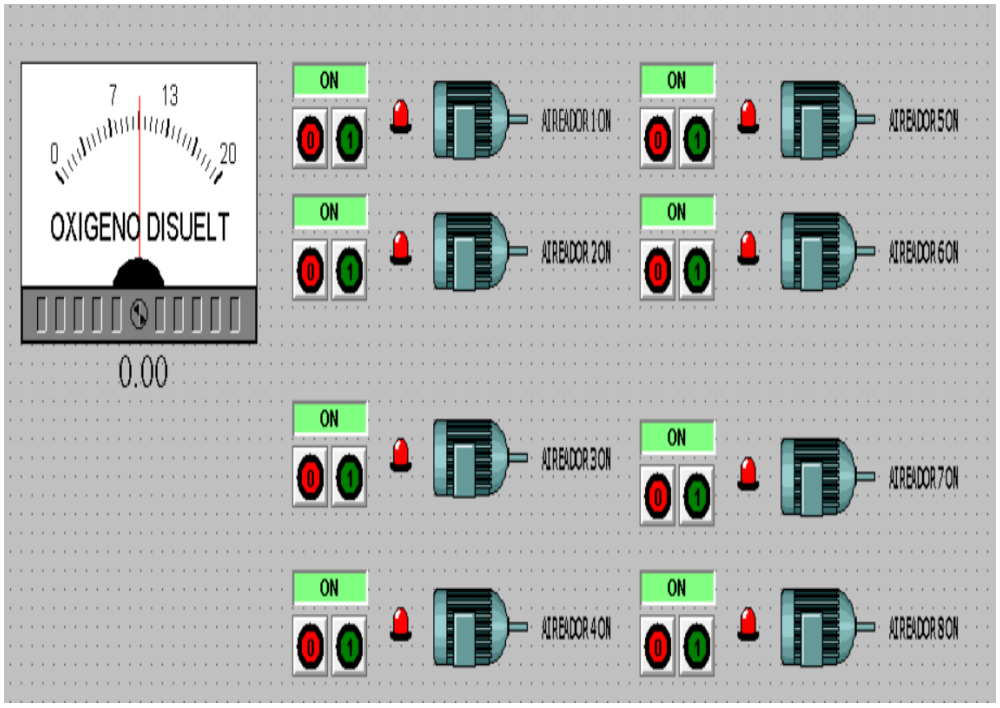
En la industria, IoT se considera como al conjunto de maquinarias, sensores y redes de dispositivos conectados entre sí a una red de internet para la recolección de datos. Con el fin, de analizar patrones de comportamientos en los procesos industriales; todo este tipo de aplicaciones que se le puede asignar han mejorado e impulsado mejoras en la producción y reducción de gastos.

Las principales aplicaciones que se pueden realizar es el manejo, gestión de manera remota y monitoreo automatizada; las cuales están conectadas a un sistema central, la cual implica que a través de software se pueda controlar procesos de varias plantas que estén ubicadas en diferentes puntos. En la actualidad, con este tipo de aplicaciones se puede

tomar decisiones en tiempo real en los procesos de producción y de esta forma poder analizar datos que se lleva monitoreando, una de sus funciones principales seria la adquisición de data para que luego dicha información pueda procesarse y mediante algoritmos buscar mejoras que nos ayuden con los procesos buscando así que los procesos en las industrias sean más eficaces y confiables (Pérez-López, 2015).

Figura 6

Aplicación intouch control y monitoreo



Entre estas aplicaciones se pueden generar tipos de alertas o estados de condiciones, el cual suelen ser notificaciones de alertas para avisar en caso de que algún parámetro que está siendo monitoreado este fuera del umbral establecido, este tipo de acciones da paso a una serie de ventajas a este tipo de aplicaciones que nos previene de algún riesgo que pueda ocasionarse en el proceso.

1.2.6 Microcontroladores y actuadores

Un control automatizado está compuesto de microcontroladores y actuadores que por lo general son sistemas embebidos, se puede considerar a un microcontrolador como un chip

integrado que tiene memoria, procesador, entradas y salidas; las mismas que están destinadas para cumplir funciones como procesar información y poder realizar tareas que sean asignadas según el requerimiento que se desee.

A continuación, partes y funcionamiento importantes de un microcontrolador:

Tabla 5

Partes principales de un microcontrolador

Nombre	Función
Unidad central de procesamiento (CPU)	Controla el funcionamiento y ejecuta instrucciones guardadas en el microcontrolador
ROM	Guarda el programa que se ejecutara en el microcontrolador
RAM	Memoria volátil, se presentan datos temporales de ejecución
GPIO	Puertos configurables: entradas/salidas; digitales/analógicas
Timmer/Counters	Realizan conteos y gestionan tiempo
Interfaces de comunicación	Comunicación: I2C, UART, SPI, etc.

Este tipo de microcontroladores suelen tener aplicaciones como en sistemas autónomos, sistema de control Industrial, electrodoméstico, agricultura, etc. Los cuales tienen la misma función dar paso al procesamiento de señales, adquisición y gestionar de los datos que se manejará en el sistema embebido que se programará en el mismo.

Los actuadores en este caso son parte de una señal de control generada por el microcontrolador, en mucho de los casos es generada por un pulso eléctrico, convirtiéndose en una acción física, ya sea este un movimiento o una fuerza efectuada a un instrumento como también el paso o cierre de una señal.

A continuación, presentamos algunos de los tipos de actuadores:

Tabla 6

Tipos de actuadores

Tipo	Función, nombre
Actuador eléctrico	Motores DC/AC: Energía eléctrica en movimientos. Sercomotores: posicionamiento de un eje en un ángulo específico. Solenoides: actuadores que general movimiento lineal.
Actuador hidráulico	Se utilizan donde se requiere de gran fuerza, ya que se utiliza fluidos de presión para generar movimientos rotatorios o lineales.
Actuador neumático	A través de aire comprimido generan movimientos, estos son comunes en sistemas de automatización industrial.

Los actuadores suelen utilizarse en el campo de la robótica, automatización industrial, sistemas autónomos, etc. Al integrar ambos componentes procesan la información y luego envía señales para que pueda realizar la acción que se le asigna realizar según las condiciones dadas.

1.2.7 Adquisición de datos

DAQ denominada en sus siglas en inglés la adquisición de datos es el proceso de medir, recopilar y analizar la información que se adquiere a través del entorno físico ya sea este por un microcontrolador o un PLC. La adquisición de datos puede ser de variables como: oxígeno, temperatura, presión, humedad, etc.

Componentes básicos para la adquisición de datos para un sistema:

Tabla 7

Componentes básicos de una DAQ

Componentes	Función
Sensores	Dispositivos de lecturas en variables físicas.
Acondicionamiento de señal	Amplificadores: ayuda a implementar la señal Filtros: elimina ruido generado por factores externos.
Dispositivo de adquisición	Tarjetas y módulos DAQ: dispositivos externos que se instalan en una PC mediante interfaces. Registro de datos: Equipo autónomo que se encarga de almacenar datos.
Software de adquisición de datos	Análisis de datos e interfaz de usuario: software que se utiliza para la visualización de datos y análisis; también permite monitorear y configurar el sistema DAQ.

La DAQ está en aplicaciones como la investigación y desarrollo, controles ambientales, automatización Industrial, etc. Uno de sus mayores beneficios es la precisión y exactitud que nos presenta en tiempo real además de analizar las tendencias y comportamiento de la data.

1.2.8 Sistemas de monitoreo

Un sistema de monitoreo es responsable de brindar información sobre desempeños que se están evaluando en algún proceso, con la finalidad que nos ayude a la toma de decisiones y prevención de riesgos que alteren patrones de funcionamiento del sistema, también se enfoca en medir resultados de producción.

1.2.8.1 SCADA

Un sistema de monitoreo SCADA por sus siglas en inglés (control system architecture comprising computers) es una arquitectura de control industrial que permite supervisar, controlar, recopilar, analizar datos y estadísticas en tiempo real. A continuación, se presenta los componentes básicos para un sistema SCADA.

Tabla 8

Componentes básicos de un sistema SCADA

Componentes	Función
Microcontroladores y Controladores Lógicos Programables	Dispositivos y controladores que permiten conectar sensores y controlar que se ejecuten la programación lógica para la automatización de procesos.
Sensores y actuadores	Medición de variables y dispositivos que ejecutaran una acción según las condiciones dadas.
Comunicación	Redes y protocolos de comunicación Modbus. Ethernet/IP.
Estación de operador	Interfaz gráfica donde se presentan datos en tiempo real.
Base de datos	Almacenamiento de datos históricos y generación de reportes.

Entre sus principales funciones aparte de generar un histórico de reporte permite generar alarmas, que nos ayudara a generar un historial para luego con un poco de tecnología poder prevenir siniestros y predecir sucesos que se pueden evitar según el sistema del proceso ya establecido.

1.2.9 Inteligencia artificial (IA) y sus aplicaciones

La inteligencia artificial se centra en el campo de la informática que es capaz de realizar tareas que requieren de inteligencia humana. La IA se basa en el aprendizaje autónomo analizando patrones de comportamiento, al analizar un conjunto de datos y con el autoaprendizaje nos puede ayudar para poder generar predicciones del comportamiento del proceso que se está llevando.

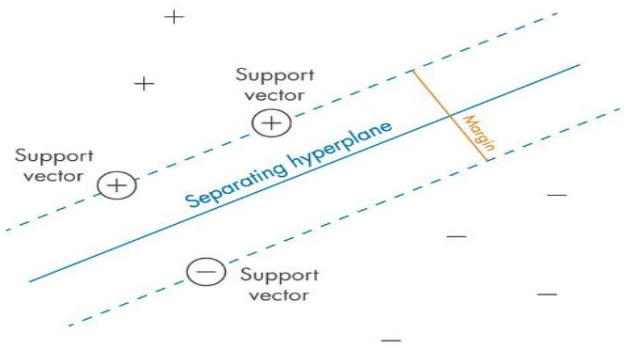
1.2.9.1 Support Vector Machine (SVM)

Entre la inteligencia artificial existen varias técnicas de aprendizaje según la necesidad que se requiera analiza, entre ellas esta SVM (máquina de vectores soporte) en sus siglas en inglés, es un algoritmo que se utiliza en procesos de clasificación, entre más aplicaciones como procesamiento de señales, reconocimiento de voz, imágenes y procesamiento de lenguaje natural.

El principal objetivo es encontrar un hiperplano donde separa los datos en dos clases, de la mejor forma posible. Donde esto implica marcar un margen, el cual se define como la anchura entre datos en su región paralela al hiperplano; este tipo de aprendizaje maximiza el margen permitido para la clasificación errónea.

Figura 7

Definición del "margen" entre clases: criterio que los SVM intentar optimizar



1.2.9.2 Árbol de decisiones

Un árbol de decisión es un mapa visual que ayuda a la toma de decisiones de posibles resultados de decisiones relacionadas. Este nos ayuda a priorizar los resultados estimados y tomar decisiones de manera estratégica para la mejora y planificación. A continuación, se detalla los componentes de un árbol de decisión:

Tabla 9

Componentes básicos del Árbol de decisión

Componentes	Función
Nodo Raíz	Esto representa la decisión inicial que se tomara o pregunta inicial.
Nodos Internos	Cada nodo es una prueba o decisión, estos nodos son subconjuntos basados en decisiones.
Ramas	Cada rama representa un resultado de una prueba.
Nodo Hoja	Representan las decisiones finales o las clasificaciones después de seguir todas las decisiones previas.

CAPÍTULO 2. METODOLOGÍA

2.1. Contexto de la investigación

La camaronera perteneciente a la asociación de camaroneras Acuarios del Golfo está ubicada en la parroquia Taura perteneciente al cantón Naranjal, se encuentra ubicado en una zona de varias hectáreas de camaroneras, donde se realiza la crianza y engorde de camarón con un aproximado mensual de 400.000lb, la misma consta de 21 piscinas de engorde y 11 pre criaderos distribuidos estratégicamente.

Figura 8

Camaronera

La piscina 10 cuenta con 8.63 hectáreas, en donde se implementará el proyecto esta cuenta con aireación mecánica de combustión a diésel, aproximadamente la piscina cuenta con un aireador por hectárea por lo cual dicha piscina tiene 8 aireadores en zonas estratégicas para la generación del OD.

Figura 9

Piscina 10 Satukin

Ilustración 9. Piscina 10 Satukin

Los aireadores que se encuentran colocados en la piscina 10 son de 16HP marca CHANGFA, estos cuentan de un sistema de paletas que al accionar el motor generará las partículas de oxígenos, que ayudaran a mantener un nivel adecuado para la prevención de bajas de oxígeno y que ocurran siniestros.

Figura 10

Motor Changfa ZS110BN(M)

2.2. Diseño y alcance de la investigación

El alcance de la propuesta contempla la creación de un sistema SCADA que permitirá el control y supervisión del parámetro OD correspondiente a la piscina 10 de Satukin. La piscina tiene aireación mecánica, por lo cual, se plantea realizar comparaciones de gastos por consumos, tiempos que los motores deben estar encendidos y el cálculo de generación de OD que generan cada uno de ellos. El modelo que se plantea será con la recopilación de la data real, teniendo esta información nos ayudará a definir y tomar mejores decisiones con respecto al manejo de la aireación.

Este diseño es de carácter experimental ya que tendrá dos conjuntos a estudiar, teniendo en consideración que los valores que se obtendrán mediante el sensor Knf-103b será el responsable mediante un algoritmo, el cual permitirá tomar la decisiones y condiciones adecuadas para el funcionamiento de la aireación y obtener consumo y eficiencia, todo esto se presentará mediante un SCADA donde se obtendrán los valores en tiempo real y nos ayudara a tomar acciones que se darán según las condiciones dadas.

Adicional, como alcance se espera obtener una investigación analítica que nos permita medir relaciones causales entre ellos; con ello se plantea receptar datos de oxígeno a través de un sistema autónomo que permita llevar un control constante y así disminuir el consumo de energía en los motores y de esta forma ser comparados.

2.3. Tipo y métodos de investigación

Al obtener el parámetro de oxígeno disuelto en el agua entraremos a una investigación cuantitativa, en la cual, tendremos inferencia según los parámetros que se obtengan y así poder extrapolar los resultados de las muestras obtenidas, mediante la recepción que nos permitirá identificar los momentos críticos y patrones del comportamiento. Con ello, podemos tener una idea del patrón de comportamiento del oxígeno en la piscina, que a su vez a través de la investigación nos dará paso para poder comparar la cantidad de motores necesarios con los que se encuentran en la piscina, esto nos ayudará con la demanda de motores necesarias para la generación de OD.

Mientras tanto la investigación cuantitativa cobra sentido cuando se relaciona con otras mediante correlaciones, regresiones o pruebas de hipótesis que es lo que se quiere realizar con la investigación. Por ello, el enfoque que se dará es Hipotético–Deductivo ya que estará basada en la formulación de hipótesis para la solución de problemas, aceptación o modificación de la hipótesis y por último la deducción de las consecuencias observadas del proceso. Con la adquisición de la data se utiliza una metodología analítica ya que se efectuarán análisis de los resultados y sus variaciones según el enfoque que se le pueda dar, ya sea esta en la generación del OD o consumos realizados.

2.4. Población y muestra

Se considerará como muestras a los parámetros de lecturas de oxígeno que presente la piscina, supervivencia de los organismos sembrados y los costos comparativos de producción.

Tabla 10

Producción de piscina 10

Parámetros	Datos
Oxígenos	(1.0 a 8 mg/l)
Libras estimadas	110000lb
Supervivencia estimada	85%

Las muestras y adquisición de los datos nos ayudaran para el entrenamiento que se dará para la toma de decisiones, la piscina cuenta con aireación mecánica.

2.5 Técnicas e instrumentos de recolección de datos

Como base de datos los valores de oxígeno se encuentran registrados por parte del personal de la camaronera por día, la supervivencia se la lleva en cuenta mediante poblaciones realizadas semanalmente para la evaluación de los costos que se generan en la crianza del camarón, estos serán comparados con los datos que se obtendrán con el proyecto a realizar

Para recopilación y adquisición del dato del oxígeno se implementó un sistema de alimentación, sistema de comunicación y procesamiento del dato con los siguientes materiales y software:

2.5.1 Sistema de alimentación

La generación de energía para el sistema estará dada por una autonomía solar.

2.5.1.1 Panel solar

El panel solar se encarga de receptar a través de sus fotoceldas energía solar y la convierte en electricidad o corriente continua. A continuación, se detalla los aspectos principales del **Solar PV Module** que se utilizó.

- Rated Maximum Power (Pmax): **50W**
- Voltage at Pmax (Vmp): **18.6V**

- Current at Pmax (Imp): **2.69A**

Figura 11

Panel solar 50W

2.5.1.2 Controlador de carga

Usualmente los controladores de carga son utilizados en los sistemas autónomos solares para regular el flujo de energía, el controlador de carga PWM (modulación de ancho de pulso) es utilizado en circuitos donde las variaciones de voltajes no son muy inestables; incluyen protecciones contra sobrecargas, también a través de su pantalla nos ayuda a verificar que las conexiones estén realizadas de manera correcta, además de ver el estado de la batería. A continuación, presentamos parámetros básicos del **SOLAR CHARGE CONTROLLER PWM:**

- Rated voltaje: **12V/24V**
- Rated current: **10A**
- Max.PV Voltage: **50V**
- Max. Pv Input power: **130w(12V), 260W(24V)**

Figura 12

Controlador de carga

2.5.1.3 Batería

La batería se encarga de almacenar energía que es suministrada por el controlador; son fundamentales en varias aplicaciones, desde sistemas de almacenamiento de energía hasta dispositivos electrónicos. A continuación, detallaremos aspectos importantes de la batería que se utilizó.

- Voltaje 12v
- Amperaje 20A
- Batería de uso fotovoltaicas, rápida carga

Figura 13

Batería DACAR 12/20

2.5.2 Sistema de comunicación y adquisición del dato

Para la adquisición de dato es necesario un microcontrolador o un PLC que sea el encargado de receptar el dato para luego transmitirlo, con ello se debe tomar en cuenta el protocolo de comunicación que posee el sensor para establecer la comunicación con el microcontrolador.

2.5.2.1 ESP32

El ESP32 es un microcontrolador con conectividad inalámbrica IoT mediante Wi-Fi o Bluetooth, al contar con este tipo de comunicación se evita matrices de la radiofrecuencia

(RF), esta gama de microcontrolador es de bajo costo, consumo y alto rendimiento. En la tabla que se muestra a continuación sus características importantes:

Tabla 11

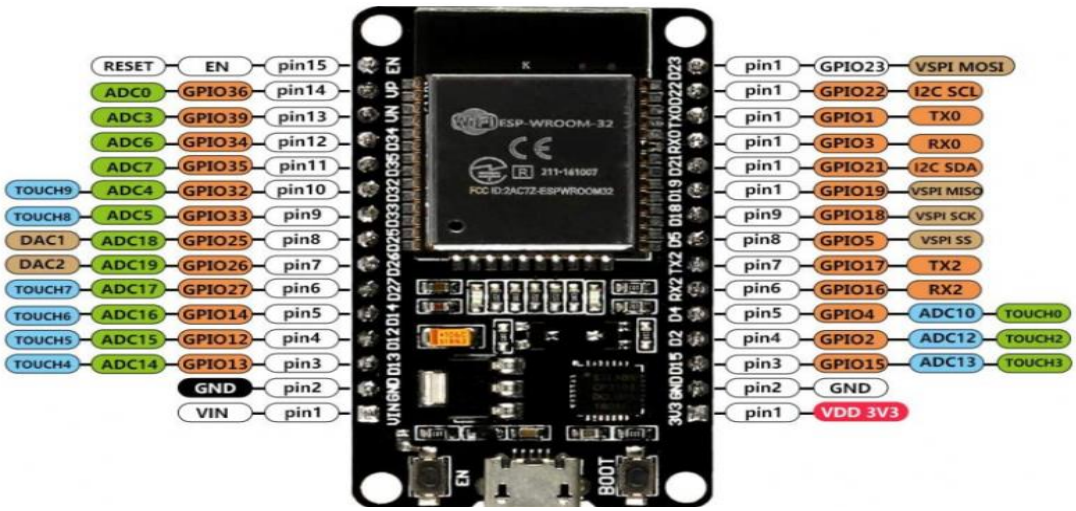
Características generales del ESP32

Características Generales	Especificaciones
CPU	Dual-core o single-core, 32bits de arquitectura.
Memoria	SRAM integrada de 520kb. ROM integrada de 448kb.
Conectividad Inalámbrica	Wi-Fi 802.11 b/g/n. Bluetooth 4.2
Entrada y Salida	34 pines GPIO (General Purpose Input/Output).
Periféricos Integrados	Sensores, temporizadores, contadores y reloj en tiempo real.
Seguridad	Cifrado AES, SHA, RSA y ECC.
Consumo de energía	Modos ahorros de energía (Deep sleep) e hibernación.

El ESP32 pude ser utilizado en varios entornos de desarrollo como: Arduino IDE, PlataformIO, Espressif IDF y MicroPython; es un microcontrolador potente y versátil con una variedad de características según el uso que se le pueda dar.

Figura 14

ESP32



2.5.2.2 Sensor KNF-103b

El sensor de oxígeno disuelto utiliza el principio de excitación de fluorescencia para medir la concentración de oxígeno disuelto en el agua. La detección óptica con la que cuenta puede eliminar eficazmente la interferencia de la fluctuación del pH, el nitrógeno amoniacal y metales pesados en el agua, proporcionando así resultados más estables y precisos durante un periodo de tiempo más largo (Ltd, s. f.).

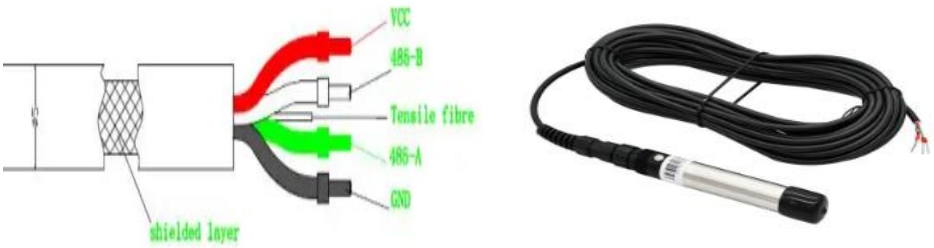
A continuación, presentamos parámetros del sensor KNF-103b:

- Lectura de rango: 0-20mg/l.
- Precisión: 3%.
- Tiempo de respuesta: 60s.
- Tamaño de producto: 22mm*188.3mm.
- Cable de sonda: 5 metros.
- Vida útil de la tapa de fluorescencia: un año.
- Presión máxima de funcionamiento: 6 bar.
- Salida de datos: RS485 y MODBUS RTU.
- Fuente de alimentación: DC 12v, corriente <50mA.

- Nivel de protección: IP68.

Figura 15

Sensor de OD KNF-103b



Este tipo de producto esta adecuado para la acuicultura (camarón blanco sudamericano, pez osmanthus, etc), tratamientos de agua y otros requisitos de medición. Debemos tomar en cuenta que se debe evitar exponer la superficie interior de la tapa fluorescente a la luz solar. A continuación, se detallará el tipo de conexión de los cables referente a la ilustración 15

- Línea color rojo: fuente de alimentación (VCC)
- Línea color verde: Dato (485_A)
- Línea color blanca: Dato (485_B)
- Línea color negro: Cable de tierra (GND)

2.5.2.3 MAX485

El dispositivo MAX485 en un transceptor de comunicación, es un estándar utilizado para comunicación serial y tiene capacidades para recepción o envío de datos a largas distancias; presentaremos algunas de sus características a continuación:

- Este dispositivo puede operar en medio dúplex
- Funciona a una tensión de 5V
- Soporta velocidades de transmisión hasta 2.5Mbps
- Capacidad hasta 32 transceptores en un mismo bus de comunicación

Figura 16

MAX485

Este dispositivo cuenta con puertos (A y B) que llevan señales complementarias, el cual ofrece varias ventajas como: inmunidad al ruido y mayor distancia de transmisiones. A continuación, mostraremos los puertos de conexión del MAX485:

- **RO (Receiver Output):** Salida del receptor.
- **RE (Receiver Enable):** Habilita/deshabilita al receptor. Activo bajo
- **DE (Driver Enable):** Habilita/deshabilita al transmisor. Activo alto
- **DI (Driver Input):** Entrada de datos para el transmisor.
- **A y B:** Líneas de transmisión diferencial.
- **Vcc:** Alimentación (5V).
- **GND:** Tierra.

2.5.3 Procesamiento del dato

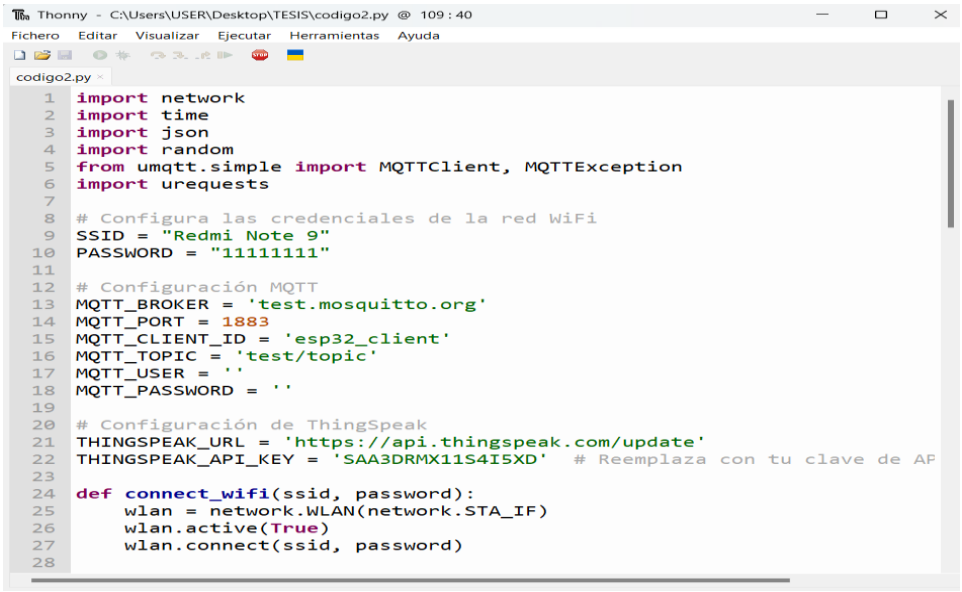
Para el procesamiento y presentación del resultado se utilizará distintos Softwares, de los cuales, cada uno tendrá una función diferente pero esencial para el procesamiento y muestras de resultados.

2.5.3.1 Thonny

El programa Thony es un entorno de desarrollo integrado, desarrollado en la Universidad de Tartu en Estonia, Thonny tiene características que facilitan el aprendizaje de la programación.

Figura 17

Aplicación Thonny



Thonny es compatible con MicroPython lo que lo hace útil para utilizar dispositivos como el ESP8266 y ESP32; además de ello proporciona un entorno seguro y facilita la depuración y desarrollo de programas para dispositivos embebidos.

2.5.3.2 MQTT Explorer

La aplicación MQTT Explorer está diseñada para interactuar con Brokers MQTT y la supervisión de los mensajes que circulan por su sistema. El protocolo MQTT es utilizado ampliamente en IoT para la comunicación entre dispositivos, entre sus funciones principales esta:

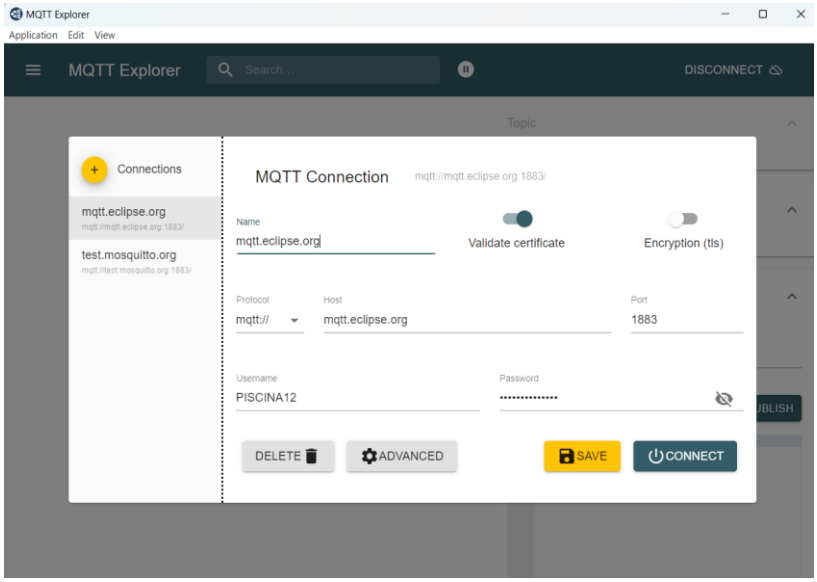
- Visualización de Estructura de temas.
- Publicación y Suscripción de temas.

- Historial de mensajes.
- Visualización de mensajes en tiempo real.
- Compatibilidad de varios Brokers.

Por lo general se utiliza en aplicación IoT, monitoreo de redes IoT y para mostrar su funcionamiento como herramienta educativa para protocolos MQTT; una de sus mayores ventajas es su fácil configuración y la visualización en tiempo real que proporciona la aplicación.

Figura 18

Aplicación MQTT Explorer



2.5.3.3 InfluxDB

InfluxDB es una base de datos, la cual está diseñada para el manejo de volúmenes de datos generados por sensores, aplicaciones de monitoreo con marca de tiempo real. A continuación, se muestra características principales de InfluxDB:

- Optimizar datos de Series Temporales.
- Alta Disponibilidad y Escalabilidad.
- Almacenamiento y comprensión Eficiente.
- Consultas y agregaciones en tiempo real.

Está base de datos altamente segura y eficiente al manejar datos de series temporales, también al adquirir datos en tiempo real nos ayuda con el análisis de la data y sus valores históricos generar un posible entrenamiento de IA para la ayuda de toma de decisiones de los procesos.

Figura 19

*In*fluxDB

2.5.3.4 KepServer

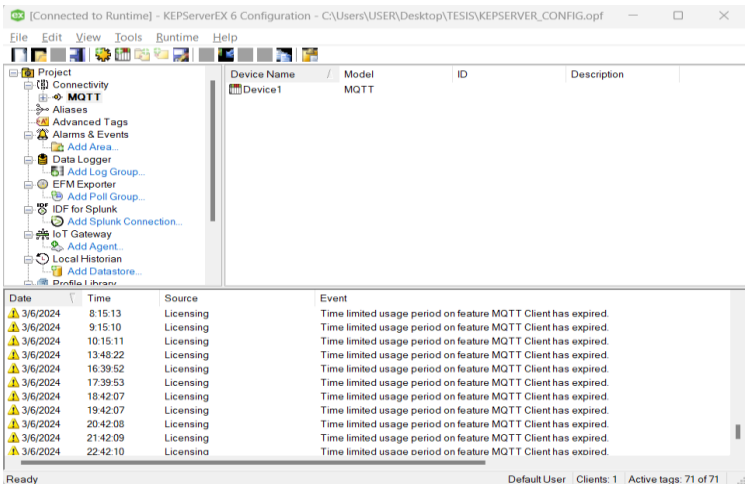
Es una aplicación de conectividad industrial, su principal objetivo es facilitar la intercomunicación entre Software y dispositivos; es muy útil para la gestión y control de procesos industriales. A continuación, se mostrará las características principales de kepserver:

- Compatibilidad con Múltiples Protocolos
- Arquitectura Modular
- Configuración y Gestión Centralizada
- Escalabilidad
- Integración con IT

Entre sus usos, facilita la comunicación entre dispositivos de diferente fabricantes y sistemas de control, por lo que la hace una solución de conectividad versátil; ayudaría fundamentalmente a procesos que están regidos a varios tipos de subprocesos que se deben monitorear exhaustivamente para que esta sea eficiente y con la adquisición de datos tenga la potestad de la toma de decisiones para mejorar los estados de los procesos.

Figura 20

Programa KepServer



2.5.3.5 Intouch y thingSpeak

Ambas aplicaciones sirven para la muestra de datos en tiempo real y análisis, una de forma local (Intouch) que se programa de manera gráfica las variables de lecturas y configuraciones le quieres dar al sistema SCADA y la otra a través de la red (thingSpeak), donde solo será tipo lectura de datos; estas aplicaciones permiten la recopilación de datos a usuarios destinados que tengan los accesos necesarios para los mismos. Para ello, detallaremos algunos aspectos importantes de estas aplicaciones:

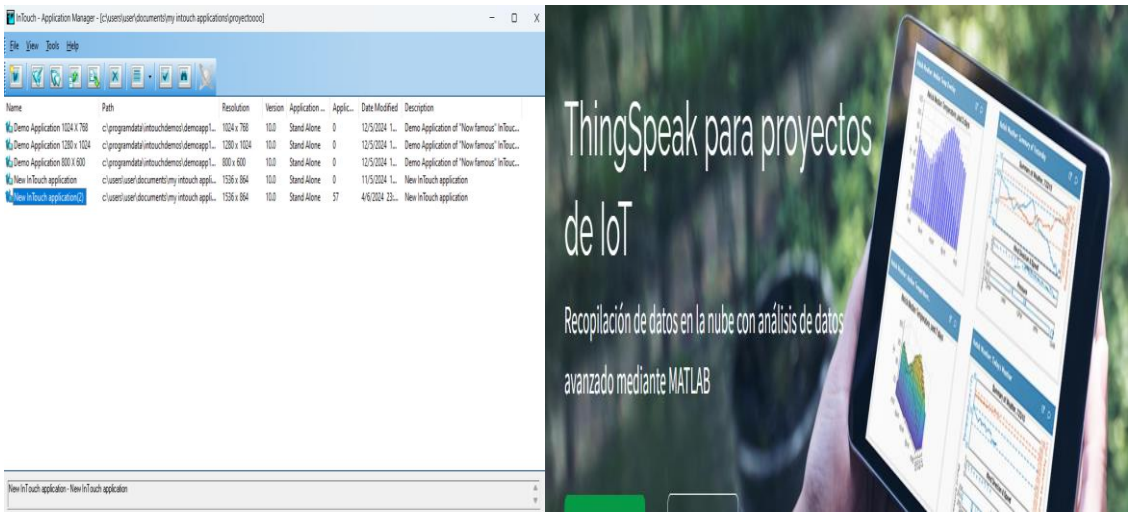
- Almacenamiento local y en la nube

- Visualización de datos.
- Análisis y procesamiento de datos.
- Alertas y notificaciones.
- Interacciones con dispositivos IoT.
- Accesos y control de datos.

Usos comunes se da en la parte de monitoreos y creaciones de sistemas SCADA; estas plataformas se adecuan tanto para aplicaciones personales e industriales, ThingSpeak al ser una integración de Matlab aumenta sus capacidades haciéndola una herramienta poderosa para ámbitos de Iot.

Figura 21

Aplicación Intouch y plataforma ThingSpeak



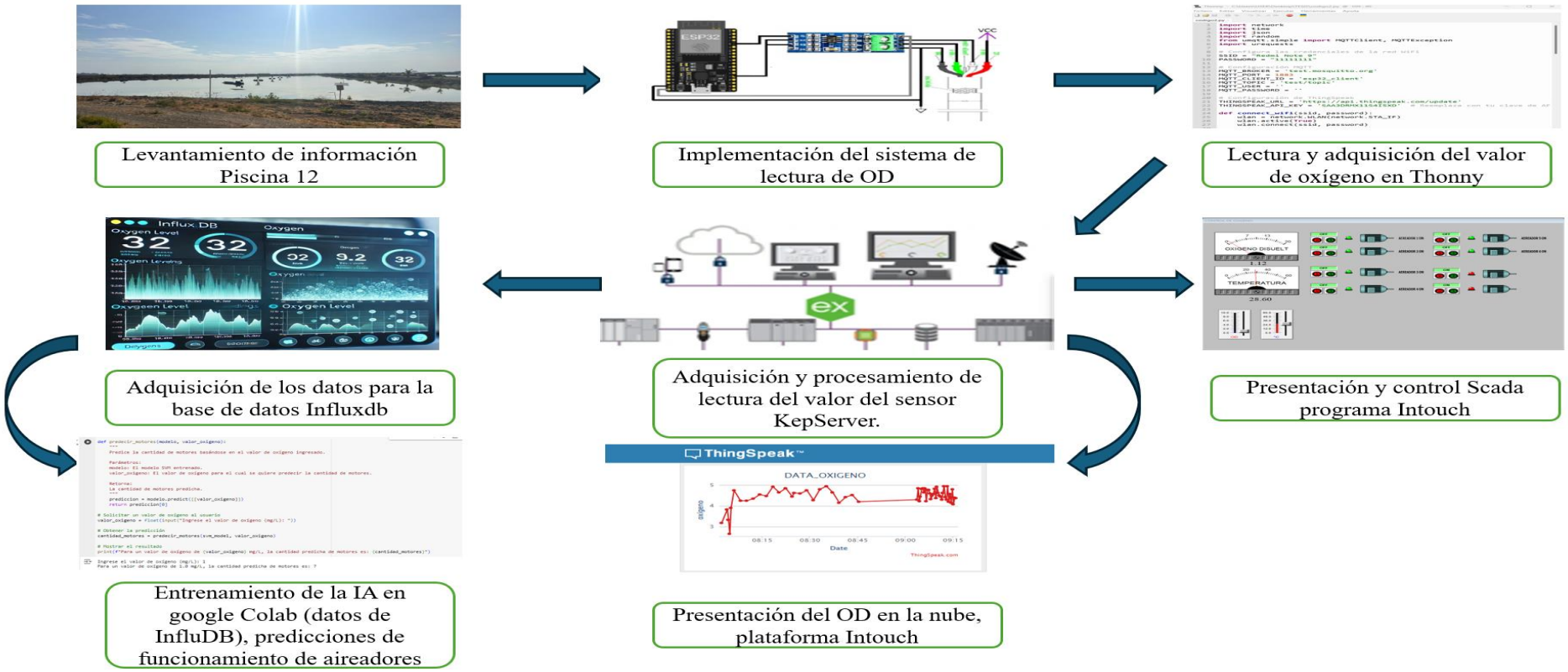
2.6. Procesamiento de la evaluación: Validez y confiabilidad de los instrumentos aplicados para el levantamiento de información.

La evaluación se realizará mediante la comparación de datos obtenidos por los equipos de medición que se encuentran en la actualidad funcionando en camaronera y los datos que se obtendrán con el proyecto, la fiabilidad que garantizará el buen funcionamiento

radicará en el análisis estadístico que se obtendrá al evaluar y comparar resultados de los equipos de medición que estarán implementados.

Figura 22

Metodología del desarrollo de la propuesta



2.6.1 Visualización y Procesamiento de la adquisición del sensor de Oxígeno

La visualización y procesamiento de los datos del sensor de oxígeno usando un protocolo MQTT para mostrarlo en ThingSpeak nos proporcionan una solución potente para el monitoreo, análisis y toma de decisiones en tiempo real.

2.6.2 Desarrollo y evaluación de los datos obtenidos con los equipos de medición tradicionales

Este proceso nos permitirá evaluar los datos obtenidos a través del equipo de medición tradicional con el sistema que se implementará obteniendo así una precisión y fiabilidad que asegure que los equipos funcionen de manera correcta.

2.6.3 Toma de decisiones según los parámetros establecidos y modelos de IA

La combinación de parámetros establecidos y modelos de IA facilitan la toma de decisiones informadas y automatizadas, mejorando la eficiencia y efectividad en las aplicaciones automatizadas que se le asignará al sistema.

CAPÍTULO 3. RESULTADOS Y DISCUSIÓN

A continuación, se mostrará los resultados obtenidos; conexiones de equipos, adquisiciones de datos, sistema de comunicación entre dispositivos, desempeño de las arquitecturas, implementación de la IA para detección de fallas y toma de decisiones que nos ayudaría para la obtención de mejores resultados.

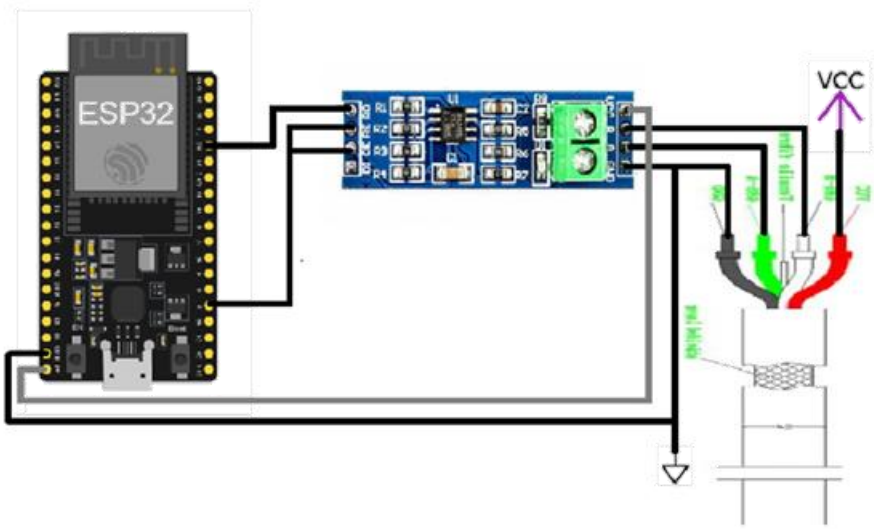
3.1 Adquisición de datos del sensor de oxígeno

3.1.1 Conexión física de equipos y alimentación del sistema

Para la implementación de la comunicación entre equipos, se debe identificar cada puerto de los equipos que vamos a utilizar para la correcta comunicación entre sí, en este caso utilizaremos un **ESP32** que nos permitirá tener la conexión a internet, de la misma forma nos ayudará para la recepción de datos a través del módulo **MAX485**, el cual mediante una conexión modbus RTU nos permitirá receptar el dato del sensor **KNF-103B.**

Figura 23

Conexión ESP32, MAX485 y KNF-103b



Conexiones

Tabla 12

Conexiones ESP32-MAX485 y MAX485-KNF-103b

Esp32	Max485	Max485	Knf-103b
Ro	Rx	Vcc	Vcc (5v Esp32)
Re	GPIO2	A	A
De	GPIO2	B	B
		Gnd	Gnd y común

El sistema de alimentación de los equipos está dado por un sistema solar autónomo, este tipo de sistemas opera de manera independiente de la red eléctrica, de esta forma está diseñada para capturar energía del sol a través del panel solar, el controlador se encarga de que la energía generada por los paneles se almacene de manera eficiente en las baterías, de la misma forma sirve para proteger de sobrecargas a la salida del terminal de consumo.

Figura 24

Sistema de alimentación solar

3.1.2 Conexión a internet y adquisición de datos

Como primer paso debemos tener actualizado el software más reciente del esp32, con ello procedemos con la siguiente etapa del desarrollo como anexo 1 queda la programación implementada en el microcontrolador. A continuación, se presenta en la siguiente figura 25 el código que se generó en Thonny para establecer conexión a la red WI-FI denominada “Redmi Note 9”.

Figura 25

Codificación para la conexión a internet del ESP32

```
import machine
import time
import network

# Nombre y contraseña de tu red WiFi
SSID = "Redmi Note 9"
PASSWORD = "1111111"

# Conectarse a la red WiFi
def connect_wifi(ssid, password):
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    wlan.connect(ssid, password)
    while not wlan.isconnected():
        pass
    print("Conectado a la red WiFi")
    print("Dirección IP:", wlan.ifconfig()[0])

# Llamar a la función para conectar a la red WiFi
connect_wifi(SSID, PASSWORD)
```

Una vez establecida la conexión a internet, el ESP32 imprimirá la dirección IP asignada. En este caso, se ha asignado la **IP 192.168.12.10,** esta dirección IP nos servirá para poder establecer comunicación con el resto de los dispositivos y software que deberán estar conectados entre sí. Debemos tomar en cuenta que el servidor será la PC donde estará instalada la aplicación kepServer. Por lo tanto, es necesario ejecutar el comando ip config en el símbolo de sistema (cmd) de Windows para poder obtener la dirección IP de nuestra PC, donde se asignó **IP 192.168.12.22** que nos presenta para que se establezca la comunicación.

Por lo tanto, se genera un código donde intervienen todos los protocolos para establecer comunicación modbus con sus respectivas Id y puertos asignados como se explica en el siguiente código.

Figura 26

Codificación para leer dato del sensor por protocolo Modbus RTU

```
# Función para leer datos Modbus
def read_modbus_data():
    # Crear un socket TCP para la conexión Modbus
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    try:

        # Conectar al servidor Modbus
        sock.connect((MODBUS_SERVER_IP, MODBUS_SERVER_PORT))

        # Leer datos Modbus
        request = tcp.read_input_registers(slave_id=1, starting_address=0, quantity=2)
        response = tcp.send_message(request, sock)
```

En el código generado en la figura 26 se define las conexiones wifi a la misma red donde estará conectado al conjunto de ip del puerto servidor Modbus, luego de ejecutar se envía una solicitud modbus para leer los datos en los puertos establecidos.

Una vez que ya tenemos generado el código de lectura de datos, se procede a realizar el código para la conexión a **kepserver** donde se deberá crear un canal de comunicación con ello procederemos a estableceremos a crear una comunicación MQTT.

3.2 Implementación de protocolo MQTT mediante una ESP32

Para implementar la transmisión de datos a través del protocolo MQTT, utilizaremos la aplicación **MQTT Explorer,** en ella podemos generar los topic necesarios según los requerimientos que se necesite realizar, en este caso crearemos uno para la recepción del dato de OD. Para la configuración del protocolo MQTT, se debe tener claro que la configuración del:

BRÓKER es esencial ya que es la pieza central en la red, facilita la comunicación entre dispositivos por ello se debe configurar con el mismo nombre tanto como en la aplicación como en el código para validar que el suscriptor esté conectado al Bróker.

PORT el puerto asignado 1883 se utiliza como un puerto predeterminado para MQTT, el mismo se ha estandarizado y se lo utiliza entre comunicaciones cifradas y no cifradas, facilitando el despliegue y la interoperabilidad en sistemas en entornos IOT.

CLIENTE_ID se da un nombre de usuario para que pueda ser identificado el suscriptor.

TOPIC se utilizan para identificar y organizar los mensajes en una cadena de texto jerárquica que se envía entre el cliente a través de un Bróker, en este caso será el topic (test/topic) el que se encargará de receptar el dato del sensor de oxígeno.

USER Y PASSWORD se los utiliza para restringir información a diferentes tipos de usuarios, en este proceso en caso de tener un usuario y contraseña deben estar tanto en la codificación como en el software para que pueda ingresar a el registro del topic.

Figura 27

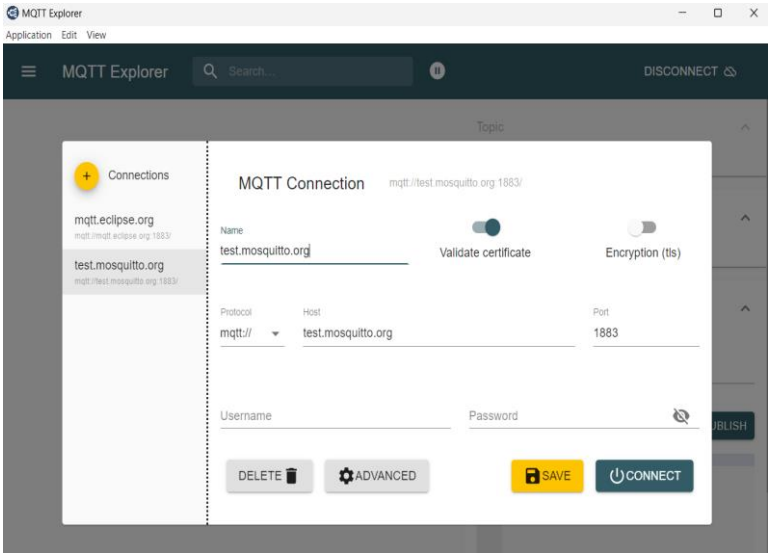
Codificación Thonny para establecer comunicación con MQTT Explorer

```
# Configuración MQTT
MQTT_BROKER = 'test.mosquitto.org'
MQTT_PORT = 1883
MQTT_CLIENT_ID = 'esp32_client'
MQTT_TOPIC = 'test/topic'
MQTT_USER = ''
MQTT_PASSWORD = ''
```

Luego de tener establecida la codificación en Thonny, con los parámetros necesarios que requiere el protocolo de comunicación MQTT, procedemos a configurar los parámetros ya antes descritos en la Figura 27 para poder establecer comunicación entre ambas plataformas tal y como se muestra en la siguiente Figura:

Figura 28

Configuración de parámetros de comunicación en MQTT Explorer



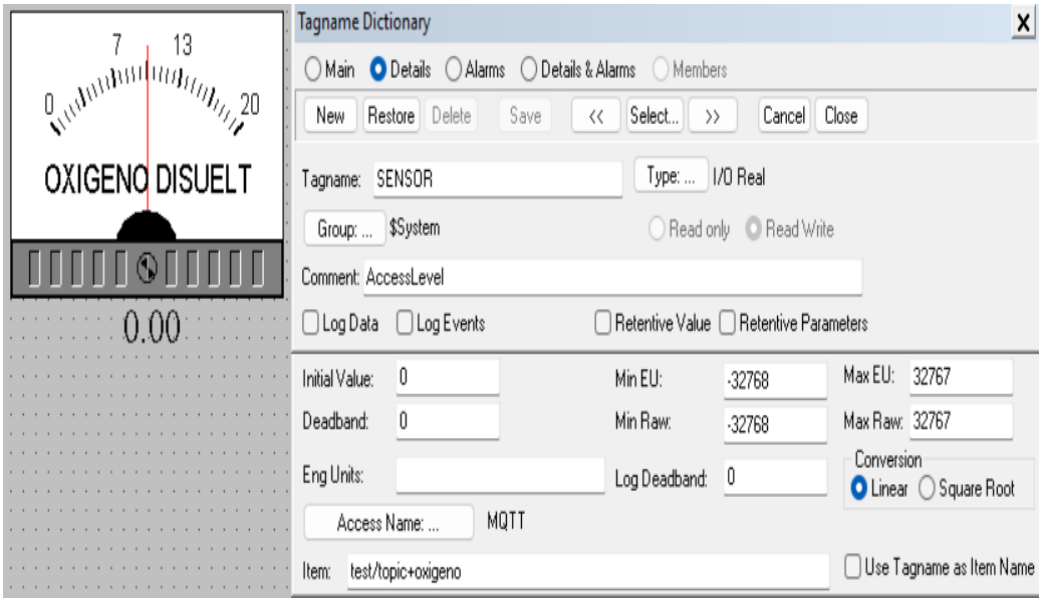
3.3 Diseño del sistema SCADA

Para el diseño del sistema de monitoreo en la plataforma Intouch, primero debemos tomar en cuenta las variables que vamos a utilizar y sus respectivas configuraciones, al ser un programa versátil nos ayuda a crear sistemas simulados y reales. Por ello, se debe tener en consideración que las variables de lectura de un sensor real o activaciones de motores deben estar asignadas al tipo de variable correspondiente. A continuación, presentamos los pasos que se realizó para la creación de los datos de lectura de OD y los controladores utilizados para su funcionamiento:

- Creación de la ventana editable de Intouch y colocaremos las variables entradas/salidas del sistema SCADA.
- Acondicionamiento de la pantalla donde se va a reflejar el valor del OD, los motores y lámparas que se encenderán con cada proceso requerido.
- Configuración de entras I/O y protocolos de MQTT con el topic correspondiente configurado para la lectura desde la aplicación MQTT Explorer y kepServer; con ello pueda recibir la información de entradas y salidas del sistema.

Figura 29

Configuración modo lectura para el sensor de OD



En este software establecemos los puertos de entrada y salida de la misma forma en que se va a configurar en la plataforma kepserver; con ello, los registros guardados sean presentados. A continuación, se muestra la tabla de tag establecidos:

Tabla 13

Registros y tags de entradas y salidas

TAG	Tipo (dirección)	MODO
SENSOR	Char (400001)	Lectura
M1	Bolean (000001)	Salida
M2	Bolean (000002)	Salida
M3	Bolean (000003)	Salida
M4	Bolean (000004)	Salida
M5	Bolean (000005)	Salida
M6	Bolean (000006)	Salida

M7	Bolean (000007)	Salida
M8	Bolean (000008)	Salida

Teniendo establecidos todos los parámetros en los programas procedemos con la lectura del sensor donde se recolecto los datos, teniendo los siguientes resultados en un día de seguimiento.

3.4 Implementación de influxDB

Para la base de datos de nuestro trabajo utilizaremos influxDB, donde obtendremos la data en .csv, la misma que nos servirá para el entrenamiento para la IA. A continuación, se mostrarán los pasos a seguir para obtener los valores que se generaron en la lectura del sensor

En Kepsever ya tenemos creado el Tag llamado SENSOR como se muestra en la tabla 13 se procede a generar en InfluxDB a través de código la creación de la base de datos con el siguiente comando CREATE DATABASE OXIGENODB.

Para la configuración entre kepserver y influx necesitaremos de la interfaz NODE-RED donde previo la instalación necesitaremos instalar el paquete "node-red-contrib-influxdb", el que nos permitirá recibir datos desde Kepserver y escribir en InfluxDB.

3.5 Configuración de la plataforma en la nube (thingSpeak)

Para mostrar los resultados en la plataforma ThingSpeak se debe realizar las creaciones de los canales, de la misma forma realizar la codificación en el IDE de Thonny para la intercomunicación entre las plataformas. A continuación, los pasos que se realizaron para la configuración en ThingSpeak después de tener creado una cuenta:

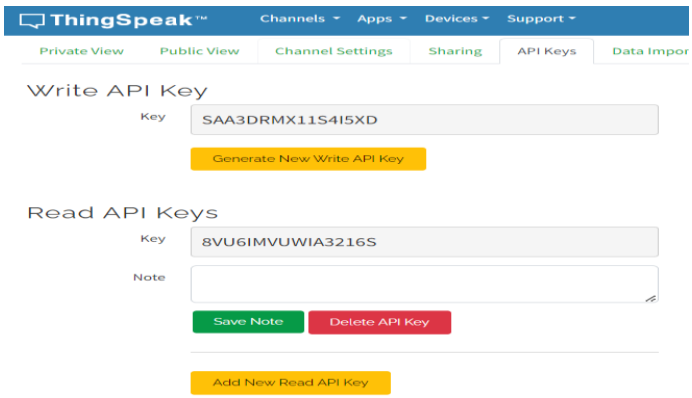
- Crear un canal: "My channels" y seleccionamos "New Channel".
- Configuración de canal: "Name= DATOS_OXÍGENO".
- Proporcionar descripción del canal creado, configurar datos de cada campo en este caso solo creamos uno que es el de oxígeno, cabe recalcar que ThingSpeaak

permite la creación de 8 campos por cada canal creado; en esta opción podemos crear etiquetas para identificar y organizar tus canales.

- Guardamos el canal creado.
- Obtenemos las claves “API Keys” para poder realizar las configuraciones en el IDE de Thonny tanto como de lectura o escritura como se requiera en el caso. Como nosotros vamos a realizar una escritura en la plataforma del sensor del oxígeno por ende utilizaremos la Write API Key: SAA3DRMX11S4I5XD

Figura 30

API Key de la plataforma ThingSpeak



Con la creación del canal donde se va a realizar la presentación de los datos a través de la nube se generan los API Keys, mismos que deberán colocarse en el código de Thonny para establecer comunicación para la transmisión de dato.

Figura 31

Configuración en Thonny para establecer comunicación con ThingSpeak

```
# Configuración de ThingSpeak
THINGSPEAK_URL = 'https://api.thingspeak.com/update'
THINGSPEAK_API_KEY = 'SAA3DRMX11S4I5XD'  # Reemplaza con tu clave de API de escritura
```

Con ello tendremos establecida la comunicación con ambas plataformas, en la codificación se genera bucle para verificación de comunicación entre plataformas y presentación del dato de oxígeno.

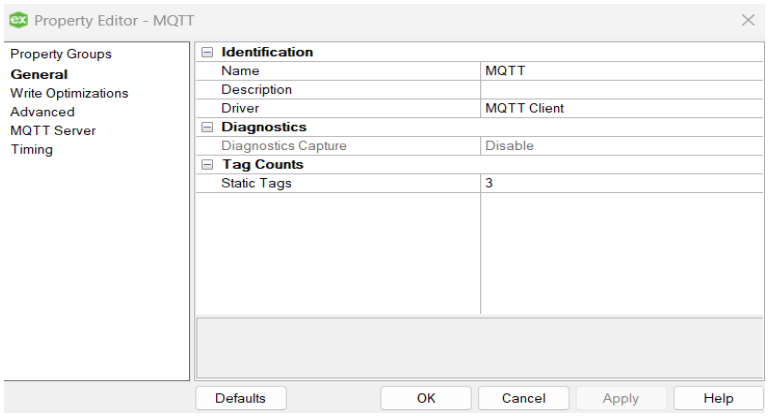
3.6 Configuración de KepServer

Utilizar KEPServerEX junto con MQTT permite no solo la recolección de datos de sensores y equipos industriales, sino también la transmisión de esta información a plataformas IoT para su análisis y visualización en tiempo real. MQTT Explorer es una aplicación de escritorio que ayuda a explorar y monitorear los tópicos de un Broker MQTT, haciendo más sencillo verificar y depurar la comunicación MQTT. A continuación, presentaremos los pasos de configuración para la comunicación entre sí:

- Primero creamos el canal en KepServer: “Name: MQTT” y “Drive: MQTT Client”.

Figura 32

Creación del canal KepServer

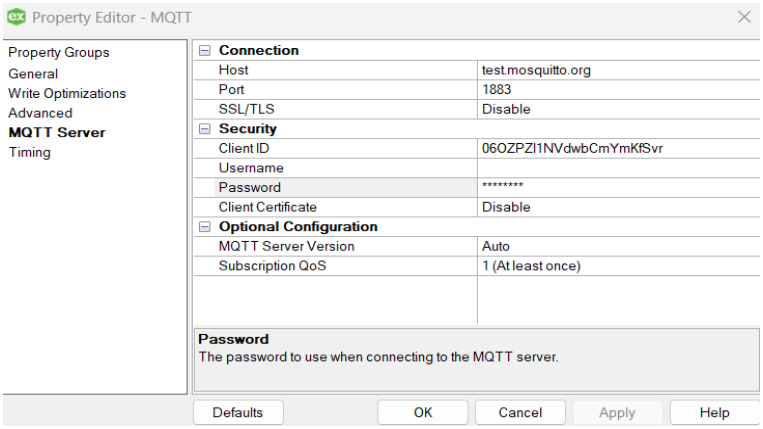


- Como siguiente paso procedamos a configurar el Drive, entre los parámetros importantes para la conectividad de la plataforma MQTT Explorer se debe tener

las mismas configuraciones como el Host, Port, Client ID y si en la aplicación MQTT le hemos puesto usuario y contraseña también deberíamos configurarlo en caso de tener y no configurarlo no se podrán conectar.

Figura 33

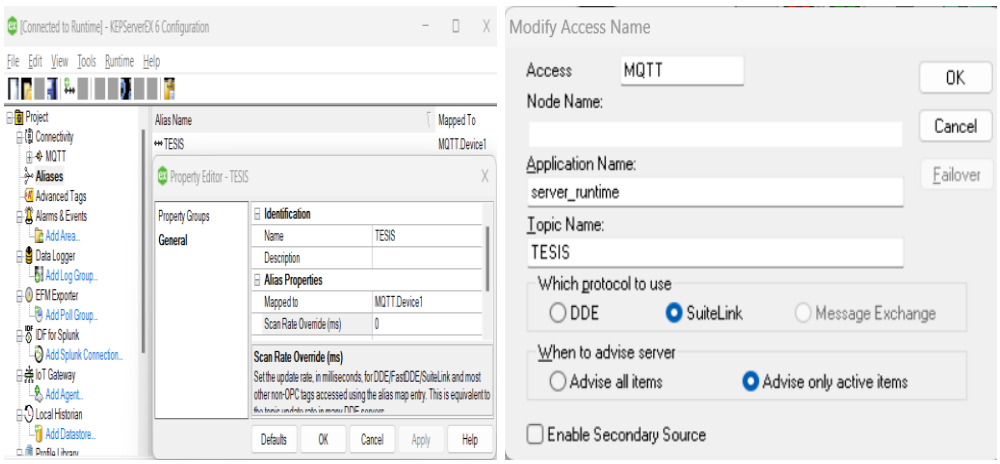
Configuración de Drive en KepServer



- Procedemos a configurar el Dispositivo y tags para suscripción y publicación.
- Para establecer comunicación desde Kepserver a Intouch creamos un Aliases y lo configuramos en modo MQTT, al realizar las configuraciones en intouch debe tener el nombre que se creó en KepServer para la correcta comunicación entre Software como se muestra a continuación:

Figura 34

Configuración en MQTT y Intouch para establecer comunicaciones



3.7 Pruebas y resultados

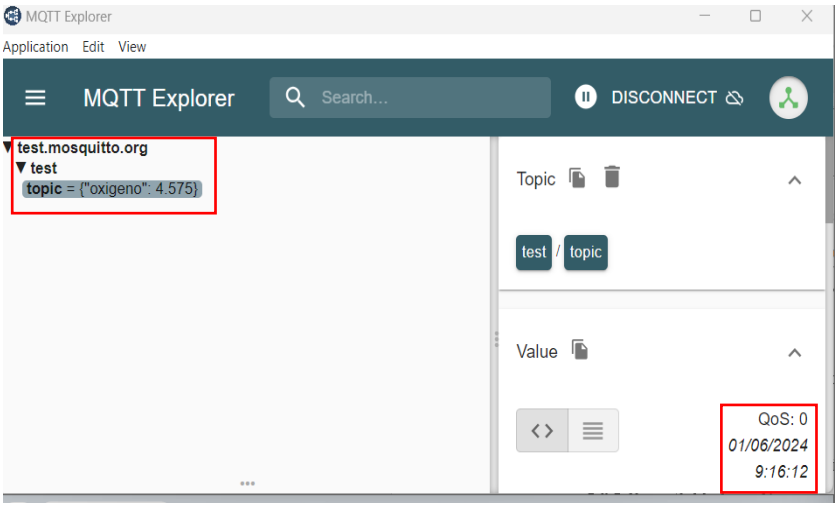
Para las pruebas realizadas, se presentarán los datos obtenidos por nuestro sistema implementado y datos obtenidos por el equipo de medición tradicional de OD, con ello se plantea mejorar la producción del camarón llevando un constante control del Oxígeno teniendo en cuenta que al tener este parámetro muy elevado también trae riesgo de enfermedades. Con ello, podemos supervisar y tomar las medidas que sean necesarias para mantener las condiciones adecuadas para mantener con un mayor porcentaje de supervivencia.

3.7.1 Prueba de tasa de envió mediante M*QTT*

Para realizar las pruebas de envió de datos mediante MQTT Explorer, implica enviar mensajes a través de un Broker MQTT, esto nos permitirá monitorear la tasa de mensajes para ver la frecuencia y el contenido de los mensajes recibidos, estos mensajes se podrán ver en la consola del Script en donde se podrá ver la cantidad de tiempo que tomo los mensajes y la tasa de envío calculada en mensajes por segundo. A continuación, se mostrará una prueba con hora y fecha la recepción de dato del OD.

Figura 35

Prueba en MQTT Explorer, recibiendo datos del sensor



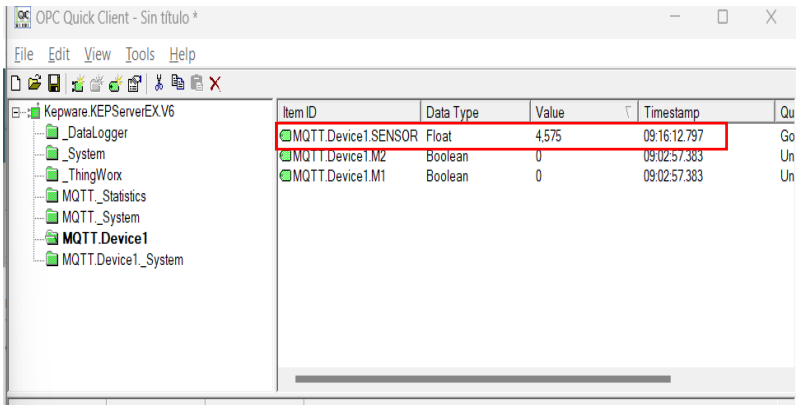
Podemos observar la hora, fecha y el valor de OD que recibe con ello este será nuestro Broker para la intercomunicación del resto de plataforma y transmisión para el monitoreo y tomas de decisiones para el sistema implementado.

3.7.2 Prueba de recepción del dato en KepServer

La recepción del valor en KepServer efectuando las configuraciones ya antes expuestas en la Figura 32 y 33, permite que se conecte al Broker del MQTT Explorer y poder obtener el dato, el mismo que nos ayudará para poder generar aliases para este valor de OD sea presentado en Intouch. Procedemos a presentar el valor reflejado y la hora de recepción que está siendo receptada a través del MQTT (figura 35). Se puede comprobar con la hora que esta presentando el mismo dato para luego llevar este valor al programa de monitoreo.

Figura 36

Prueba de recepción de dato en Kepserver a través de MQTT Explorer

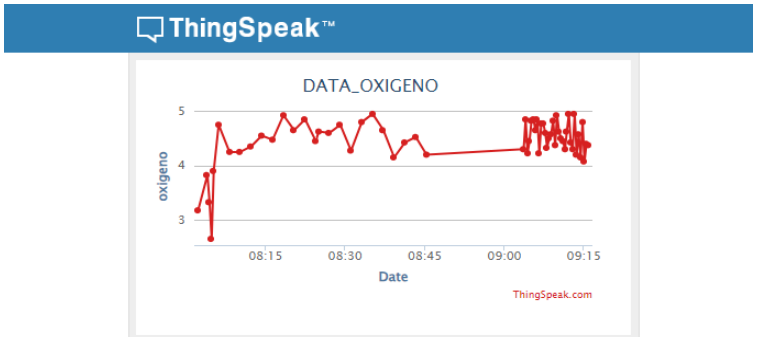


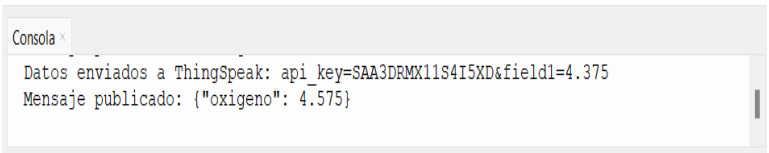
3.7.3 Prueba de recepción del dato en ThingSpeak y Thonny

Las pruebas realizadas en la recepción de datos obtenidos por Thonny son transmitidos por el ESP32, la transmisión de dotos llega primero a MQTT Explorer y ThingSpeak a través de la programación realizada en el IDE, para luego a través del MQTT puedan ser transmitidos al resto de Software para adquisición de base de datos y monitoreo que se va a realizar. Las pruebas se ven reflejado en la consola de Thonny y es transmitida a ThingSpeak con las configuraciones realizadas como en las Figuras 30 y 31

Figura 37

Publicación de datos del OD en ThingSpeak y consola de Thonny





Con ello, se puede observar que el dato que se está enviando es reflejado en todas las plataformas de manera correcta para su respectivo uso y configuraciones que se les dará según su aplicación.

3.7.4 Cálculos analíticos y valores reales de motores para la generación de OD

Para el cálculo de HP que se necesita según la densidad sembrada en las piscinas (110000lb) se procede a realizar los siguientes cálculos como se estableció en la Ecuación (1):

$$Requerimiento\ de\ HP = \frac{(1hp) * (libras\ estimadas\ )}{881lb} \quad (1)$$

$$Requerimiento\ de\ HP = \frac{(1hp) * (110000lb)}{881lb}$$

$$Requerimiento\ de\ HP = 124.85hp$$

$$Requerimiento\ de\ HP \approx 125hp$$

Al conocer la necesidad del valor de HP que necesita la piscina para la generación de OD debemos tomar los datos de los motores (16HP), cabe recalcar que esta generación de oxígeno también toma en cuenta parámetros físicos, químicos por ello los cálculos se realiza como aproximados. A continuación, realizamos el siguiente calculo para ver la necesidad de cuantos motores necesitamos:

Valor de motores necesarios

$$necesidad\ de\ motores\ requeridos = \frac{Requermiento\ de\ HP}{HP\ de\ motores\ a\ utlizar} \quad (2)$$

$$necesidad\ de\ motores\ requeridos = \frac{125HP}{16HP}$$

$$necesidad\ de\ motores\ requeridos = 7.81$$

$$necesidad\ de\ motores\ requeridos \approx 8$$

Con ello sabemos que se necesita un aproximado de 8 motores de 16HP para generar la potencia adecuada para la estimación de libras de camarones que se tiene. La piscina 10 de Satukin cuenta con 8 motores que se encuentran ubicados estratégicamente en zonas de mayor afluencia de camarón; para el cálculo de recuperación de OD se la realizó de manera experimental en el campo teniendo la necesidad de encender los motores de acuerdo con la siguiente tabla para recuperar el nivel de oxígeno para que se encuentre en su umbral (4mg/l a 7mg/l).

Tabla 14

Datos Receptados en campo por motores para la generación de OD

Oxígeno	Motores encendidos	Oxígeno	Motores encendidos
0,1	8	1,8	4
0,2	8	1,9	3
0,3	8	2	3
0,4	8	2,5	3
0,5	8	3	2
0,6	8	3,5	1
0,7	7	4	0
0,8	7	4,5	0
0,9	7	5	0
1	6	5,5	0
1,1	6	6	0
1,2	6	6,5	0
1,3	5	7	0
1,4	5	7,5	0
1,5	5	8	0
1,6	4	8,5	0
1,7	4	9	0

Cabe recalcar que las pruebas fueron realizadas con motores que tienen diferente tipo de vida útil apegándose a la realidad que generan de OD estos motores, con estos datos

podemos procesar el almacenamiento de la data y el entrenamiento con la información de la tabla para la simulación de encendidos de motores según el requerimiento estimado y consumos.

3.7.5 Almacenamiento de datos influxDB e IA

Almacenar datos históricos en InfluxDB, una base de datos de series temporales es una excelente manera de gestionar y consultar grandes volúmenes de datos generados en tiempo real. InfluxDB es especialmente útil para aplicaciones de monitoreo, IoT, y análisis de series temporales. Obtendremos un archivo (.csv) el cual será registrado cada valor en una lectura de 10 min, este mismo nos ayudará a ver cuándo el nivel de oxígeno baja para tomar las medidas de precauciones adecuadas y generar a través de la aire ación OD para mantener los umbrales adecuados. Cabe recalcar que esta prueba se realizó en la Piscina 10 de Satukin con la aireación mecánica apagada para poder obtener una lectura real del para metro de oxígeno por 24 horas, obteniendo los siguientes resultados:

Tabla 15

Lectura de OD mg/l 24 horas sin aireación encendida para pruebas

Oxígenos	Motores encendidos	Horario 1	Horario 2	Oxígenos	Motores encendidos	Horario 1	Horario 2	Oxígenos	Motores encendidos	Horario 1	Horario 2
1.5,1.4	5	0:00	0:05	1.2,1.2	7	8:00	8:05	6.5,6.5	0	16:00	16:05
1.4,1.4	5	0:10	0:15	1.2,1.2	7	8:10	8:15	6.4,6.4	0	16:10	16:15
1.4,1.5	5	0:20	0:25	1.2,0.9	7	8:20	8:25	6.3,6.4	0	16:20	16:25
1.5,1,4	5	0:30	0:35	1.8,1.8	4	8:30	8:35	6.2,6.1	0	16:30	16:35
1.5,1.5	5	0:40	0:45	1.8,1,8	4	8:40	8:45	6.0,6.1	0	16:40	16:45
1.5,1.5	5	0:50	0:55	1.8,1.8	4	8:50	8:55	6.0,5.8	0	16:50	16:55
1.5,1.5	5	1:00	1:05	2.5,2.6	2	9:00	9:05	5.8,5.8	0	17:00	17:05
1.5,1.5	5	1:10	1:15	2.6,2.7	2	9:10	9:15	5.6,5.7	0	17:10	17:15
1.5,1.4	5	1:20	1:25	2..7,2.6	2	9:20	9:25	5.5,5.3	0	17:20	17:25
1.5,1.4	5	1:30	1:35	2.6,2.6	2	9:30	9:35	5.2,5.2	0	17:30	17:35
1.5,1.4	5	1:40	1:45	2.6,2.6	2	9:40	9:45	5.0,4.8	0	17:40	17:45
1.5,1.5	5	1:50	1:55	2.6,2.6	2	9:50	9:55	4.8,4.8	0	17:50	17:55
1.3,1.4	5	2:00	2:05	2.6,2.6	2	10:00	10:05	4.6,4.5	0	18:00	18:05
1.3,1.3	5	2:10	2:15	2.7,2.8	2	10:10	10:15	4.6,4.5	0	18:10	18:15
1.3,1.3	5	2:20	2:25	2.9,3.0	2	10:20	10:25	4.6,4.5	0	18:20	18:25
1.3,1.3	5	2:30	2:35	4.0,4.1	0	10:30	10:35	4.6,4.5	0	18:30	18:35
1.3,1.3	5	2:40	2:45	4.0,4.1	0	10:40	10:45	4.6,4.5	0	18:40	18:45
1.4,1.5	5	2:50	2:55	4.1.4.0	0	10:50	10:55	4.3,4.2	0	18:50	18:55
1.5,1.5	5	3:00	3:05	4.1,4.3	0	11:00	11:05	4.2,4.2	0	19:00	19:05
1.5,1.5	5	3:10	3:15	4.4,4.8	0	11:10	11:15	4.1,4.2	0	19:10	19:15

1.5,1.5	5	3:20	3:25	5.0,5.1	0	11:20	11:25	4.2,4.1	0	19:20	19:25
1.5,1.3	6	3:30	3:35	5.2,5,3	0	11:30	11:35	4.0,4.2	0	19:30	19:35
1.3,1.2	6	3:40	3:45	5.4,5.6	0	11:40	11:45	4.2,4.1	0	19:40	19:45
1.1,1.1	6	3:50	3:55	6.0,6.1	0	11:50	11:55	4.0,4.0	0	19:50	19:55
1.1,1,1	6	4:00	4:05	6.2,6.4	0	12:00	12:05	4.0,4.0	0	20:00	20:05
1.1,1.1	6	4:10	4:15	6.5,6.5	0	12:10	12:15	4.2,4.1	0	20:10	20:15
1.1,1.1	6	4:20	4:25	6.7,6.8	0	12:20	12:25	4.0,3.8	0	20:20	20:25
1.1,1.1	6	4:30	4:35	7.0,7.1	0	12:30	12:35	3.1,2.8	2	20:30	20:35
1.1,1.1	6	4:40	4:45	7.1,7,1	0	12:40	12:45	2.9,3.0	2	20:40	20:45
1.1,1.1	6	4:50	4:55	7.0,6.9	0	12:50	12:55	3.1,2,8	2	20:50	20:55
1.1,1.0	6	5:00	5:05	7.0,7.1	0	13:00	13:05	2.9,3.0	2	21:00	21:05
1.1,1.2	6	5:10	5:15	7.1,7,1	0	13:10	13:15	3.1,2,9	2	21:10	21:15
1.2,1.2	6	5:20	5:25	7.0,7.0	0	13:20	13:25	3.1,2.8	2	21:20	21:25
1.3,1.3	5	5:30	5:35	7.0,7.1	0	13:30	13:35	2.9,3.0	2	21:30	21:35
1.3,1.3	5	5:40	5:45	7.1,7,1	0	13:40	13:45	3.1,2,8	2	21:40	21:45
1.3,1.3	5	5:50	5:55	7.0,7.0	0	13:50	13:55	2.9,3.0	2	21:50	21:55
1.3,1.3	5	6:00	6:05	6.8,6.8	0	14:00	14:05	2.6,2.4	3	22:00	22:05
1.3,1.3	5	6:10	6:15	6.5,6.6	0	14:10	14:15	2.4,2.4	3	22:10	22:15
1.3,1.3	5	6:20	6:25	6.5,6.5	0	14:20	14:25	2.6,2.4	3	22:20	22:25
1.6,1.6	4	6:30	6:35	6.4,6.4	0	14:30	14:35	2.4,2.4	3	22:30	22:35
1.6,1.7	4	6:40	6:45	6.5,6.5	0	14:40	14:45	2.2,2.1	3	22:40	22:45
1.7,1.7	4	6:50	6:55	6.4,6.4	0	14:50	14:55	1.8,1.9	3	22:50	22:55
1.8,1,8	4	7:00	7:05	6.5,6.5	0	15:00	15:05	2.0,2.1	4	23:00	23:05
1.7,1.7	4	7:10	7:15	6.4,6.4	0	15:10	15:15	2.0,1.8	4	23:10	23:15
1.7,1.7	4	7:20	7:25	6.5,6.5	0	15:20	15:25	2.0,1.8	4	23:20	23:25
1.7,1.7	4	7:30	7:35	6.5,6.5	0	15:30	15:35	2.0,1.8	4	23:30	23:35
1.7,1.7	4	7:40	7:45	6.5,6.5	0	15:40	15:45	1.7,1.7	4	23:40	23:45
1.7,1.7	4	7:50	7:55	6.5,6.5	0	15:50	15:55	2.0,1.8	4	23:50	23:55

3.7.6 Cálculo de consumo de aireación

Por lo general la aireación pasa encendida por 14 horas (desde 4pm a 6am), los 8 motores de 16HP con un consumo de 0.3gph según las especificaciones técnicas de los motores podemos realizar un cálculo aproximado de los gastos generados al mantener todos los aireadores encendidos. A continuación, utilizaremos la ecuación (3), para determinar el consumo en galones de un motor:

$$consumo\ total(gal) = (consumo\ especifico\ gph) * (tiempo\ de\ operacion(h)) \quad (3)$$

$$consumo\ total(gal) = (0.3) * (14)$$

$$consumo\ total = 4.2\ galones$$

Cada motor va a generar un consumo diario aproximadamente de 4.2galones y al mantener 8 motores encendidos generaría un total de:

Consumo total de 8 motores encendidos las 14H al día

$$consumo\ total\ (cabtidad\ de\ motores) = (consumo\ en\ gal) * (cantidad\ de\ motores) \quad (4)$$

$$consumo\ total\ (cabtidad\ de\ motores) = (4.2) * (8)$$

$$consumo\ total = 33.6 galones$$

Por los 8 motores encendidos se consumiría 33.6 galones diarios. A continuación, vamos a realizar los cálculos del consumo con los datos obtenidos en la tabla 15 donde según los estudios realizados no es necesario mantener las 14 horas los 8 aireadores para mantener el umbral adecuado el OD.

Primero vamos a calcular por horas los motores que deben estar encendidos:

$$consumo\ total(gal) = (consumo\ especifico\ gph) * (tiempo\ de\ operacion(h)) \quad (5)$$

Luego de ello calculamos por la cantidad de consumo de galones por los motores encendidos:

$$consumo\ total\ (cabtidad\ de\ motores) = (consumo\ en\ gal) * (cantidad\ de\ motores) \quad (6)$$

Se presentará el cálculo del primer horario para muestra y luego se presentará en una tabla los resultados de todos los consumos.

- Desde las 0:00 hasta las 3:30 permanecen encendidos 5 motores por 3 horas y media

$$consumo\ total(gal) = (consumo\ especifico\ gph) * (tiempo\ de\ operacion(h)) \quad (7)$$

$$consumo\ total(gal) = (0.3) * (3.5)$$

$$consumo\ total(gal) = 1.05 galones$$

$$consumo\ total\ (cabtidad\ de\ motores) = (consumo\ en\ gal) * (cantidad\ de\ motores) \quad (8)$$

$$consumo\ total\ (cabtidad\ de\ motores) = (1.05) * (5)$$

$$consumo\ total\ (cabtidad\ de\ motores) = 5.25 galones$$

Tabla 16

Consumo de combustible con lectura de sensor constante

Horarios	Tiempo en horas	Cantidad de motores encendidos	Consumo de combustible (gal)
0:00 – 3:30	3.5	5	5.25
3:30 – 5:30	2	6	3.6
5:30 – 6:30	1	5	1.5
6:30 – 8:00	1.5	4	1.8
8:00 – 8:30	0.5	7	1.7
8:30 – 9:00	0.5	4	0.6

9:00 – 10:30	1.5	2	0.9
20:30 – 22:00	1.5	2	0.9
22:00 – 23:00	1	3	0.9
23:00 – 24:00	1	4	1.2
		Total	17.72 gal

Al tener calculado procedemos a sumar los galones de consumo diario por hora, esta si estuviera automatizado el sistema de aireación para que se encienda cuando realmente es necesario

$$consumo\ total = 17.72 galones$$

Con estos resultados podemos ver la comparación:

- Consumo de motores encendidos 14 horas 8 motores = 33.6 galones
- Consumo de motores cuando es necesario= 17.72galones

Cálculo de diferencia de consumos con lectura de sensor

$$Cálculo\ de\ diferencia = 33.6 - 17.72\ galones \quad \textbf{(9)}$$

$$Cálculo\ de\ diferencia = 15.88\ galones$$

Diferencia porcentual de consumo

$$diferencia\ porcentual = \left(\frac{15.88}{33.6}\right) * 100 \quad \textbf{(10)}$$

$$diferencia\ porcentual = 47.26\%$$

Con estos resultados podemos observar que el ahorro de galones de diésel disminuirá en un 47.26%, el cual beneficiaría no solo en el ámbito de reducción de gastos, sino más bien generar el OD cuando realmente sea necesario. Ya que con las lecturas del sensor se observa que los motores son apagados en horarios que necesitan mantenerse encendidos por las bajas de oxígeno que presenta la piscina.

3.7.7 Diseño de sistema IA con datos generados por InfluxDB

Para la simulación de un entrenamiento en una IA se utilizó Google Colab, donde se implementó los valores de lectura obtenidos por InfluxDB (archivo.csv) y la cantidad de motores que deben encender según lo requerido por el sistema. Para el entrenamiento del sistema se utilizó SVM ya que su principal objetivo es encontrar un hiperplano en un espacio de alta dimensión que separa los datos de diferentes clases de manera óptima. A continuación, se presenta la programación para el entrenamiento que se realizó para la determinación de precisión de la IA con respecto a los datos reales ingresados.

- Generación de comandos para Google Colab y subir archivos recopilados desde InfluxDB datos de oxígeno y datos reales de cuantos motores deben estar encendidos según el valor de oxígeno.

```
import pandas as pd
from google.colab import files

# Subir los archivos CSV a Google Colab
print("Sube el archivo de datos de oxígeno")
uploaded_oxigeno = files.upload()
```

```
Sube el archivo de datos de oxígeno
```

oxigeno.csv(text/csv) - 128875 bytes, last modified: 25/5/2024 - 100% done

```
Saving oxigeno.csv to oxigeno.csv
```

```
print("Sube el archivo de datos de motores")
uploaded_motores = files.upload()
```

```
Sube el archivo de datos de motores
```

motores.csv(text/csv) - 30000 bytes, last modified: 25/5/2024 - 100% done

```
Saving motores.csv to motores.csv
```

- Al subir los archivos .CSV procedemos a leer la información y presentar los primeros 5 valores de cada una de las filas para verificar que se han leído correctamente.

```
# Leer los datos desde los archivos CSV
oxigeno_df = pd.read_csv(list(uploaded_oxigeno.keys())[0])
motores_df = pd.read_csv(list(uploaded_motores.keys())[0])
# Mostrar las primeras filas de los datos para verificar que se han leído correctamente
print("Datos de oxígeno:")
print(oxigeno_df.head())
```

```
Datos de oxígeno:
   9.185322488
0     4.203312
1     8.776629
2     8.955902
3     7.203317
4     1.669924
```

```
print("Datos de motores:")
print(motores_df.head())
```

```
Datos de motores:
   0
0  0
1  0
2  0
3  0
4  4
```

- Combinamos los datos en una dataframe y verificamos que se haya efectuado.

```
# Asegurarnos de que las columnas se llamen correctamente
oxigeno_df.columns = ['Oxigeno']
motores_df.columns = ['Motores']
# Combinar los datos en un solo DataFrame
data_df = pd.concat([oxigeno_df, motores_df], axis=1)
# Verificar los datos combinados
print("Datos combinados:")
print(data_df.head())
```

```
Datos combinados:
    Oxigeno  Motores
0  4.203312        0
1  8.776629        0
2  8.955902        0
3  7.203317        0
4  1.669924        4
```

- Como siguiente paso procedemos a llamar las librerías que utilizaremos para el entrenamiento SVC que se realizara con la data y en entrenamiento que le daremos al modelo.

```
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score,
classification_report
# Separar las características (X) y la variable objetivo (y)
X = data_df[['Oxigeno']].values
y = data_df['Motores'].values
# Dividir los datos en conjunto de entrenamiento y de prueba
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
# Crear y entrenar el modelo SVM
svm_model = SVC(kernel='linear')
svm_model.fit(X_train, y_train)
```

SVC

```
SVC(kernel='linear')
```

```
# Realizar predicciones
y_pred = svm_model.predict(X_test)
# Evaluar el modelo
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

- Con el entrenamiento realizado a 10000 datos tenemos como resultados la siguiente data tiene como exactitud del entrenamiento es de 99%, en el entrenamiento el parámetro que tiene menor precisión es de 95% cuando los parámetros de encendidos son 5 motores.

Figura 38

Resultados del entrenamiento de la IA

```
Accuracy: 0.9965
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1235
           1       0.99      1.00      1.00       103
           2       1.00      0.99      0.99        96
           3       1.00      1.00      1.00       208
           4       1.00      0.98      0.99        62
           5       0.95      1.00      0.97        53
           6       1.00      0.95      0.97        59
           7       0.97      1.00      0.98        65
           8       1.00      0.98      0.99       119

    accuracy                           1.00      2000
   macro avg       0.99      0.99      0.99      2000
weighted avg       1.00      1.00      1.00      2000
```

- Para observar de mejor manera los datos, vamos a realizar un gráfico donde tendremos mediante el entrenamiento SVM las predicciones vs los datos reales.

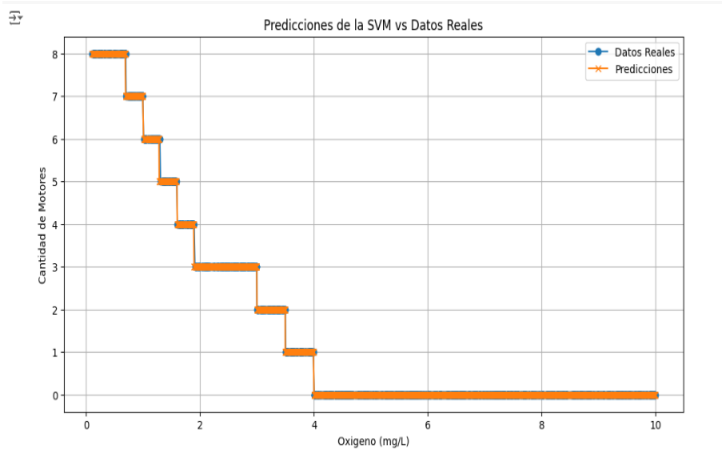
```
import matplotlib.pyplot as plt
# Ordenar los datos para una mejor visualización
sorted_indices = X_test.flatten().argsort()
X_test_sorted = X_test.flatten()[sorted_indices]
y_test_sorted = y_test[sorted_indices]
y_pred_sorted = y_pred[sorted_indices]

# Crear el gráfico
plt.figure(figsize=(12, 6))
plt.plot(X_test_sorted, y_test_sorted, label='Datos Reales',
marker='o')
plt.plot(X_test_sorted, y_pred_sorted, label='Predicciones',
marker='x')

# Añadir etiquetas y título
plt.xlabel('Oxigeno (mg/L)')
plt.ylabel('Cantidad de Motores')
plt.title('Predicciones de la SVM vs Datos Reales')
plt.legend()
plt.grid(True)
plt.show()
```

- Como resultado podemos observar que los datos reales y las predicciones siguen el mismo patrón, teniendo este tipo de gráficas podemos observar que el método de entrenamiento escogido es el correcto. Con ello, al generar el método de

predicción después del entrenamiento puede tomar las decisiones de encendido y apagado de motores de aireación y serán muy acertadas.

Figura 39

Entrenamiento SVM

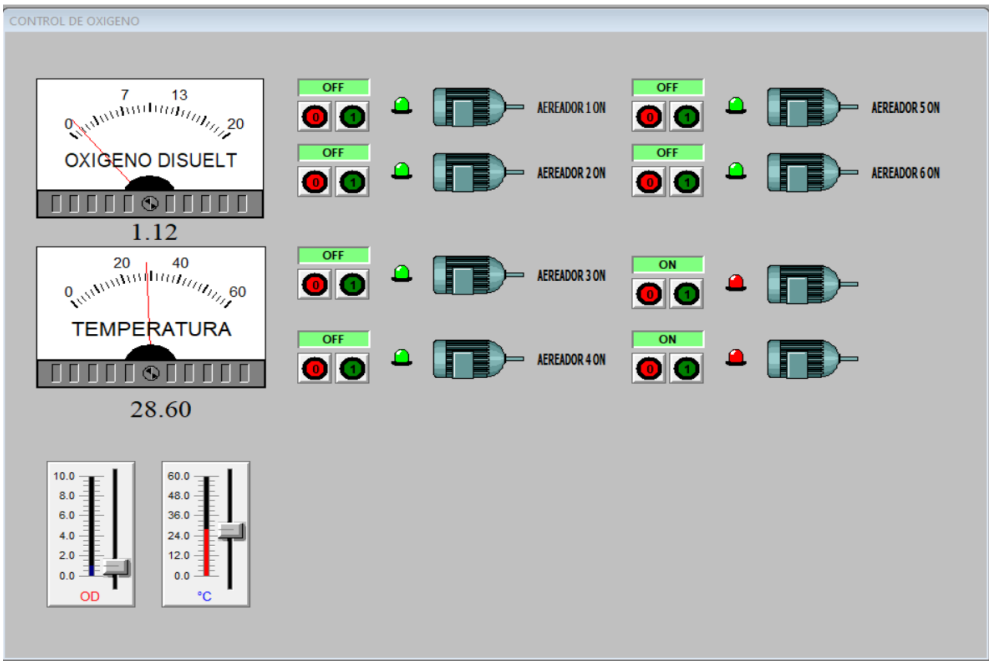


3.*7.8* Prueba de sistema SCADA en Intouch

Con los datos obtenidos determinamos según la necesidad de motores de aireación que se encenderán por la baja de oxígeno en la piscina se estableció de la siguiente manera. Para la primera prueba se captura de la data en el horario de entre las 3:50 a 5:00 am (tabla 15) donde el oxígeno disuelto esta entre los 1 a 1.1 mg/l por lo cual lo establecido en la tabla de encendido deberían encenderse 6 motores por lo cual esto debería reflejar en la aplicación Intouch, la misma que se presentara a continuación:

Figura 40

Scada Intouch lecturas



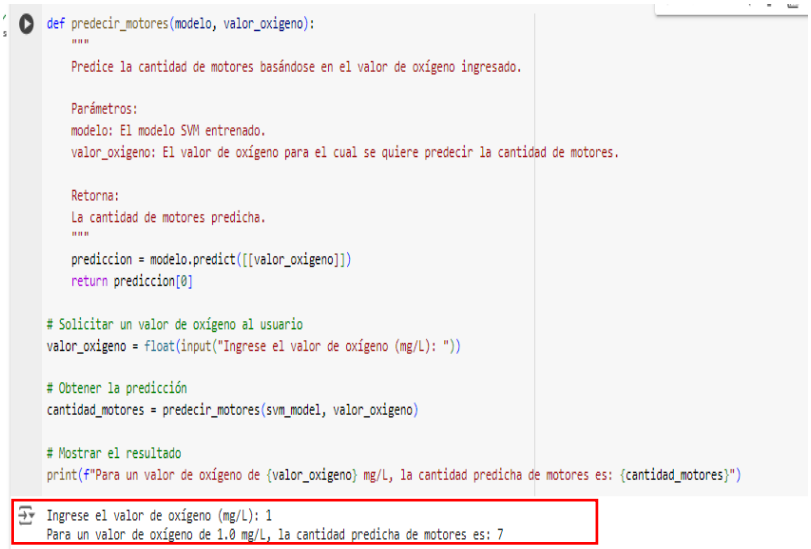
Como se observa en la imagen al recibir el dato del oxígeno que se encuentra en 1.1mg/l están encendidos los 6 motores y según el valor de oxígeno que ingrese al valor de lectura se activaran por las condiciones dadas en las tablas y configuraciones.

3.7.9 Inteligencia Artificial aplicada en aireación (data de entrenamiento y simulación)

Teniendo los resultados del entrenamiento, podemos asignar este entrenamiento para poder realizar las predicciones del control de motores a encender, en esta misma IA se generó para que nos dé el cálculo de motores que deben encenderse dependiendo del valor de OD que tenga la piscina como se muestra en la siguiente imagen:

Figura 41

Resultado del entrenamiento, toma de decisiones



```
def predecir_motores(modelo, valor_oxigeno):
    """
    Predice la cantidad de motores basándose en el valor de oxígeno ingresado.

    Parámetros:
    modelo: El modelo SVM entrenado.
    valor_oxigeno: El valor de oxígeno para el cual se quiere predecir la cantidad de motores.

    Retorna:
    La cantidad de motores predicha.
    """
    prediccion = modelo.predict([[valor_oxigeno]])
    return prediccion[0]

# Solicitar un valor de oxígeno al usuario
valor_oxigeno = float(input("Ingrese el valor de oxígeno (mg/L): "))

# Obtener la predicción
cantidad_motores = predecir_motores(svm_model, valor_oxigeno)

# Mostrar el resultado
print(f"Para un valor de oxígeno de {valor_oxigeno} mg/L, la cantidad predicha de motores es: {cantidad_motores}")
```

```
Ingrese el valor de oxígeno (mg/L): 1
Para un valor de oxígeno de 1.0 mg/L, la cantidad predicha de motores es: 7
```

.

CONCLUSIONES

- Al implementar un sistema de monitoreo de oxígeno ofrece una solución integral para garantiza la recopilación de datos, de una manera confiable y eficiente que contribuye a la prevención de la supervivencia del camarón.
- El diseño de una red de comunicación para la integración de compatibilidad entre sensores, controladores y actuadores es fundamental para impulsar la evolución y la eficacia de los sistemas electrónicos, brindando una plataforma sólida para la automatización inteligente y la optimización de procesos en diversos sectores industriales y aplicaciones prácticas.
- El desarrollo de un modelo matemático para evaluar el rendimiento energético de los aireadores en piscinas automatizadas representa un paso crucial hacia la eficiencia y la sostenibilidad en la gestión de instalaciones acuáticas. Este enfoque proporciona una herramienta poderosa para la toma de decisiones informadas y la implementación de medidas de ahorro de energía, contribuyendo así a la preservación de recursos y la reducción de costo.
- Las tecnologías de transmisión de datos como Modbus TCP/IP y el protocolo MQTT, que son fundamentales para la comunicación efectiva entre los sensores y los sistemas de control en los sistemas de acuicultura. Estos protocolos aseguran una transmisión de datos fiable y en tiempo real.
- El trabajo detalla el uso de diversas herramientas y tecnologías, como Thonny, MQTT Explorer, InfluxDB y KepServer, que son esenciales para la adquisición, procesamiento y visualización de datos. La integración de estas herramientas facilita un monitoreo y control precisos del entorno acuícola.
- El presente trabajo resalta la importancia de utilizar tecnologías avanzadas y metodologías de investigación sólidas para optimizar la producción de camarones, mostrando que la combinación de IA, IoT y sistemas SCADA puede llevar a una gestión más eficiente y rentable en la acuicultura.

.

RECOMENDACIONES

- Se debe asegurar que el sistema de monitoreo de oxígeno sea robustos y confiables, utilizando sensores de alta precisión que garanticen la recopilación de datos de manera continua y precisa. Esto contribuirá significativamente a la prevención de la mortalidad de los camarones y a la optimización de su crecimiento.
- Proporcionar capacitaciones continuas al personal en el uso y gestión de las redes de comunicación, asegurando que estén familiarizados con las últimas tecnologías y protocolos.
- Basado en los resultados del modelo, se debe implementar medidas concretas para el ahorro de energía, optimizando el uso de los aireadores y reduciendo los costos operativos.
- Realizar un monitoreo continuo de la red de comunicación para detectar y corregir posibles fallos o interferencias que puedan afectar la transmisión de datos.
- Fomentar la adopción de tecnologías avanzadas y metodologías de investigación sólidas para optimizar la producción de camarones.
- Analizar los requisitos de comunicación de los diferentes componentes electrónicos (sensores, controladores, actuadores) y selecciona el protocolo de comunicación adecuado (como Modbus TCP/IP, MQTT, etc.)
- Realizar un estudio detallado del sistema de aireación de las piscinas, incluyendo los componentes involucrados, el consumo eléctrico y los factores que afectan al rendimiento energético.

REFERENCIAS

AIX 7.1. (2023, marzo 24). https://www.ibm.com/docs/es/aix/7.1?topic=protocol-tcpip-protocols

Arcia, D. L. (s. f.). *Comparación de la supervivencia y crecimiento del camarón blanco del Pacífico en agua fertilizada con urea y nitrato de sodio en Zamorano, Honduras*.

Bajaña Araujo, J. A., & Junqui Sudario, J. J. (2018). *Diseño de un sistema de oxigenación y alimentación automática de larvas de camarón para aumentar la producción de la Empresa Marbeth S.A.* [bachelorThesis, Espol]. http://www.dspace.espol.edu.ec/handle/123456789/47437

Carbajal Hernández, J. J., & Sánchez Fernández, L. P. (2013). Diagnóstico y predicción del hábitat en la camaronicultura. *Computación y Sistemas*, *17*(3), 435-455.

Carchipulla Leal, V. M. (2018). *Importancia del oxígeno disuelto para mejorar la calidad de agua en estanques de camarón blanco litopenaeus vannamei*. http://repositorio.utmachala.edu.ec/handle/48000/12905

Chuya Zhangallimbay, D. B., & Landívar Zambrano, J. (2021). *Criterios para el dimensionamiento de la aeración en la producción de un cultivo de camarón blanco (Penaeus vannamei)* [Thesis, ESPOL. FIMCM: Acuicultura]. http://www.dspace.espol.edu.ec/handle/123456789/55412

comercial. (2020, octubre 13). Protocolo MQTT. *Ditel Diseños y Tecnología S.A.* https://www.ditel.es/protocolo-mqtt/

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

Folleto del agua—Trabajo de investgacion—Folleto Informativo Oxígeno Disuelto (OD) ¿Qué es el—Studocu. (s. f.). Recuperado 1 de junio de 2024, de https://www.studocu.com/bo/document/universidad-indigena-boliviana-aymara-tupak-katari/arte-cultura/folleto-del-agua-trabajo-de-investgacion/40139101

Lombeida Vásconez, J. N., & Samaniego Reyes, J. J. (2022). *Desarrollo de un tablero demostrativo para el Control de Sistemas de aireadores y alimentadores supervisados y monitoreados mediante radiofrecuencia por medio de Sistema Scada* [bachelorThesis]. http://dspace.ups.edu.ec/handle/123456789/22834

López, J. F. (2021). *DISEÑAR E IMPLEMENTAR UN SISTEMA DE OXIGENACIÓN PARA CRIADERO DE PECES HACIENDO USO DE CONTROLADOR EN EL PERÍODO ABRIL - SEPTIEMBRE 2021*. http://dspace.tecnologicosudamericano.edu.ec/jspui/handle/123456789/366

Ltd, S. W. T. C. (s. f.). *Oxygen sensor for dissolved oxygen—Oxygen sensors—Diribo—Find and compare products*. Diribo.com. Recuperado 4 de junio de 2024, de https://www.diribo.com/en/product/oxygen-sensor-for-dissolved-oxygen

Modbus: Qué es y cómo funciona | Comunicaciones Industriales. (2020, mayo 5). https://www.cursosaula21.com/modbus-que-es-y-como-funciona/

Pérez-López, E. (2015). Los sistemas SCADA en la automatización industrial. *Revista Tecnología en Marcha*, *28*(4), 3. https://doi.org/10.18845/tm.v28i4.2438

Sierra, C. A., & Maroso, J. J. (2019). *Estrategias de manejo para mejorar la producción piscícola en un sistema de jaulas flotantes con Tilapia roja Oreochromis sp, en Montería, Córdoba*.

TCP (Transmission Control Protocol): Retrato del protocolo de transporte. (2020, marzo 2). IONOS Digital Guide. https://www.ionos.com/es-us/digitalguide/servidores/know-how/que-es-tcp-transport-control-protocol/

ANEXOS

Anexo 1: Código Thonny

```
import network
import time
import json
from machine import UART, Pin
from umodbus.serial import Serial
from umodbus.client.serial import ModbusSerialClient as ModbusClient
from umqtt.simple import MQTTClient, MQTTException
import urequests

# Configura las credenciales de la red WiFi
SSID = "Redmi Note 9"
PASSWORD = "11111111"

# Configuración MQTT
MQTT_BROKER = 'test.mosquitto.org'
MQTT_PORT = 1883
MQTT_CLIENT_ID = 'esp32_client'
MQTT_TOPIC = 'test/topic'
MQTT_USER = ''
MQTT_PASSWORD = ''

# Configuración de ThingSpeak
THINGSPEAK_URL = 'https://api.thingspeak.com/update'
THINGSPEAK_API_KEY = 'SAA3DRMX11S4I5XD'  # Reemplaza con tu clave de API de escritura

# Configuración Modbus RTU
```

```
MODBUS_BAUDRATE  = 9600
MODBUS_ADDRESS  = 1  # Dirección del dispositivo Modbus
MODBUS_REGISTER  = 0  # Registro de lectura

# Configuración del pin para DE/RE del MAX485
DE_RE_PIN = 2  # GPIO2

# Función para conectar a WiFi
def connect_wifi(ssid, password):
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    wlan.connect(ssid, password)

    while not wlan.isconnected():
        print('Conectando a la red WiFi...')
        time.sleep(1)

    print('Conectado a WiFi:', wlan.ifconfig())

# Callback de MQTT
def mqtt_callback(topic, msg):
    print('Mensaje recibido en el tópico %s: %s' % (topic, msg))

# Función para conectar a MQTT
def connect_mqtt():
    try:
        client = MQTTClient(MQTT_CLIENT_ID, MQTT_BROKER,
port=MQTT_PORT, user=MQTT_USER, password=MQTT_PASSWORD)
        client.set_callback(mqtt_callback)
        client.connect()
```

```
        print('Conectado al broker MQTT')
        return client
    except MQTTException as e:
        print('Error al conectar al broker MQTT:', e)
        return None

# Función para publicar mensajes en MQTT
def publish_mqtt(client, topic, msg):
    if client:
        try:
            client.publish(topic, msg)
            print('Mensaje publicado: %s' % msg)
        except MQTTException as e:
            print('Error al publicar mensaje:', e)
    else:
        print('Cliente MQTT no conectado. No se puede publicar el mensaje.')

# Función para leer el sensor Modbus
def read_modbus_sensor():
    uart = UART(1, baudrate=MODBUS_BAUDRATE,  tx=17, rx=16)
    de_re_pin = Pin(DE_RE_PIN, Pin.OUT)
    client = ModbusClient(uart, de_pin=de_re_pin, re_pin=de_re_pin)

    try:
        client.connect()
        response = client.read_input_registers(MODBUS_ADDRESS,
MODBUS_REGISTER,  1)
        client.close()
        if response:
            oxigeno = response[0] / 100.0  # Ajusta según la resolución de tu sensor
```

```
            return oxigeno
        else:
            print("Error al leer el sensor Modbus")
            return None
    except Exception as e:
        print('Error al leer el sensor Modbus:', e)
        return None

# Función para enviar datos a ThingSpeak
def send_to_thingspeak(oxigeno):
    try:
        datos = 'api_key={}&field1={}'.format(THINGSPEAK_API_KEY, oxigeno)
        response = urequests.post(THINGSPEAK_URL, data=datos)
        print('Datos enviados a ThingSpeak:', datos)
        response.close()
    except Exception as e:
        print('Error al enviar datos a ThingSpeak:', e)

# Conectar a WiFi
connect_wifi(SSID, PASSWORD)

# Conectar a MQTT
mqtt_client = connect_mqtt()

if mqtt_client:
    try:
        while True:
            # Leer el sensor de oxígeno vía Modbus
            sensor_oxigeno = read_modbus_sensor()
```

```
        if sensor_oxigeno is not None:
            # Crear un diccionario con el valor del sensor de oxígeno
            datos_sensor = {
                "oxigeno": sensor_oxigeno
            }

            # Convertir el diccionario a una cadena JSON
            mensaje_json = json.dumps(datos_sensor)

            # Publicar el mensaje JSON como una cadena en MQTT
            publish_mqtt(mqtt_client, MQTT_TOPIC, mensaje_json)

            # Enviar el dato a ThingSpeak
            send_to_thingspeak(sensor_oxigeno)

        # Esperar 10 minutos antes de enviar el siguiente valor
        time.sleep(600)
except KeyboardInterrupt:
    print("Programa detenido por el usuario.")
except Exception as e:
    print(f"Ocurrió un error: {e}")
```

Anexo 2: Practica de instalación y mantenimiento del sensor

INSTALACIÓN DEL SISTEMA AUTÓNOMO DE OD EN PISCINA

Objetivo

El objetivo de esta práctica es aprender a instalar y configurar un sensor de oxígeno disuelto en una piscina de acuicultura, así como entender los pasos necesarios para su mantenimiento regular.

Materiales Necesarios

- Sensor de oxígeno disuelto (por ejemplo, un sensor KNF-103b).
- Módulo de comunicación RS485 (como MAX485).
- ESP32 u otro microcontrolador compatible.
- Fuente de alimentación.
- Cables y conectores.
- Computadora con software de programación (Thonny, MQTT Explorer, etc.).
- Soluciones de calibración para el sensor de oxígeno.
- Herramientas de instalación (destornilladores, alicates, etc.).
- Materiales de fijación (abrazaderas, soportes, etc.).

Instalación del Sensor de Oxígeno

- Elige un lugar adecuado en la piscina (compuerta de salida) donde se colocará el sensor. Debe estar sumergido a una profundidad de 30 cm del nivel total de la piscina para obtener las lecturas representativas.
- Instalación del sistema autónomo de energía panel solar, controlador y batería.
- Asegúrate de que el lugar seleccionado permita un fácil acceso para mantenimiento y calibración.
- Fija el sensor al soporte seleccionado utilizando abrazaderas o soportes. Asegúrate de que esté firmemente sujeto y no se mueva con el flujo de agua.

Figura 1

Instalación del sistema de alimentación y colocación del sensor

- Conecta los pines del módulo RS485 al microcontrolador (ESP32), RX del MAX485 al TX del ESP32, TX del MAX485 al RX del ESP32, y el pin de control DE/RE al GPIO2 del ESP32(como se muestra en la Tabla 13).
- Conecta la fuente de alimentación al ESP32 y asegúrate de que todo esté correctamente alimentado.

Figura 2

Conexión de puertos Esp32, Max485 y Sensor Knf-103b

Configuración del Sistema

- Programa el ESP32 utilizando el siguiente código para leer los datos del sensor de oxígeno y enviarlos a un servidor MQTT y ThingSpeak (Anexo 1)

Mantenimiento y Limpieza del Sensor

- Realiza una calibración regular del sensor de oxígeno utilizando las soluciones de calibración recomendadas por el fabricante.
- Sumerge el sensor en la solución de calibración y ajusta los parámetros según las instrucciones del fabricante.
- Registra y ajusta los valores de calibración en el sistema de monitoreo.
- Revisa el sensor semanalmente para asegurarte de que no esté obstruido o sucio.

- Limpia el sensor con un paño suave y agua destilada. No utilices productos químicos agresivos que puedan dañar el sensor.
- Verifica la integridad de los cables y conectores.

Verificación de Datos

- Compara regularmente los datos del sensor con mediciones manuales para verificar su precisión.
- Ajusta y calibra el sensor si los datos son inconsistentes o si se observa una desviación significativa.

Figura 3

verificación del funcionamiento del sensor Knf-103b

Conclusión

La instalación y mantenimiento adecuados de un sensor de oxígeno disuelto son cruciales para garantizar la salud y el crecimiento óptimos de los camarones en una piscina de acuicultura. A través de este procedimiento, se puede lograr una monitorización precisa y confiable, mejorando la eficiencia y reduciendo costos operativos.

Recomendaciones Finales

- **Monitoreo Continuo**: Implementa un sistema de monitoreo continuo para detectar cualquier anomalía en tiempo real.
- **Mantenimiento Regular**: Establece un calendario de mantenimiento y calibración para asegurar el rendimiento del sensor.

- **Capacitación**: Capacita al personal en la correcta instalación, calibración y mantenimiento del sensor de oxígeno.
- **Análisis de Datos**: Utiliza herramientas de análisis de datos para identificar tendencias y optimizar las condiciones del agua en la piscina.