

**UNIVERSIDAD NACIONAL DE CAAGUAZU  
FACULTAD DE CIENCIAS Y TECNOLOGIAS  
CARRERA DE INGENIERÍA EN ELECTRONICA**



**IMPLEMENTACIÓN DE UN SISTEMA WEB PARA EL  
MONITOREO Y CONTROL AUTOMATIZADO DE ESTANQUES  
PARA CRIANZA DE ALEVINES EN EL CENTRO PISCÍCOLA  
NACIONAL DE EUSEBIO AYALA, CORDILLERA**

**CESAR RAMÓN ROJAS MERELES**

Coronel Oviedo - Paraguay  
Año 2023

**UNIVERSIDAD NACIONAL DE CAAGUAZU  
FACULTAD DE CIENCIAS Y TECNOLOGIAS  
CARRERA DE INGENIERÍA EN ELECTRONICA**

**IMPLEMENTACIÓN DE UN SISTEMA WEB PARA EL MONITOREO Y  
CONTROL AUTOMATIZADO DE ESTANQUES PARA CRIANZA DE  
ALEVINES EN EL CENTRO PISCÍCOLA NACIONAL DE EUSEBIO  
AYALA, CORDILLERA**

**Elaborado por**

**CESAR RAMON ROJAS MERELES**

**Tutor**

**Prof. Ing. Derlis Rodrigo Arredondo Onieva**

Trabajo presentado a la Facultad de Ciencias y  
Tecnología de la Universidad Nacional de Caaguazú,  
como requisito para la obtención del título de Ingeniero en  
Electrónica

**Coronel Oviedo - Paraguay  
Año 2023**



## Atribución-NoComercial 4.0 Internacional (CC BY-NC 4.0)

Usted es libre de:

- **Compartir** — copiar y redistribuir el material en cualquier medio o formato
- **Adaptar** — remezclar, transformar y construir a partir del material

Bajo los siguientes términos:

- **Atribución** — Usted debe dar [crédito de manera adecuada](#), brindar un enlace a la licencia, e [indicar si se han realizado cambios](#). Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.
- **NoComercial** — Usted no puede hacer uso del material con [propósitos comerciales](#).

## DERECHO DE AUTOR

Quien suscribe **CESAR RAMÓN ROJAS MERELES**, autor del trabajo de investigación titulado **“IMPLEMENTACIÓN DE UN SISTEMA WEB PARA EL MONITOREO Y CONTROL AUTOMATIZADO DE ESTANQUES PARA CRIANZA DE ALEVINES EN EL CENTRO PISCÍCOLA NACIONAL DE EUSEBIO AYALA, CORDILLERA”**, declara que voluntariamente cede a título gratuito en forma pura y simple ilimitada e irrevocablemente a favor de la Facultad de Ciencias y Tecnologías – UNCA, el derecho de autor de contenido patrimonial, que le corresponde sobre el trabajo de referencia. Conforme a lo anteriormente expresado, esta sesión le otorga a la FCyT la Facultad de comunicar la obra divulgarla, publicarla y reproducirla en soportes analógicos o digitales en la oportunidad que así lo estime conveniente. La FCyT deberá indicar qué autoría o creación del trabajo corresponde a mi persona y hará referencia al autor y a las personas que hayan colaborado en la realización del presente trabajo de investigación.

En la ciudad de Coronel Oviedo a los .... , del mes de ..... del 2024

.....

Firma

## PÁGINA DE APROBACIÓN

Trabajo de fin de grado para la obtención del Título de Ingeniero en Electrónica aprobado en representación de la Facultad Ciencias y Tecnología de la Universidad Nacional de Caaguazú, por el Tribunal Examinador constituido por los siguientes profesores y con la siguiente nota final:

-----  
Prof. Ing.

-----  
Prof. Ing.

-----  
Prof. Ing.

Acta Nro.:-----

Fecha:-----

Calificación:-----

**Dedicado a:**

Quisiera dedicar esta tesis a mi familia, amigos y mentores, quienes me han dado su apoyo incondicional y me han ayudado a llegar hasta donde estoy hoy. Agradezco a ellos por su apoyo y por permitirme aspirar a objetivos más altos. Estoy muy agradecido por el amor, la preocupación y el apoyo que me han proporcionado para que logre mis sueños.

## **Agradecimientos:**

En primer lugar, doy gracias a Dios por todas las bendiciones que me han permitido llegar aquí. En segundo lugar, dedico mi agradecimiento a todas aquellas personas que han contribuido de alguna manera para que esta tesis se lograra.

A mis profesores, quienes me orientaron durante todo el proceso y me ayudaron a perfeccionar mi trabajo. También deseo agradecer a mis compañeros y amigos, quienes me apoyaron y me motivaron a seguir trabajando.

Por último, quiero expresar mi gratitud a mi familia, que me impulsó a continuar luchando hasta alcanzar mis metas académicas. Esta tesis es el resultado de una dedicación y esfuerzo constantes durante mucho tiempo, y nunca olvidaré a todos aquellos que ayudaron a que fuera posible.

# IMPLEMENTACIÓN DE UN SISTEMA WEB PARA EL MONITOREO Y CONTROL AUTOMATIZADO DE ESTANQUES PARA CRIANZA DE ALEVINES EN EL CENTRO PISCÍCOLA NACIONAL DE EUSEBIO AYALA, CORDILLERA

Cesar Ramón Rojas Mereles

## RESUMEN

Este trabajo final de grado en Ingeniería Electrónica se centra en la implementación de un sistema web para el monitoreo y control automatizado de estanques para la crianza de alevines en el Centro Piscícola Nacional de Eusebio Ayala, Cordillera. El objetivo principal es mejorar los procesos de control y monitoreo de los estanques y la gestión de los datos recolectados, con el fin de optimizar la gestión de recursos, aumentar la productividad y mejorar la calidad de los productos del centro piscícola. Este trabajo se justifica por la necesidad de obtener datos en tiempo real sobre el estado de los estanques de crianza de alevines, proporcionando información precisa para la toma de decisiones acertadas. En el proceso de recolección y análisis de datos sobre las prácticas de crianza de alevines en el centro piscícola, se emplearon métodos de investigación tanto cuantitativos como cualitativos. Estas técnicas e instrumentos se utilizaron con el objetivo de obtener la información más precisa y detallada posible. Este enfoque integral permitió una comprensión profunda de las operaciones actuales y las áreas de mejora potencial. La ingeniería de diseño del proyecto pasó por varias etapas críticas. Inicialmente, se realizó una exhaustiva búsqueda de posibles alternativas de solución, considerando una variedad de enfoques y tecnologías. Posteriormente, se definieron criterios claros y objetivos para evaluar estas alternativas. Estos criterios se utilizaron para realizar una evaluación rigurosa y seleccionar los elementos que cumplían en mayor medida con los requisitos establecidos.

Finalmente, se desarrolló un sistema que no solo es técnicamente sólido y capaz de cumplir con los objetivos del proyecto, sino que también es económicamente factible. Este sistema representa una solución efectiva y eficiente para mejorar las operaciones del centro piscícola, optimizando el proceso de crianza de alevines y permitiendo una gestión más eficaz de los recursos.

**Palabras clave:** alevines, aplicación web, automatización, centro piscícola, control, monitoreo

# IMPLEMENTATION OF A WEB SYSTEM FOR THE AUTOMATED MONITORING AND CONTROL OF PONDS FOR BREEDING OF FRYING IN THE NATIONAL FISHERY CENTER OF EUSEBIO AYALA, CORDILLERA

Cesar Ramón Rojas Mereles

## SUMMARY

This final degree project in Electronic Engineering focuses on the implementation of a web system for the automated monitoring and control of ponds for raising fingerlings at the National Fish Center of Eusebio Ayala, Cordillera. The main objective is to improve the control and monitoring processes of the ponds and the management of the data collected, in order to optimize resource management, increase productivity and improve the quality of the products of the fish farming center. This work is justified by the need to obtain real-time data on the state of the fry rearing ponds, providing accurate information for making correct decisions. In the process of collecting and analyzing data on fry rearing practices at the fish farm, both quantitative and qualitative research methods were used. These techniques and instruments were used with the aim of obtaining the most accurate and detailed information possible. This comprehensive approach allowed for a deep understanding of current operations and areas for potential improvement. The project design engineering went through several critical stages. Initially, an exhaustive search for possible alternative solutions was carried out, considering a variety of approaches and technologies. Subsequently, clear and objective criteria were defined to evaluate these alternatives. These criteria were used to carry out a rigorous evaluation and select the elements that most closely met the established requirements. Finally, a system was developed that is not only technically sound and capable of meeting the project objectives, but is also economically feasible. This system represents an effective and efficient solution to improve the operations of the fish farming center, optimizing the fry raising process and allowing more effective management of resources.

**Keywords:** fingerlings, web application, automation, fish farming center, control, monitoring

# CONTENIDO

DERECHO DE AUTOR.....	iv
PÁGINA DE APROBACIÓN.....	v
Dedicado a:.....	vi
Agradecimientos:.....	vii
RESUMEN.....	viii
SUMMARY.....	ix
CONTENIDO.....	x
ÍNDICE DE FIGURAS.....	xiii
ÍNDICE DE TABLAS.....	xiv
I. INTRODUCCIÓN.....	1
II. MARCO TEÓRICO.....	2
1. CAPITULO 1: CONCEPTOS BÁSICOS.....	2
1.1. ANTECEDENTES.....	2
1.2. ESTADO DEL ARTE.....	7
1.3. DEFINICIÓN DE TÉRMINOS BÁSICOS.....	7
2. CAPÍTULO 2: CONCEPTOS GENERALES SOBRE ACUICULTURA Y PISCICULTURA.....	9
2.1. Acuicultura.....	9
2.2. Piscicultura.....	9
2.2.1. Estanques.....	10
2.3. Variables de interés en sistema de acuicultura.....	11
2.3.1. Temperatura.....	11
2.3.2. Oxígeno disuelto.....	11
2.3.3. PH.....	11
2.3.4. Turbidez del agua.....	12
3. CAPÍTULO 3: CONCEPTOS GENERALES SOBRE SISTEMAS DE CONTROL Y DESARROLLO WEB.....	13
3.1. Control en Lazo Cerrado.....	13
3.2. Control en Lazo Abierto.....	13
3.3. Elementos de un Sistema de Control.....	14
3.3.1. Interfaces Hombre Maquina.....	14

3.3.2.	Sensores o transductores .....	14
3.3.3.	Actuadores .....	14
3.3.4.	Comunicaciones digitales .....	14
3.3.5.	Controladores Automáticos.....	15
3.4.	Desarrollo web .....	17
3.4.1.	HTML.....	17
3.4.2.	CSS .....	17
3.4.3.	JavaScript.....	17
3.4.4.	JSON.....	18
3.4.5.	PHP .....	18
3.4.6.	Bootstrap .....	18
3.4.7.	Servidor web.....	19
3.4.8.	Bases de datos .....	20
III.	RESUMEN EJECUTIVO.....	21
4.	CAPITULO 4 .....	21
4.1.	Descripción del trabajo .....	21
4.2.	Tipo de Investigación .....	21
4.3.	Técnicas e instrumentos de recolección de Datos.....	22
4.4.	Métodos y análisis de datos .....	22
4.5.	Justificación.....	23
4.6.	Objetivos .....	23
4.6.1.	Objetivo general.....	23
4.6.2.	Objetivos específicos .....	23
4.7.	Fases metodológicas .....	24
IV.	DISEÑO DE INGENIERÍA .....	27
5.	CAPITULO 5 .....	27
5.1.	Descripción del trabajo .....	27
5.2.	Variables claves para el monitoreo.....	27
5.3.	Criterios para la selección del controlador, los sensores y actuadores .....	28
5.4.	Evaluación y selección de controladores.....	29
5.5.	Especificaciones técnicas de los elementos seleccionados.....	31
5.5.1.	Sensor de TDS: .....	31
5.5.2.	Oxígeno disuelto en el agua: .....	32
5.5.3.	PH del agua: .....	33

5.5.4.	Módulo de relé para Arduino:.....	35
5.5.5.	Calentador de agua: .....	36
5.5.6.	Bomba de agua: .....	36
5.5.7.	Arduino Mega 2560 .....	37
5.5.8.	Ethernet shield:.....	38
5.5.9.	Alimentador de peces: .....	39
5.6.	Arquitectura del Sistema .....	40
5.7.	Interfaz gráfica HMI .....	41
5.7.1.	Descripción de la interfaz gráfica .....	41
5.8.	Funcionamiento y programación del hardware .....	44
V.	ANÁLISIS ECONÓMICO.....	50
6.	CAPITULO 6.....	50
6.1.	Lista de materiales y costos .....	50
6.2.	Indicadores financieros .....	51
6.3.	Flujo de caja.....	52
6.4.	Análisis de sensibilidad .....	53
6.4.1.	Escenario optimista .....	53
6.4.2.	Escenario pesimista.....	54
6.5.	Conclusión de factibilidad.....	54
VI.	CONCLUSIONES Y RECOMENDACIONES .....	55
VII.	BIBLIOGRAFÍA.....	56
VIII.	ANEXO.....	59

## ÍNDICE DE FIGURAS

Figura 1: Acuicultura, crías en ambiente acuático. ....	9
Figura 2: Piscicultura intervención tecnológico en cría de animales acuático .....	10
Figura 3: diagrama de bloque de un controlador Automático industrial [7].....	16
Figura 4: Sensor de temperatura.....	31
Figura 5: Sensor de turbidez .....	32
Figura 6: Sensor de oxígeno disuelto en el agua .....	33
Figura 7: Sensor de pH del agua.....	34
Figura 8: Módulo relé de cuatro canales.....	35
Figura 9: Calentador de agua.....	36
Figura 10: Bomba de agua .....	37
Figura 11: Arduino Mega 2560 .....	37
Figura 12: Ethernet shield para Arduino .....	38
Figura 13: Alimentador de peces.....	39
Figura 14: Esquema de bloques completo del sistema.....	40
Figura 15: Diagrama de archivos de la aplicación web.....	41
Figura 16: Interfaz de inicio de sesión .....	42
Figura 17: Interfaz principal de monitoreo y control .....	43
Figura 18: Interfaz de visualización de grafico de temperatura vs. Tiempo.....	44
Figura 19: Diagrama de conexión entre Arduino Mega y sensor de temperatura. ....	45
Figura 20: Diagrama de conexión entre Arduino Mega y sensor de PH del agua. ....	46
Figura 21: Diagrama de conexión entre Arduino Mega y sensor de Turbidez del agua.46	
Figura 22: Diagrama de conexión entre Arduino Mega y sensor de oxígeno disuelto en el agua.....	47
Figura 23: Diagrama de conexión entre Arduino Mega y calentador de agua.....	47
Figura 24: Diagrama de conexión entre Arduino Mega y bomba de agua. ....	48
Figura 25: Diagrama de conexión entre Arduino Mega y alimentador de peces. ....	48
Figura 26: Esquema completo de la propuesta .....	49

## ÍNDICE DE TABLAS

Tabla 1: Peces comunes y rangos de temperatura [4].....	11
Tabla 2: Efecto de pH en estanques de peces [4]. .....	12
Tabla 3: Evaluación de alternativas de control .....	30
Tabla 4: Lista de materiales .....	50
Tabla 5: Costo del sistema .....	51
Tabla 6: Resumen de rendimiento anual de alevines incubados .....	51
Tabla 7: Detalles de alevines .....	51
Tabla 8: Cantidad de alevines para comercialización .....	52
Tabla 9: Detalles de pérdidas de alevines .....	52
Tabla 10: Flujo de caja en el escenario moderado .....	53
Tabla 11: Flujo de caja en el escenario optimista .....	54
Tabla 12: Flujo de caja en el escenario pesimista .....	54

## **I. INTRODUCCIÓN**

El trabajo presentado aborda el tema de la implementación de un sistema web para el monitoreo y control automatizado de estanques para crianza de alevines en el Centro Piscícola Nacional de Eusebio Ayala, Cordillera. La motivación para realizar este trabajo es mejorar los procesos de control y monitoreo de los estanques, así como la gestión de los datos recolectados, a fin de lograr una mejor gestión de los recursos, incrementar la productividad y mejorar la calidad de los productos que el centro piscícola produce. El objetivo principal del trabajo es implementar un sistema web para el monitoreo y control automatizado de los estanques, que permita la recolección de datos en tiempo real para su posterior gestión y análisis. El sistema web será capaz de recolectar los datos y generar los reportes necesarios para tener un mejor control de los estanques. La justificación para el trabajo expuesto se basa en la necesidad de obtener datos en tiempo real sobre el estado de los estanques de crianza de alevines, para obtener información precisa para la toma de decisiones acertadas. Además, el trabajo cubrirá las necesidades de monitoreo y control de los estanques de crianza de alevines, sin la implementación de ningún otro sistema adicional.

El trabajo se estructura en nueve secciones, siendo la más amplia el marco teórico, que se desglosa en varios capítulos. El primer capítulo ofrece una introducción al tema, presentando los antecedentes de la automatización de estanques, el estado del arte y la definición de términos básicos. El segundo capítulo aborda conceptos sobre acuicultura y piscicultura, así como las variables ambientales asociadas. El tercer capítulo expone conceptos generales sobre sistemas de control, controladores, sensores y actuadores. El cuarto capítulo se centra en los controladores electrónicos y sus diferentes tipos. Finalmente, el quinto capítulo describe los lenguajes de programación y el desarrollo web, proporcionando los elementos necesarios para el diseño de una aplicación web.

Las secciones restantes presentan la justificación del proyecto, los objetivos y el marco metodológico. La sección seis se dedica al desarrollo de la ingeniería de diseño, seguida de un análisis económico y, finalmente, las conclusiones y recomendaciones del trabajo.

## II. MARCO TEÓRICO

### 1. CAPITULO 1: CONCEPTOS BÁSICOS

#### 1.1. ANTECEDENTES

Dentro de los antecedentes, destacan diversos proyectos ejecutados. Uno de ellos es el estudio “SISTEMA DE MONITOREO REMOTO DE ACUICULTURA EN ESTANQUES PARA LA CRIANZA DE CAMARONES”, publicado en 2018 por la Revista Chilena de Ingeniería. Esta investigación fue realizada por Susana Flores Mollo y Diego Aracena Pizarro de la Universidad Ingeniería de Tarapacá, Chile. Su contribución principal fue la creación de un sistema de monitoreo remoto que automatiza la recolección de datos de las variables físicas del agua. Algunas conclusiones derivadas de este trabajo incluyen:

- Se desarrolló un sistema de monitoreo con acceso remoto de las condiciones del agua de los estanques y del ambiente de un criadero de camarones. El software entrega herramientas que permiten al usuario visualizar datos del ambiente a través del tiempo, con el fin de analizar el comportamiento y prevenir condiciones críticas del medio acuático en el que vive el camarón de río.
- Para llevar a cabo el proyecto no sólo se utilizaron herramientas del área de la informática, sino que también se interactuó con profesionales de acuicultura y mecánicos, con el objetivo de desarrollar una aplicación real y no simulada.

El trabajo titulado “DISEÑO DE UN SISTEMA AUTOMATIZADO PARA LA PRODUCCIÓN PISCÍCOLA EN EL INSTITUTO SUPERIOR TECNOLÓGICO TSA´CHILA” publicado en la Revista Científica Multidisciplinar Ciencia Latina, de Ecuador, en el año 2023. Dicho trabajo consistió en la presentación de un diseño automatizado para oxigenación y alimentación adecuada evitando pérdidas en el transcurso de la crianza. Las principales conclusiones del trabajo fueron las siguientes:

- Se determinaron las variables clave a controlar para la producción piscícola eficiente, incluyendo el nivel de oxígeno (4-5 ppm), el pH del agua (6.5-7), la temperatura del agua (26-32°C), y las condiciones de limpieza.
- Se desarrolló un sistema de alertas visuales y audibles que indica anomalías en la temperatura, oxígeno y pH del agua, utilizando programación local y sensores con tecnología de sistemas embebidos de Arduino.
- Se utilizaron conocimientos bibliográficos, empíricos y prácticos para recopilar la información necesaria y desarrollar el algoritmo de operación y los programas aplicativos con tecnologías de Arduino.

Otro trabajo titulado “IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO PARA EL CONTROL DE TEMPERATURA DEL AGUA Y ALIMENTACIÓN DE ALEVINES UTILIZANDO UNA PLATAFORMA CON SOFTWARE LIBRE” fue realizado por Víctor Alfonso Sánchez Villacis y presentado en la Universidad Regional Autónoma de los Andes (Uniandes), Facultad de Sistemas Mercantiles, Ecuador, en 2017, como requisito para obtener el título de Ingeniero en Sistemas e Informática. Este proyecto se centró en la creación de un software diseñado para regular la temperatura del agua y garantizar una alimentación eficaz de los alevines en tiempos específicos, con el fin de potenciar la producción y reducir las pérdidas debidas a una alta tasa de mortalidad. Las conclusiones principales derivadas de esta investigación son:

- En base a la investigación realizada se determinó que los principales parámetros que permiten una buena condición del ambiente de producción de alevines son: la temperatura y la alimentación los mismos que fueron los principales actores del desarrollo (SACTA).
- Se diseñó un aplicativo web que facilita el acceso a la información de las mediciones obtenidas desde un módulo electrónico que conecta los sensores que reposan en el estanque y que obtienen los parámetros de la calidad del agua. Así como también se diseñó un Módulo electrónico que es capaz de interconectar sensores y componentes Arduino que facilitan la obtención de las mediciones de

los parámetros de la calidad del agua las mismas que se almacenan en una base de datos para su posterior manipulación.

El estudio titulado "DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO DE OXIGENACIÓN DEL AGUA PARA EL CRIADERO ACUÍCOLA VALLE DEL MAR UBICADO EN SANTA MARTA" fue conducido por Pedro Daniel Contreras Aristibal y Michael Andrés Pérez Vado. Este trabajo se presentó en la Facultad de Ingeniería de Santa Marta, Colombia, en diciembre de 2019, como parte de los requisitos para obtener el título de Ingeniero Electrónico. La principal contribución de este proyecto fue proponer una solución al problema identificado, satisfaciendo la demanda existente y mejorando la calidad de vida de los peces. Las conclusiones clave de esta investigación son:

- Se logró identificar las variables fisicoquímicas a controlar para la regulación de los niveles del agua y el nivel de oxígeno que se requiere en el Criadero Acuícola Villa del Mar.
- Con el desarrollo de este proyecto, se confirma la veracidad de que es factible diseñar e implementar un sistema automatizado, capaz de oxigenar las 24 horas el agua de los estanques del criadero Villa del Mar a través del método de aireación del agua en la que habitan las tilapias en caso de fallas, avisando al operario y así evitar grandes mortalidades. Por lo tanto, con este prototipo se dan por cumplido los objetivos de la investigación.

En otro estudio, titulado "SOLUCIÓN IoT (EL INTERNET DE LAS COSAS) PARA LA OPTIMIZACIÓN DEL PROCESO DE PISCICULTURA EN EL CENTRO DE DESARROLLO AGROALIMENTARIO EL LIMONAL", realizado por Lisbeth Haydee Rivera Betancur, se presentó como requisito para obtener el título de Magíster en Redes y Sistemas de Comunicación en la Universidad Santo Tomás, División de Ingenierías y Arquitectura, Bucaramanga, en mayo de 2020. La principal innovación de este proyecto fue la capacidad de monitorear, a distancia y en tiempo real, variables esenciales del proceso. Esto permitió tomar decisiones rápidas que mejoran

los resultados en la multiplicación artificial de peces. Las conclusiones fundamentales obtenidas de esta investigación son:

- Se seleccionaron los parámetros más relevantes de dicho proceso en la granja, que logran determinar la calidad del agua para que los peces estén en un entorno que no sea mortal, esto se hizo bajo un modelo de Internet de las Cosas, teniendo en cuenta lo revisado en documentos científicos.
- Se logró plantear un framework conceptual basado en el Internet de las Cosas para la optimización del desempeño de las variables del proceso productivo seleccionado, mediante la identificación de sus características tecnológicas como tipos de sensores, transmisión, almacenamiento de datos.
- Se diseñó la arquitectura de solución tecnológica IoT para la optimización de un proceso productivo en el Centro de Desarrollo Agroalimentario El Limonal, basándose en las capas del modelo de referencia en la que se incluyen la capa de dispositivo, la capa de red y la capa de soporte y aplicaciones.

El estudio titulado "DISEÑO DE UN MÓDULO ELECTRÓNICO PARA LA CRIANZA AUTOMATIZADA DE PECES MEDIANTE MODELAMIENTO MATEMÁTICO MULTIPARAMÉTRICO QUE SIMULE LAS CONDICIONES BÁSICAS NECESARIAS PARA LA CRIANZA, EN ESTANQUES ARTIFICIALES EN FUNCIÓN DE PARÁMETROS FISICOQUÍMICOS" fue conducido por Hernán Díaz Lopez y Yezid Vargas Gómez. Este trabajo fue publicado en la Revista UIS Ingenierías en el año 2018. Las principales contribuciones de este proyecto fueron proponer un modelo matemático para reducir los costos de implementación y desarrollar un sistema de automatización eficiente para la crianza de peces. Las conclusiones clave de esta investigación son:

- Se identificaron los costos de instrumentación, proponiendo trabajar con un número reducido de elementos y utilizando un modelo matemático para estimar o correlacionar variables, reduciendo así los costos de implementación sin comprometer el proceso.

- Se diseñó un sistema de automatización que ahorra energía y establece periodos de trabajo óptimos para los dispositivos, controlando algunos parámetros de la calidad del agua en función de otros.
- La recopilación de la mayor cantidad de datos posibles y la validación de diferentes modelos antes de seleccionar el definitivo son esenciales para obtener precisión y reducir errores en el sistema.
- Se analizaron y ajustaron cuatro modelos matemáticos a la realidad de los procesos piscícolas en la región Guanentá, culminando en un único modelo que integra las mejoras y prioriza las variables más representativas, reduciendo así la instrumentación necesaria.
- El régimen de alimentación constante mejora la conversión del alimento en masa corporal, recomendando su redefinición cada 60 a 90 días para optimizar el proceso.
- Los perfiles de crecimiento de los peces están influenciados por la época del año, sugiriendo la monitorización de parámetros específicos como TGC y FTGC en lugar de SGR para una caracterización más precisa.
- Se describieron y verificaron varios modelos en casos experimentales reales, demostrando la capacidad de los modelos matemáticos para simular escenarios y optimizar la productividad sin medir parámetros innecesarios.
- Los resultados muestran las ventajas de los modelos matemáticos para predecir la dinámica de múltiples variables durante el ciclo de producción, estableciendo condiciones iniciales óptimas para mejorar la productividad de los proyectos piscícolas.
- A pesar de la falta de datos experimentales diarios para algunas variables, los modelos matemáticos demostraron una alta capacidad para representar procesos reales en piscicultura, destacando la necesidad de seguir mejorando y ampliando el modelo con más datos experimentales.

## **1.2. ESTADO DEL ARTE**

La automatización se refiere al proceso mediante el cual máquinas o sistemas automatizados llevan a cabo tareas sin requerir intervención humana directa ni supervisión constante. Esta área ha experimentado un crecimiento notable, con innovaciones tecnológicas y técnicas de control emergiendo y aplicándose en campos diversos como la industria, medicina, acuicultura y aeronáutica, entre otros. Esta evolución no se limita únicamente a la replicación y mecanización de acciones humanas; ahora, las máquinas poseen capacidades inteligentes para desempeñar funciones específicas.

En el contexto de la acuicultura, la automatización proporciona datos valiosos, permitiendo a los acuicultores tener una comprensión más profunda de cómo sus insumos influyen en el crecimiento de peces y camarones bajo diferentes condiciones. Acá, los sensores desempeñan un papel crucial, recolectando datos que posteriormente alimentan algoritmos avanzados, optimizando la toma de decisiones automatizadas.

El acceso a información en tiempo real sobre parámetros fisicoquímicos y biológicos del sistema de producción ofrece al administrador de la piscifactoría una visión sin precedentes de las condiciones físicas y biológicas. Este nivel de detalle sería inviable con sistemas de monitoreo manual, dada la labor intensiva necesaria para recolectar, registrar datos y luego elaborar gráficos e informes. Así, queda claro que la acuicultura automatizada ofrece considerables beneficios, permitiendo producciones más eficientes y sostenibles en términos de costos. La integración de tecnología en la acuicultura ha dejado de ser una alternativa para convertirse en algo necesario.

## **1.3. DEFINICIÓN DE TÉRMINOS BÁSICOS**

- **Control:** Es el conjunto de actividades ejercidas en forma continua por el abastecedor con el objetivo de verificar que la calidad del agua suministrada al estanque cumpla con lo establecido.
- **Carga:** podemos definir la carga orgánica del agua como el contenido de compuestos de carbono en un efluente. Dichos compuestos de carbono son

estructura química donde el carbono esta enlazado a hidrogeno y otros elementos como azufre, oxigeno, nitrógeno, fosforo y cloro

- **Equipos:** son instrumento de medición que sirve para determinar la densidad relativa de los líquidos sin necesidad de calcular ante su masa y volumen
- **Sistema:** Sistema acuícola es el cultivo de organismos acuáticos, en particular peces, moluscos y planta acuáticas. La actividad de cultivo implica la intervención del hombre en el proceso de cría para aumentar la producción, en operaciones como la siembra, la alimentación y la protección de depredadores.
- **Sistema de control automatizado para la cría de peces:** Tecnología que se utiliza para monitorear y controlar los parámetros ambientales necesarios para el buen crecimiento de los peces. Estos sistemas incluyen sensores que vigilan los parámetros del agua, alimentación automatizada, control de temperatura, iluminación, oxígeno, filtración y otros.
- **Sensores:** Dispositivos de medición que se utilizan para monitorear parámetros ambientales, como la temperatura, la iluminación y el oxígeno, para garantizar el bienestar de los peces.
- **Sistemas recirculantes:** Sistemas que permiten un uso eficiente de los recursos, minimizando al mismo tiempo la contaminación del entorno. Estos sistemas requieren una configuración inicial adecuada, vigilancia continua y mantenimiento para garantizar el éxito de la producción.
- **Monitoreo en tiempo real:** Es el proceso de recopilar y analizar datos para detectar cambios en los parámetros del agua o en el comportamiento de los peces. Esto permite que los criadores actúen rápidamente ante situaciones imprevistas.

## 2. CAPÍTULO 2: CONCEPTOS GENERALES SOBRE ACUICULTURA Y PISCICULTURA

### 2.1. Acuicultura

La acuicultura es el cultivo o crianza de recursos hidrobiológicos (plantas acuáticas y peces). El objetivo es incrementar la producción de la especie ya sea en ambientes naturales o artificiales. Es uno de los sectores de alimentación de más rápido crecimiento y representa aproximadamente el 50% del pescado consumido a nivel mundial. Así, la acuicultura está superando a la pesca en rendimiento. Por lo tanto, la demanda creciente de pescado de la población deberá ser atendida solamente por la piscicultura [1]. Algunos ejemplos referentes a la acuicultura se presentan a continuación en la figura 2.1.

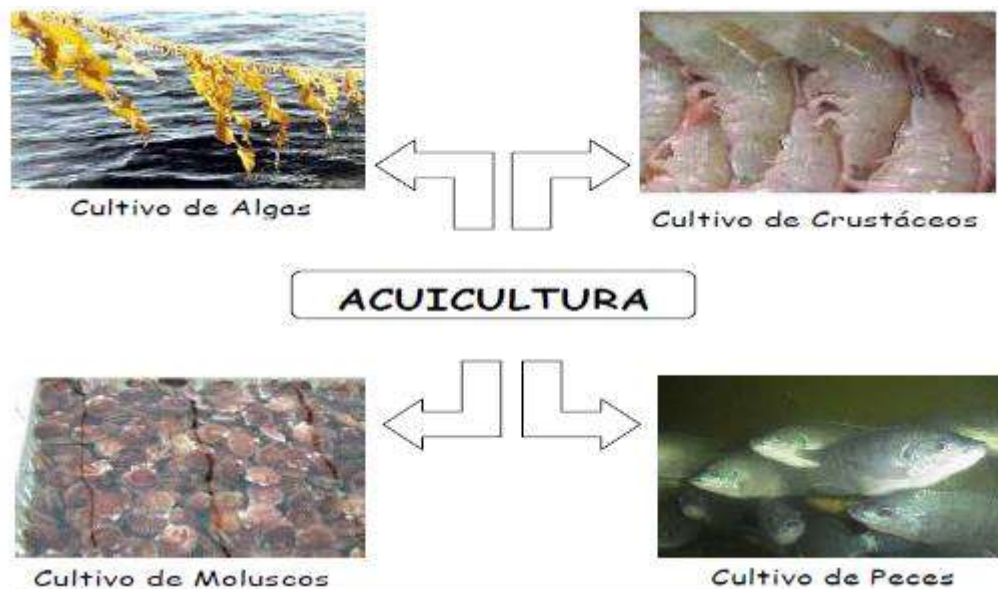


Figura 1: Acuicultura, crías en ambiente acuático.

### 2.2. Piscicultura.

“La piscicultura, en sentido riguroso de la palabra, no es otra cosa que la cría y multiplicación de los peces, verifíquense estas en plena libertad o en depósitos de agua cerrados. Por lo tanto, el arte de la piscicultura está basado en la fecundación natural o artificial y en la incubación de los huevos del pescado, con el objeto de obtener crías o

alevines que se engordan o engruesan para que sirvan de alimento al hombre [2]. A continuación se puede visualizar en el ejemplo de la figura 2.2 algunos tipos de pisciculturas.



Figura 2: Piscicultura intervención tecnológico en cría de animales acuático

### 2.2.1. Estanques

El estanque es un recinto acuático poco profundo (1.0-2.5m) con entrada y salida de agua controladas, construido para cultivar organismos acuáticos. Se lo confunde comúnmente con piscina, laguna y tajarar. En cuanto a la forma, la mayoría de los cuerpos de agua pueden ser útiles para acuicultura si las condiciones ambientales son las adecuadas, pero los cuerpos de agua rectangulares ofrecen varias ventajas, principalmente para el manejo hidráulico y de la cosecha.

Los estanques pueden ser construidos de diferentes materiales, los cuales están relacionados con el tipo de suelo local y la disponibilidad de elementos de construcción al alcance del productor [3]. A continuación, se presentan ejemplos de estanques construidos en Paraguay con diversos materiales disponibles en el país:

## 2.3. Variables de interés en sistema de acuicultura

### 2.3.1. Temperatura

Los peces son organismos poiquiloterms cuya temperatura corporal depende del medio en que viven. Por lo tanto la temperatura es la variable más importante y determinante para el cultivo, y a su vez la más difícil de controlar. Cada especie posee un rango óptimo para crecer y desarrollarse [3]. Los peces son ectotérmicos, esto quiere decir que su temperatura corporal depende del ambiente en el que se encuentren, por eso las condiciones de temperatura del acuario deben mantenerse dentro de los niveles tolerados por los organismos, los cambios bruscos pueden llegar a causar estrés provocando enfermedades; cada especie tiene un rango o intervalo óptimo de temperatura en el cual puede crecer rápidamente. El cuadro siguiente se puede apreciar algunos peces de acuario comunes y sus respectivas temperaturas. Conocer las variaciones de la temperatura a lo largo del día, así como de una estación a otra, permitirá decidir la especie a cultivar y determinar el tipo de manejo a realizar.

[4].

Especie	Nombre común	Temperatura °C
<i>Tilapia mossambica</i>	Tilapia	25-35
<i>Osphronemus goramy</i>	Goumrami	24-28
<i>Puntius javanicus</i>	Tawes	25-33
<i>Cyprinus carpio</i>	Carpa común	20-25
<i>Ctenopharyngodon idellus</i>	Carpa herbívora	25-30
<i>Anguilla japonica</i>	Anguila	20-28

Tabla 1: Peces comunes y rangos de temperatura [4]

### 2.3.2. Oxígeno disuelto.

La cantidad de oxígeno disuelto en el agua es limitante para la sobrevivencia de los peces. Los encargados de producir oxígeno en un estanque son el fitoplancton y las plantas acuáticas. Esta producción tiene variantes a lo largo del día, siendo alta durante las horas de luz solar y mínima antes del amanecer.

[3].

### 2.3.3. PH

El pH se expresa a través de una escala que va de 0 a 14. El valor 7 corresponde a un agua neutra, si su pH es inferior a este valor es ácida, y de ser superior es alcalina [3]. La mayoría de los tanques de agua dulce oscilan entre los 6 y 9 de pH un rango aceptable para el cultivo de peces, siendo una variable donde depende de la temperatura. La influencia directa del pH en los estanques de peces se puede observar en la Tabla 2.2., puede haber excepciones, como por ejemplo algunos peces en el río Amazonas viven y se reproducen en un agua con pH de 4 a 4.5. Es importante que se conozca no sólo el valor de pH, sino la estabilidad o inestabilidad del mismo, ya que cambios bruscos de pH son perjudiciales para las especies presentes en el cultivo. [4].

pH	Efectos
4	Punto ácido de muerte
4-5	No reproducción
5-6	Crecimiento lento
6-9	Mejor crecimiento
9-11	Crecimiento lento
11	Punto alcalino de muerte

Tabla 2: Efecto de pH en estanques de peces [4].

#### **2.3.4. Turbidez del agua**

La turbidez detecta la presencia de partículas colgantes en el agua. [5]

---

### **3. CAPÍTULO 3: CONCEPTOS GENERALES SOBRE SISTEMAS DE CONTROL Y DESARROLLO WEB**

Un sistema de control es aquel en el que las variables de salida se comportan según las órdenes dadas por las variables de entrada [6].

#### **3.1. Control en Lazo Cerrado**

Los sistemas de control realimentados se denominan también sistemas de control en lazo cerrado. En la práctica, los términos control realimentado y control en lazo cerrado se usan indistintamente. En un sistema de control en lazo cerrado, se alimenta al controlador la señal de error de actuación, que es la diferencia entre la señal de entrada y la señal de realimentación (que puede ser la propia señal de salida o una función de la señal de salida y sus derivadas y/o integrales), con el fin de reducir el error u llevar la salida del sistema a un valor deseado. El termino control en lazo cerrado siempre implica el uso de una acción de control realimentado para reducir el error del sistema [7].

#### **3.2. Control en Lazo Abierto**

Los sistemas en los cuales la salida no tiene efecto sobre la acción de control se denominan sistemas de control en lazo abierto. En otras palabras, en un sistema de control en lazo abierto no se mide la salida ni se realimenta para compararla con la entrada. Un ejemplo práctico es una lavadora. El remojo, el lavado y el centrifugado en la lavadora operan con una base de tiempo. La máquina no mide la señal de salida, que es la limpieza de la ropa. En cualquier sistema de control en lazo abierto, la salida no se compara con la entrada de referencia. Así, a cada entrada de referencia le corresponde una condición de operación fija; como resultado de ello, la precisión del sistema depende de la calibración. Ante la presencia de perturbaciones, un sistema de control en lazo abierto no realiza la tarea deseada. En la práctica, el control en lazo abierto sólo se usa si se conoce la relación entre la entrada y la salida y si no hay perturbaciones internas ni externas. Es evidente que estos sistemas no son de control realimentado.

Obsérvese que cualquier sistema de control que opere con una base de tiempo está en lazo abierto. Por ejemplo, el control de tráfico mediante señales operadas con una base de tiempo es otro ejemplo de control en lazo abierto [7].

### **3.3. Elementos de un Sistema de Control**

#### **3.3.1. Interfaces Hombre Maquina**

Comprende los sinópticos de control y los sistemas de presentación gráfica. La Función de un Panel Sinóptico es la de representar, de forma simplificada, el sistema bajo control (un sistema de aprovisionamiento de agua, una red de distribución eléctrica, una factoría) [8].

#### **3.3.2. Sensores o transductores**

Los dispositivos que convierten el valor de una variable controlada a una señal eléctrica se denominan transductores eléctricos. El número de transductores eléctricos diferentes es muy grande. Se han inventado transductores eléctricos para medir virtualmente toda variable física, sin importar qué tan complicada sea. En la industria, las variables físicas más importantes que se encuentran son la posición, velocidad, aceleración, fuerza, potencia, presión, velocidad de flujo, temperatura, intensidad luminosa, y humedad [9].

#### **3.3.3. Actuadores**

Los sistemas de actuadores son los elementos de los sistemas de control que transforman la salida de un microprocesador o un sistema de control en una acción de control para una máquina o dispositivo. Por ejemplo, si es necesario transformar una salida eléctrica del controlador en un movimiento lineal que realiza el desplazamiento de una carga. Otro ejemplo sería cuando la salida eléctrica del controlador anterior requiere transformarse en una acción que controle la cantidad de líquido que entra y circula en una tubería [10].

#### **3.3.4. Comunicaciones digitales**

Un bus externo es un conjunto de líneas a través de las cuales se transmiten señales y se conectan entre sí microprocesadores, microcontroladores, computadoras y PLC, los que a su vez se conectan con equipo periférico. Para obtener la impresión de la salida generada en una computadora es necesario conectar un bus para que mediante este conectar la computadora y la impresora. Los sistemas con varios multiprocesadores son muy comunes. En una planta automatizada no sólo es necesario transferir datos entre los controladores lógicos programables, visualizadores, sensores y actuadores, y que el operador introduzca datos y programas, también es preciso contar con un medio de comunicación para intercambiar datos con otras computadoras [10].

### **3.3.5. Controladores Automáticos**

Un controlador automático compara un valor real de la salida de una planta con la entrada de referencia (el valor deseado), determina la desviación y produce una señal de control que reduce la desviación a cero o a un valor pequeño. La manera en la cual el controlador automático produce la señal de control se denomina acción de control.

La Figura es un diagrama de bloques de un sistema de control industrial que consiste en un controlador automático, un actuador, una planta y un sensor (elemento de medición). El controlador detecta la señal de error, que, por lo general, está en un nivel de potencia muy bajo, y la amplifica a un nivel lo suficientemente alto. La salida de un controlador automático se alimenta a un actuador, como un motor o una válvula neumáticos, un motor hidráulico o un motor eléctrico. (El actuador es un dispositivo de potencia que produce la entrada para la planta de acuerdo con la señal de control, a fin de que la señal de salida se aproxime a la señal de entrada de referencia).

El sensor, o elemento de medición, es un dispositivo que convierte la variable de salida en otra variable manejable, como un desplazamiento, una presión o un voltaje, que pueda usarse para comparar la salida con la señal de entrada de referencia. Este elemento está en la trayectoria de realimentación del sistema en lazo cerrado. El punto de ajuste del controlador debe convertirse en una entrada de referencia con las mismas unidades que la señal de realimentación del sensor o del elemento de medición [7].

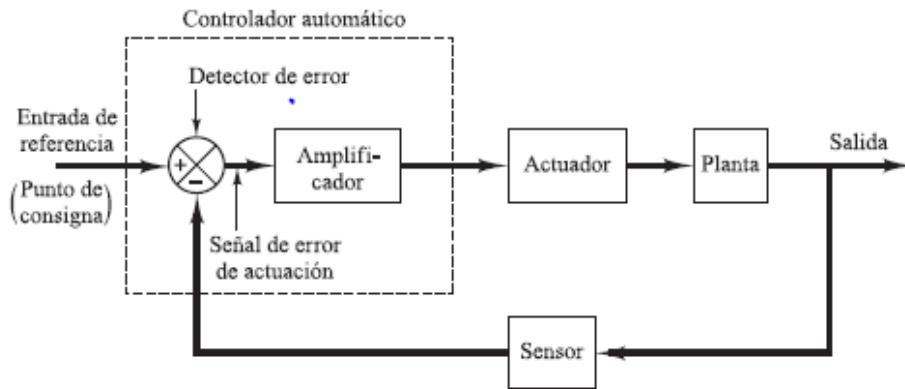


Figura 3: diagrama de bloque de un controlador Automático industrial [7].

### **3.4. Desarrollo web**

El desarrollo web se refiere a la creación y mantenimiento de sitios y aplicaciones web. Este proceso abarca tanto lo que ve el usuario (frontend) como lo que ocurre detrás de escena (backend). Para lograr esto, se utilizan diversas herramientas y lenguajes. Entre los más populares para el diseño y la interacción visual están HTML, CSS y JavaScript. Cada uno tiene una función específica: HTML estructura el contenido, CSS lo estiliza y JavaScript añade interactividad. A lo largo de los siguientes apartados se profundizará en estas herramientas y otras que complementan el mundo del desarrollo web.

#### **3.4.1. HTML**

Es un lenguaje markup (de hecho, las siglas de HTML significan Hyper Text Markup Language) usado para estructurar y presentar el contenido para la web. Es uno de los aspectos fundamentales para el funcionamiento de los sitios. Con el uso de HTML5, se puede reducir la dependencia de los plug-ins que tenemos que tener instalados para poder ver una determinada web [11].

#### **3.4.2. CSS**

CSS, que significa "Cascading Style Sheets" (Hojas de Estilo en Cascada), es un lenguaje de hoja de estilos utilizado para describir la presentación y diseño de documentos escritos en HTML o XML. En términos sencillos, mientras que HTML define la estructura y el contenido de un documento web, CSS define cómo se deben mostrar esos contenidos, estableciendo aspectos como colores, tamaños, márgenes, fuentes, alineaciones, entre otros [12] [13].

CSS permite a los desarrolladores separar el diseño y la estética de la estructura del contenido, proporcionando más flexibilidad y control en la presentación de las páginas web, y permitiendo una más fácil adaptabilidad a diferentes tipos de dispositivos y tamaños de pantalla. Esta separación también mejora la mantenibilidad y la reutilización del código [14].

#### **3.4.3. JavaScript**

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes [15].

Con JavaScript, los desarrolladores pueden crear sitios web más atractivos y funcionales, ya que permite la manipulación del contenido de la página, la comunicación con el servidor y la realización de cálculos en tiempo real, entre otras cosas [16].

#### **3.4.4. JSON**

JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos. JSON es de fácil lectura y escritura para los usuarios. JSON es fácil de analizar y generar por parte de las máquinas. JSON se basa en un subconjunto del lenguaje de programación JavaScript, Estándar ECMA-262 3a Edición - Diciembre de 1999. JSON es un formato de texto completamente independiente del lenguaje, pero que utiliza convenios que resultan familiares a los programadores de lenguajes de la familia C, incluidos C, C++, C#, Java™, JavaScript, Perl, Python y mucho otros. Estas características hacen de JSON un lenguaje de intercambio de datos ideal. [17].

#### **3.4.5. PHP**

PHP es un lenguaje de programación de código abierto utilizado principalmente para el desarrollo de aplicaciones web del lado del servidor. Fue creado en 1995 y es ampliamente utilizado en la creación de sitios web dinámicos y aplicaciones web.

PHP es compatible con varios sistemas operativos y servidores web, se integra fácilmente con bases de datos y es fácil de aprender para programadores principiantes.

PHP es un lenguaje de programación de alto nivel [18].

#### **3.4.6. Bootstrap**

Bootstrap (framework). Es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end. Es el segundo proyecto más destacado en GitHub<sup>1</sup> y es usado por la NASA y la MSNBC entre otras organizaciones [19].

Bootstrap fue desarrollado por Mark Otto y Jacob Thornton de Twitter, como un marco de trabajo (framework) para fomentar la consistencia entre las herramientas internas. Antes de Bootstrap, se usaban varias librerías para el desarrollo de interfaces de usuario, las cuales llevaban a inconsistencias y a una gran carga de trabajo en su mantenimiento [19].

### **3.4.7. Servidor web**

El término servidor tiene dos significados en el ámbito informático. El primero hace referencia al ordenador que pone recursos a disposición a través de una red, y el segundo se refiere al programa que funciona en dicho ordenador. En consecuencia aparecen dos definiciones de servidor:

- **Definición Servidor (hardware):** un servidor basado en hardware es una máquina física integrada en una red informática en la que, además del sistema operativo, funcionan uno o varios servidores basados en software. Una denominación alternativa para un servidor basado en hardware es "host" (término inglés para "anfitrión"). En principio, todo ordenador puede usarse como "host" con el correspondiente software para servidores.
- **Definición Servidor (software):** un servidor basado en software es un programa que ofrece un servicio especial que otros programas denominados clientes (clients) pueden usar a nivel local o a través de una red. El tipo de servicio depende del tipo de software del servidor. La base de la comunicación es el modelo cliente-servidor y, en lo que concierne al intercambio de datos, entran en acción los protocolos de transmisión específicos del servicio [20].

### **3.4.8. Bases de datos**

Una base de datos es una colección de datos relacionados. Por datos, nos referimos a hechos conocidos que pueden registrarse y que tienen un significado implícito. Por ejemplo, considere los nombres, números de teléfono y direcciones de las personas que conoce. Es posible que haya registrado estos datos en una libreta de direcciones indexada o que los haya almacenado en un disco duro, utilizando una computadora personal y software como Microsoft Access o Excel. Esta colección de datos relacionados con un significado implícito es una base de datos [21].

### III. RESUMEN EJECUTIVO

#### 4. CAPITULO 4

En este apartado se delinea el marco que se ha seguido para llevar a cabo esta tesis. La sección 5.4, proporciona una descripción general del trabajo realizado. A continuación, en 5.5, se especifica el tipo de investigación implementado. La sección 5.6 detalla las técnicas e instrumentos utilizados para la recolección de datos, mientras que en 5.7 se explican los métodos empleados para el análisis de estos datos. Finalmente, en la sección 5.8, se describen las diferentes fases metodológicas seguidas en el proceso de investigación. Esta estructura proporciona un marco claro para entender cómo se ha realizado el trabajo de investigación.

##### 4.1. Descripción del trabajo

El presente proyecto final de grado consiste en el diseño de un Sistema de Control Automatizado de Variables Claves, manejado mediante una aplicación Web, para la crianza de alevines del Centro Piscícola Nacional Ubicado en la localidad de Eusebio Ayala del Dpto. de Cordillera. Se espera monitorear y corregir en forma automática cualquier eventualidad presentada en los parámetros del agua que son la turbidez, la temperatura, el oxígeno, el pH. Todos esto en tiempo real desde un cliente de monitoreo, con el propósito de mejorar la condición de vida de los alevines y aumentar la producción. La metodología utilizada para abordar la resolución del mismo se expone a continuación.

##### 4.2. Tipo de Investigación

El tipo de investigación según el objeto de estudio es la investigación aplicada, pues con el proyecto se pretende buscar estrategias que puedan ser aplicadas para mejorar la eficiencia del proceso de cría de alevines, sustentadas en desarrollos tecnológicos en el área de la electrónica.

Según [22], la investigación aplicada se centra en el desarrollo de soluciones prácticas a problemas del mundo real. Implica el uso de conocimientos, teorías y técnicas existentes para abordar problemas o desafíos específicos en un campo o industria en particular.

Este estudio adopta un enfoque de investigación mixto, ya que combina métodos cuantitativos y cualitativos para obtener resultados. Por un lado, se utiliza una metodología cuantitativa, que se basa en datos numéricos recopilados y analizados estadísticamente. Por otro lado, también se emplea un enfoque cualitativo, que implica la observación directa y el análisis del proceso en estudio. Esta combinación de métodos proporciona una comprensión más completa del tema de investigación.

#### **4.3. Técnicas e instrumentos de recolección de Datos**

En este estudio, se emplearon dos técnicas principales para la recolección de datos: la observación y las encuestas. La técnica de observación, que fue no estructurada, permitió recopilar información relevante sobre el proceso piscícola y sus diversas problemáticas. Esta técnica implicó el registro descriptivo de mediciones en tiempo real. Por otro lado, se realizaron encuestas orales a través de entrevistas, utilizando una guía de entrevista como instrumento principal para el registro de datos. Además, se consultaron diversas fuentes como bibliografías, artículos científicos y repositorios académicos y páginas web para complementar la información necesaria para el desarrollo de la investigación.

#### **4.4. Métodos y análisis de datos**

En el análisis de datos fueron aplicados tanto el análisis cualitativo como el análisis cuantitativo por la razón de que los datos que fueron obtenidos mediante las técnicas e instrumentos de recolección de datos contienen ambas modalidades debido a la naturaleza del estudio.

En el análisis cuantitativo se utilizan métodos de estadística descriptiva como la media, mediana y la moda para el análisis de datos numéricos registrados en el centro piscícola, y el análisis de correlación para entender la relación entre la satisfacción del usuario o encargado con la tecnología aplicada en el proceso de cría de alevines.

Por otro lado, en el análisis de datos cualitativos se identificarán los aspectos, procedimientos, recursos y variables claves involucrados en el proceso del centro de

producción piscícola de manera descriptiva a fin de considerarlos durante el diseño de la propuesta tecnológica para el logro de su buen desempeño.

#### **4.5. Justificación**

El objetivo de este proyecto es diseñar un sistema automatizado para el monitoreo y control constante de las variables críticas en la producción de alevines. Este sistema facilitará la recopilación de datos sin intervención humana, minimizando errores y asegurando condiciones óptimas para el cultivo de alevines, lo que resultará en una mayor productividad.

El uso adecuado de instrumentos y equipos de control electrónicos permitirá abordar el problema de la baja productividad y obtener mejores resultados que los métodos manuales actuales. Este proyecto no solo proporcionará una herramienta útil, sino que también ofrecerá una solución inclusiva y participativa para el sector, al aumentar la productividad y permitir el control constante de los factores clave para la productividad del estanque.

Este sistema se podrá implementar en cualquier instalación de piscicultura con alta confiabilidad, permitiendo la observación y control en tiempo real de todas las variables críticas del proceso a través de una interfaz web. Además, permitirá la aplicación oportuna de medidas correctivas cuando se detecten variaciones fuera del rango aceptable en estas variables.

#### **4.6. Objetivos**

##### **4.6.1. Objetivo general**

Diseñar e implementar un sistema web para el monitoreo y control automatizado de estanques para la crianza de alevines en el Centro Piscícola Nacional de Eusebio Ayala, Cordillera.

##### **4.6.2. Objetivos específicos**

- Identificar las variables críticas que participan en la producción de alevines.

- Diseñar un sistema automatizado que pueda monitorear y controlar constantemente estas variables.
- Desarrollar una interfaz web que permita la observación y control en tiempo real de todas las variables críticas del proceso.
- Desarrollar un prototipo y realizar pruebas para garantizar su alta confiabilidad.
- Evaluar económicamente el proyecto

#### **4.7. Fases metodológicas**

Las técnicas empleadas con este proyecto de fin de grado para encarar el logro de los objetivos perseguidos se llevaron a cabo con las siguientes fases metodológicas:

##### **Fase I: Relevamiento de datos**

Se procedió a la recolección de datos en el Centro Nacional de Producción de Alevines y Cultivo de Surubí a fin de conocer todas las variables manejadas dentro del centro piscícola que guarden relación con la cría de alevines por medio de entrevista a los encargados y observación directa del proceso de cultivo. Se realizaron las consultas pertinentes a la Dirección Técnica del Centro piscícola. Para un mejor aprovechamiento de los recursos equipamiento/materiales se agruparon todos los componentes que puedan ser reutilizados en el proyecto

##### **Fase II: Identificación de las variables claves en el cultivo a ser medido en la acuicultura**

Se identificaron las variables claves a ser medidos y controlados para el cultivo de alevines posterior al análisis de las informaciones obtenidas del centro piscícola. De estas variables identificadas se determinó cuáles de ellos son los que generan la ineficiencia de la productividad de cultivo de alevines, estableciéndose cuales son los valores favorables de dichas variables para mejorar la productividad del cultivo. A partir de estos datos obtenidos el sistema propuesto es capaz de realizar las acciones pertinentes que permitan corregir la anomalía detectada.

##### **Fase III: Búsqueda de alternativas de solución tecnológica al problema planteado**

Se realizó una búsqueda minuciosa de las alternativas de control automático aplicables al proceso de cría de alevines, mediante consulta de libros y recursos de internet como

trabajos académicos sobre piscicultura y control automático, páginas web de fabricantes de productos, especificaciones técnicas de componentes y dispositivos electrónicos. Finalmente se elaboró una lista de alternativas que podrían aplicarse a la solución de la problemática.

#### **Fase IV: Definición de criterios para la selección de alternativas de solución**

Tras la identificación de posibles alternativas tecnológicas para la solución, se procedió a la fase de definición de criterios. En esta etapa, se establecieron diversos criterios técnicos y económicos que se consideraron para la selección de los elementos que se utilizarán en el diseño de la propuesta. Estos criterios incluyen la disponibilidad, el costo, la calidad, la compatibilidad y el respaldo técnico de las diferentes alternativas.

#### **Fase V: Evaluación y selección de la alternativa de monitoreo y control**

Durante esta etapa, se evaluaron todas las alternativas tecnológicas para el monitoreo y control basándose en criterios técnicos y económicos definidos previamente. Cada alternativa fue examinada cuidadosamente, permitiendo identificar sus fortalezas y debilidades. Posteriormente, se seleccionó el conjunto de tecnologías que mejor cumplía con estos criterios, asegurando que las alternativas elegidas estuvieran alineadas con los objetivos del proyecto y proporcionaran la mejor solución posible.

#### **Fase VI: Elaboración de la propuesta**

En esta fase se realiza el diseño de ingeniería. Que a su vez consta de diferentes etapas. Al principio se realiza una representación gráfica del sistema, mediante un diagrama de bloques y bosquejos a fin de tener un panorama general de lo que se hará en el proyecto, luego se realizan los esquemas electrónicos de las conexiones entre los diferentes componentes de hardware, tales como controladores, sensores y actuadores una vez que se tienen los esquema con las especificaciones técnicas requeridas, se procede al montaje físico que es un paso necesario para las pruebas durante el desarrollo de la aplicación web . Además, se realiza la programación de la aplicación web conjuntamente con la del controlador electrónico a fin de minimizar los errores y asegurar que el prototipo final funcione correctamente.

#### **Fase V Análisis de la factibilidad económica**

En esta última fase se evalúa si el proyecto es económicamente viable. Se realiza un análisis de costos y beneficios para determinar si el proyecto proporcionará un retorno

de inversión adecuado. También se consideran factores como el costo de adquisición o desarrollo de la tecnología necesaria, los costos operativos del sistema y cualquier posible ingreso o ahorro que el sistema pueda generar

## IV. DISEÑO DE INGENIERÍA

### 5. CAPITULO 5

#### 5.1. Descripción del trabajo

El diseño de la propuesta consiste en un sistema de monitoreo y control automático de un estanque para la crianza de alevines, este sistema cuenta con elementos de hardware y software. Por un lado, el software consiste en una interfaz HMI a nivel de aplicación considerando las capas del modelo OSI de las redes de comunicaciones electrónicas que servirá para que el encargado de los estanques pueda interactuar con el sistema. Mientras que por el lado del hardware, cuenta con un controlador que consistirá en un dispositivo electrónico programable, diferentes tipos de sensores para obtener el valor de las variables claves para la toma de decisiones del controlador y elementos actuadores para mantener la calidad del agua en el estanque.

La interfaz HMI será una aplicación web con funcionamiento en una red local pero con la posibilidad de escalarse para permitir el monitoreo a través de internet desde cualquier ubicación del encargado o funcionarios habilitados por el centro piscícola.

#### 5.2. Variables claves para el monitoreo

- Ph:

El pH es una medida de cuán ácida o alcalina es el agua. En términos sencillos, es una medida de la concentración de iones de hidrógeno en el agua. Este valor es crucial para la salud de los alevines, ya que diferentes especies de peces requieren diferentes niveles de pH para crecer de manera óptima. Un pH inadecuado puede estresar a los peces y hacerlos más susceptibles a enfermedades.

- Temperatura:

La temperatura del agua puede tener un impacto significativo en el metabolismo y el crecimiento de los peces. Los peces son animales de sangre fría, lo que significa que su temperatura corporal está determinada por la temperatura del ambiente. Si el agua está

demasiado caliente o demasiado fría, puede tener un impacto negativo en la salud y el crecimiento de los alevines.

- Turbidez del agua:

La turbidez se refiere a la claridad del agua. Cuando el agua es turbia, contiene partículas en suspensión que pueden interferir con la capacidad de los peces para ver y buscar alimentos. Además, la turbidez puede afectar la cantidad de luz que penetra en el agua, lo que puede afectar el crecimiento de las plantas acuáticas y, a su vez, alterar el equilibrio ecológico del estanque.

- Oxígeno disuelto en el agua:

Los peces y otros organismos acuáticos necesitan oxígeno para sobrevivir. A diferencia de los animales terrestres, los peces extraen el oxígeno del agua a través de sus branquias. Si los niveles de oxígeno disuelto son demasiado bajos, los peces pueden asfixiarse y morir.

- Nivel del agua:

El nivel del agua es un parámetro importante para asegurar que hay suficiente espacio y profundidad para los peces. Un nivel demasiado bajo puede dar lugar a un espacio de vida limitado para los peces y un calentamiento más rápido del agua. Además, un nivel de agua estable es a menudo un indicador de que no hay fugas o problemas con la infraestructura del estanque.

### **5.3. Criterios para la selección del controlador, los sensores y actuadores**

- Disponibilidad:

Elegir componentes que estén ampliamente disponibles asegura que no haya interrupciones en el suministro y que puedas mantener y expandir el sistema fácilmente en el futuro. En el contexto de un centro piscícola, la disponibilidad rápida y constante es crucial para mantener las condiciones óptimas del agua y garantizar la salud y el crecimiento de los alevines.

- Bajo costo sin comprometer la calidad:

La selección de componentes de bajo costo es una consideración económica esencial, especialmente si el proyecto se está desarrollando con un presupuesto limitado. Reducir los costos sin sacrificar la calidad permitirá que más estanques se beneficien de este sistema, haciendo que la solución sea más accesible para una variedad de operadores.

- Amplia comunidad o disponibilidad de información y recursos:

Elegir tecnologías y componentes que tengan una amplia comunidad de soporte y una gran cantidad de información y recursos disponibles significa que habrá una ayuda considerable en el desarrollo y mantenimiento del sistema. Esto puede acelerar el desarrollo, facilitar la resolución de problemas y aumentar la longevidad del sistema, ya que las actualizaciones y mejoras serán más fáciles de implementar.

- Compatibilidad técnica con el proyecto

Este criterio es fundamental para garantizar que todos los componentes funcionen juntos de manera cohesiva. La compatibilidad técnica asegura que no habrá conflictos inesperados entre los componentes, lo que podría causar retrasos y aumentar los costos. También permite una mayor eficiencia en el diseño, ya que los componentes compatibles suelen requerir menos tiempo y esfuerzo para integrar.

#### **5.4. Evaluación y selección de controladores**

En el presente apartado, se aborda la tarea de comparar y seleccionar el controlador más adecuado para el proyecto. Los controladores constituyen el núcleo de cualquier sistema automatizado y la elección de los mismos puede tener un impacto significativo en aspectos como el rendimiento, la eficiencia y la escalabilidad del sistema.

Las alternativas de control que se han considerado para este proyecto son: el Controlador Lógico Programable (PLC), el Microcontrolador PIC, el Arduino, el Field Programmable Gate Array (FPGA) y el Raspberry Pi. Cada uno de estos controladores posee sus propias fortalezas y debilidades, y su idoneidad depende en gran medida de las necesidades específicas del proyecto.

Se ha realizado una evaluación de cada controlador en función de los criterios mencionados anteriormente. En la siguiente tabla, se presenta un análisis detallado de cada controlador basado en estos criterios.

<b>Tecnología</b>	<b>Disponibilidad</b>	<b>Costo</b>	<b>Soporte</b>	<b>Compatibilidad Técnica de control</b>	<b>Facilidad para Implementar Web en Red Local</b>
PLC (Controlador Lógico Programable)	Baja-Media	Medio-Alto	Alto	Alta	Media (depende del modelo y las capacidades de red)
Microcontrolador PIC	Alta	Bajo	Medio	Media	Baja (necesita módulos adicionales y es más complicado)
FPGA (Array de Puertas Programables)	Media	Medio	Medio	Media-Alta	Baja (a menos que se integre con un procesador adicional)
Arduino	Alta	Bajo	Alto	Alta	Baja-Media (requiere módulos de red)
Raspberry Pi	Alta	Medio	Alto	Media	Media

Tabla 3: Evaluación de alternativas de control

Se puede observar que, mientras que algunos dispositivos como el Raspberry Pi están diseñados para funciones de computación y red, los mismos no están técnicamente optimizados para actuar sobre un proceso de control automático por lo que requieren de drivers y de programación de bajo nivel siendo su implementación un poco más compleja que las demás opciones según las averiguaciones que se hicieron sobre los mismos, otros como el Arduino o PIC están más centrados en tareas de control pero también requerirían de hardware o software adicional para la implementación web. Sin embargo, Arduino posee una amplia comunidad de soporte técnico y su configuración y programación es bastante sencillo.

Dispositivos como el PLC y el FPGA tienen costos mucho más elevados que las opciones mencionadas. Por lo que por cuestiones económicas no serán utilizados en el proyecto al existir opciones mucho más accesibles y con una relación de calidad-precio adecuada.

Por tanto, la alternativa seleccionada para el control automatizado del estanque para cría de alevines es el Arduino.

### 5.5. Especificaciones técnicas de los elementos seleccionados

La selección de los sensores y actuadores se ha realizado considerando la compatibilidad de los mismos con el controlador seleccionado. A continuación, se presentan las especificaciones técnicas de cada uno.

Sensor de temperatura:



Figura 4: Sensor de temperatura

- Tensión de alimentación del sensor de temperatura: 3,0 V ~ 5,5 V
- Resolución del sensor de temperatura: resolución ajustable de 9 a 12
- Rango de temperatura: -55 ~ +125C (el plomo sólo puede soportar una temperatura máxima de 85 grados)
- Los cables de salida del sensor de temperatura son: amarillo (DATA), rojo (VCC) y negro (GND).
- Cable adaptador: verde (DATOS), rojo (VCC), negro (GND)

#### 5.5.1. Sensor de TDS:

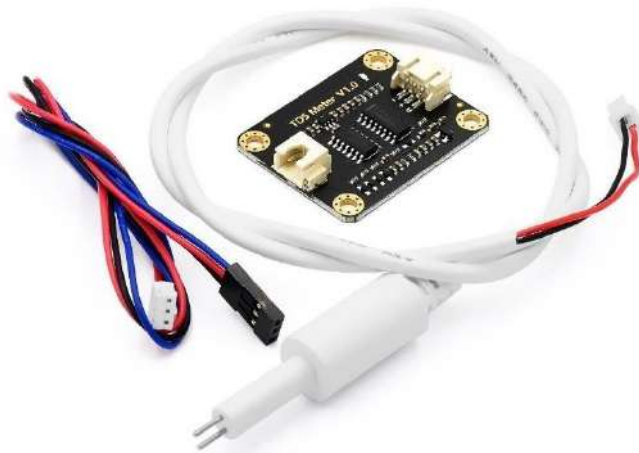


Figura 5: Sensor de turbidez

- Placa transmisora de señal:
- Voltaje de entrada: 3,3 ~ 5,5 V
- Voltaje de salida: 0 ~ 2,3 V
- Corriente de trabajo: 3~6mA
- Rango de medición de TDS: 0 ~ 1000 ppm
- Precisión de medición de TDS:  $\pm 10\%$  F.S. (25°C)
- Tamaño del módulo: 42\*32mm
- Interfaz del módulo: PH2.0-3P
- Interfaz del electrodo: XH2.54-2P
- Sonda TDS:
- Número de agujas: 2
- Longitud Total: 83cm
- Interfaz de conexión: XH2.54-2P
- Otro: sonda impermeable

### 5.5.2. Oxígeno disuelto en el agua:



Figura 6: Sensor de oxígeno disuelto en el agua

- Sonda de oxígeno disuelto
  - Tipo: Sonda Galvánica
  - Rango de detección: 0~20 mg/L
  - Tiempo de respuesta: Hasta 98 % de respuesta completa, en 90 segundos (25 °C)
  - Rango de presión: 0 ~ 50 PSI
  - Vida útil del electrodo: 1 año (uso normal)
  - Período de mantenimiento: Período de reemplazo de la tapa de membrana: 1 a 2 meses (en agua turbia); 4~5 meses (en agua limpia) Período de reemplazo de la solución de llenado: una vez al mes
  - Longitud del cable: 2 metros
  - Conector de sonda: BNC
- Placa convertidora de señal
  - Voltaje de funcionamiento: 3,3 ~ 5,5 V
  - Señal de salida: 0 ~ 3,0 V
  - Conector de cable: BNC
  - Conector de señal: Interfaz analógica por gravedad (PH2.0-3P)
  - Dimensión: 42 mm \* 32 mm

### 5.5.3. PH del agua:



Figura 7: Sensor de pH del agua

- Sensor de PH líquido
- Tensión de calentamiento:  $5 \pm 0,2$  V (CA - CC)
- Corriente de trabajo: 5-10mA
- El rango de concentración de detección: PH0-14
- El rango de detección de temperatura: 0-80 centígrados
- El tiempo de respuesta:  $\leq 5$ S
- Tiempo de estabilidad:  $\leq 60$ S
- Consumo de energía:  $\leq 0,5$  W
- La temperatura de trabajo: -10~50 centígrados (la temperatura nominal 20 centígrados)
- Humedad de trabajo: 95%RH (humedad nominal 65%RH)
- Vida útil: 3 años
- Tamaño: 42 mm x 32 mm x 20 mm
- Peso: 25g
- La salida: salida de señal de voltaje analógico
- Sonda de PH del electrodo BNC 1#
- El electrodo de PH tiene un solo cilindro que permite la conexión directa al terminal de entrada
- de un medidor de PH, controlador o cualquier dispositivo de PH que tenga un terminal de entrada BNC.

- La sonda del electrodo de PH es precisa y confiable y puede brindar lecturas casi instantáneas.
- Rango de PH: 0-14PH
- Rango de temperatura: 0-60 °C
- Punto cero:  $7 \pm 0,5$ PH
- Error alcalino: 0.2PH
- Pendiente Porcentual Teórica:  $\geq 98,5\%$
- Resistencia interna:  $\leq 250M\Omega$
- Tiempo de respuesta:  $\leq 1$ min
- Temperatura de funcionamiento: 0-60 °C
- Bloques de terminales: conector BNC
- Conector BNC adecuado para la mayoría de medidores y controladores de PH.
- Adecuado para una amplia gama de aplicaciones: acuarios, hidroponía, laboratorio, etc.

#### 5.5.4. Módulo de relé para Arduino:



Figura 8: Módulo relé de cuatro canales

- Cantidad de relés: 4
- Interfaz normalmente abierta.
- Señal de control TTL de 5V-12V.
- Se puede configurar la activación en alto o bajo mediante un puente.

- Carga máxima: CA 250 V/10 A, CC 30 V/10 A.
- Corriente de disparo: 50mA

#### 5.5.5. Calentador de agua:



Figura 9: Calentador de agua

#### Características:

- Carcasa de vidrio microcristalino (vitrocerámico).
- Resistente a la corrosión y a la abrasión, para uso en agua dulce y salada.
- Parte calefactora hecha de aleación de níquel-cromo.
- Temperatura ajustable: de 20 a 34 grados Celsius aproximadamente.
- Durante el proceso de calentamiento no se producen toxinas invisibles.
- Tensión de funcionamiento: 220 V AC.
- Consumo de energía: 100 W.

#### 5.5.6. Bomba de agua:



Figura 10: Bomba de agua

Características:

- Bomba sumergible diseñada para usarse en estanques, fuentes, cascadas, sistemas hidropónicos, sistemas de riego y fuentes de agua.
- Parte eléctrica sellada con epoxi con aislación de voltaje según normas UL e ISO.
- Altura máxima: 2,3 pies (0,7 m).
- Dimensiones: L: 2 W: 1,75 H: 1,25 pulgadas.
- Caudal ajustable. Flujo máximo 93 GPH - 350 L/H.
- Voltaje de funcionamiento: 220 V AC.
- Consumo de energía: 4,5 W. Cable de 1,7 m con enchufe no polarizado (EE.UU.)

### 5.5.7. Arduino Mega 2560



Figura 11: Arduino Mega 2560

Características:

- Voltaje de funcionamiento: 5V
- Voltaje de entrada (recomendado): 7-12V
- Voltaje de entrada (límite): 6-20 V
- Pines de E/S digitales: 54 (de los cuales 15 proporcionan salida PWM)
- Pines de entrada analógica: 16
- Corriente CC por pin de E/S: 20 mA
- Corriente CC para pin de 3,3 V: 50 mA
- Memoria Flash: 256 KB, de los cuales 8 KB son utilizados por el gestor de arranque
- SRAM: 8 KB
- EEPROM: 4KB
- Velocidad del reloj: 16MHz
- LED\_BUILTIN: 13
- Longitud: 101,52 mm
- Ancho: 53,3 mm
- Peso: 37 gramos

**5.5.8. Ethernet shield:**



Figura 12: Ethernet shield para Arduino

Características:

- Permite conexión a Internet.
- Se puede utilizar como servidor o cliente.
- No requiere soldadura para su instalación con Arduino.
- Controlador: chip Ethernet Wiznet W5100.
- Esta es la última versión de Ethernet Shield.
- Compatible con la biblioteca Ethernet oficial de Arduino.
- Posee ranura para tarjeta micro-SD, que se puede usar para almacenar archivos y servirlos a través de la red.
- Compatible con Arduino Duemilanove (168 o 328), Uno y Mega (1280/2560).
- Protocolo de comunicación TCP y UDP.
- Tamaño: 7 cm x 5,4 cm x 2,4 cm.
- Dirección MAC: 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED

**5.5.9. Alimentador de peces:**



Figura 13: Alimentador de peces

Características:

- Material: Plástico

- Peso: 260 g
- Tamaño: 14x14x8 cm
- Fuente de alimentación: 3 V
- Montaje con clip de fijación
- Capacidad de carga: 100 g

### 5.6. Arquitectura del Sistema

El diagrama de la Figura presenta la propuesta para el sistema de control de un estanque para cría de peces. El controlador electrónico programable se comunica con los sensores y actuadores a través de las interfaces correspondientes. Los sensores pueden medir diferentes parámetros del agua, como el pH, la temperatura, la conductividad (TDS), el oxígeno disuelto (OD) y el nivel del agua. Los actuadores pueden controlar diferentes aspectos del estanque, como la alimentación automática de los peces (Alim) y el calentamiento del agua (Calen). La aplicación web HMI permite al usuario monitorear y controlar el sistema a través de una interfaz gráfica.

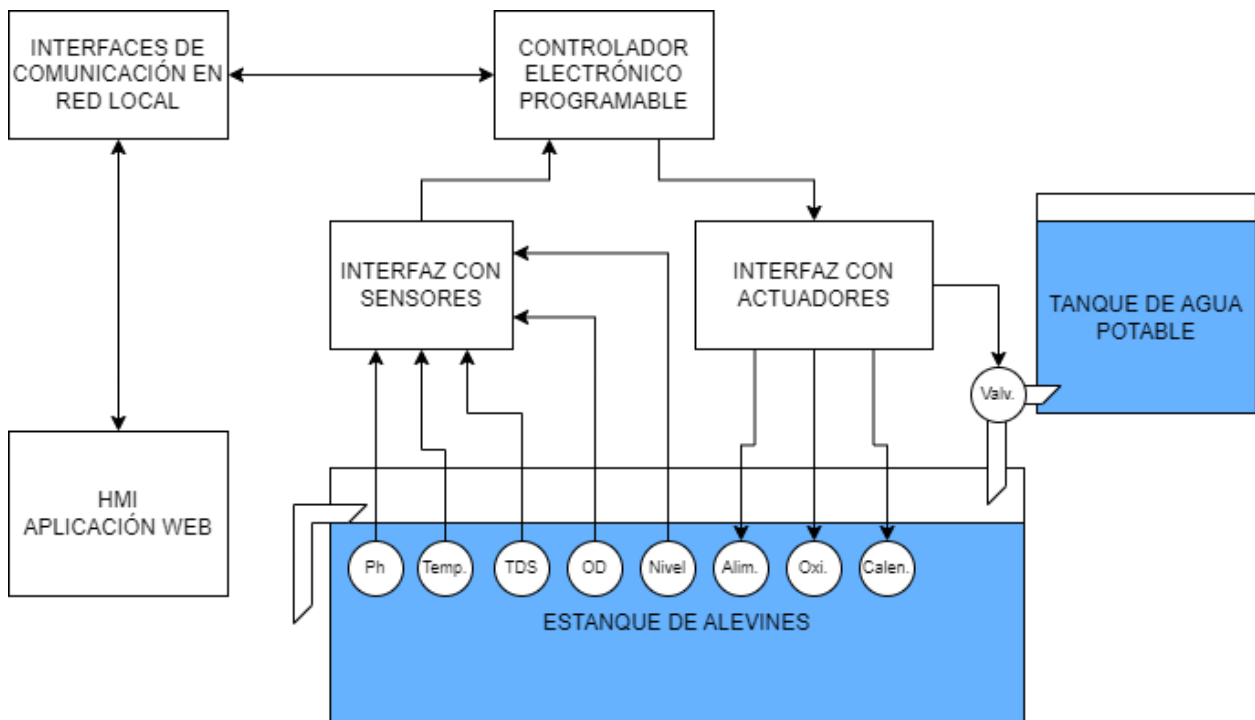


Figura 14: Esquema de bloques completo del sistema

Por otro lado, a nivel de software, se tiene el siguiente diagrama que muestra la aplicación web con sus diferentes elementos de software asociados. El funcionamiento es como sigue: HTML y CSS definen la estructura y el estilo de la página web. JavaScript añade interactividad y dinamismo, JSON facilita el intercambio de datos entre el cliente y el servidor, MySQL almacena y recupera los datos en el lado del servidor y PHP se ejecuta en el servidor para generar páginas web dinámicas e interactuar con la base de datos MySQL.

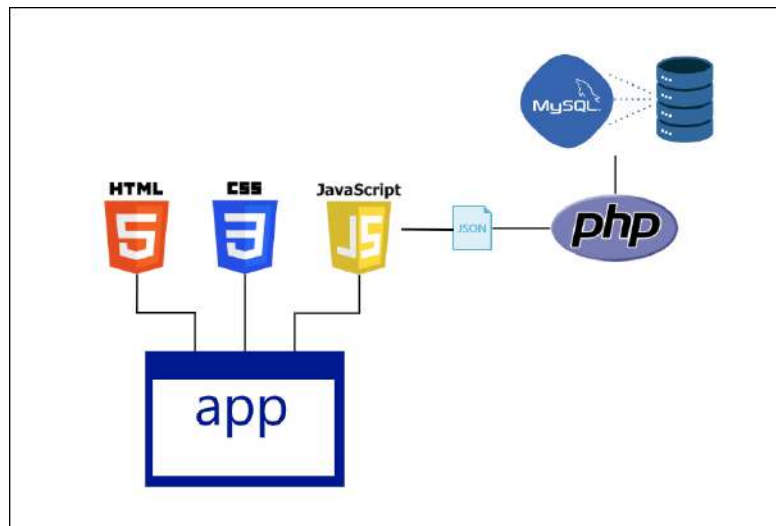


Figura 15: Diagrama de archivos de la aplicación web

## 5.7. Interfaz gráfica HMI

### 5.7.1. Descripción de la interfaz gráfica

La interfaz HMI consiste en una aplicación web que mostrará información en tiempo real sobre las condiciones del estanque, como la temperatura del agua, el nivel de oxígeno, el pH, etc. Permitirá al usuario ajustar manualmente los parámetros relacionados a las condiciones del estanque y configurar varios aspectos del sistema, como los umbrales para las alarmas o cambiar entre control manual y automático. La interfaz registrará los datos del estanque a lo largo del tiempo y proporcionará herramientas para visualizar estos datos. Si las condiciones del estanque se desvían de un rango aceptable, la interfaz alertará al usuario. Estas alertas serán visuales (por ejemplo, un indicador en la interfaz se vuelve rojo). Además, la interfaz HMI es una aplicación web con funcionamiento en una red local pero con la posibilidad de escalarse para permitir el monitoreo a través de

internet desde cualquier ubicación del encargado o funcionarios habilitados por el centro piscícola. Esto significa que el usuario puede acceder a la interfaz y controlar el sistema desde cualquier lugar con una conexión a Internet.

A continuación, mediante las siguientes figuras se pueden observar las interfaces de la aplicación web con sus diferentes partes: Inicio de sesión, Interfaz principal y sus características.



Figura 16: Interfaz de inicio de sesión

La interfaz de inicio de sesión permite que solamente los usuarios autorizados ingresen al sistema. El formulario de inicio de sesión se encuentra enlazada a una base de datos con los usuarios registrados. Una vez que el usuario ingresa los datos correctos, se abre la interfaz principal que se muestra en la siguiente figura.

### Sistema de monitoreo y control automatizado para crianza de alevines

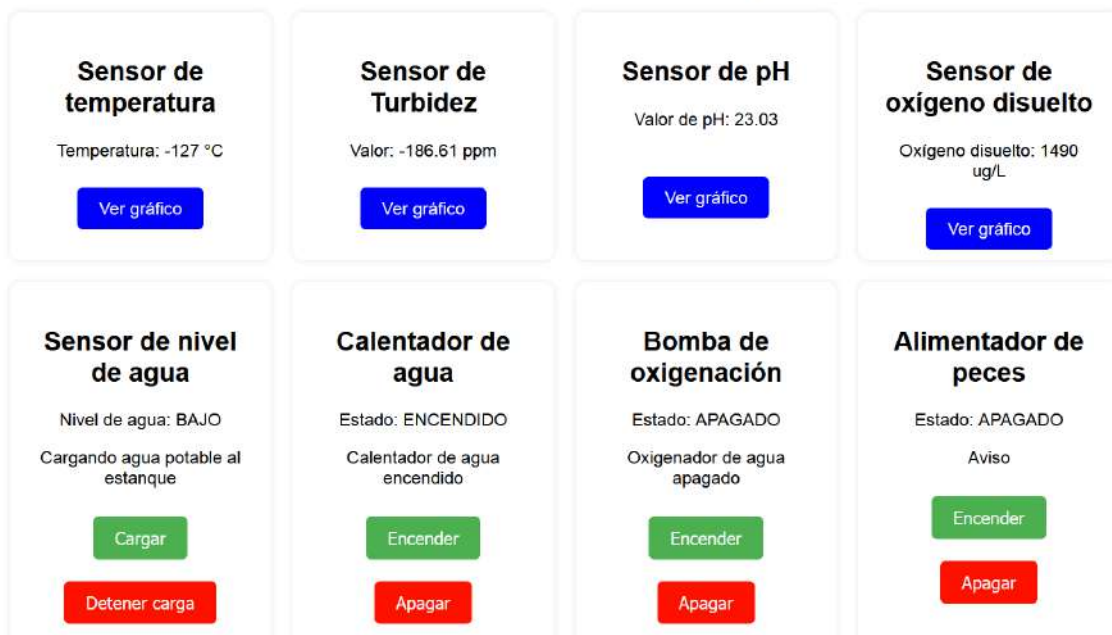


Figura 17: Interfaz principal de monitoreo y control

La interfaz principal permite al usuario tener un control total sobre las variables del sistema de control automático del estanque de cría de alevines, puede accionar los actuadores de manera manual en caso de que sea necesario o dejar que el sistema actúe automáticamente dependiendo de los valores de las variables claves.

Por otro lado, la figura muestra la interfaz de monitoreo de temperatura del sistema. Se puede apreciar la variación de los valores a medida que se accionan los actuadores, esto posibilita al usuario supervisar las acciones del sistema y corroborar los parámetros en un tiempo ínfimo.

INICIO

### Gráfico de Temperatura

Los valores se actualizan cada 5 segundos

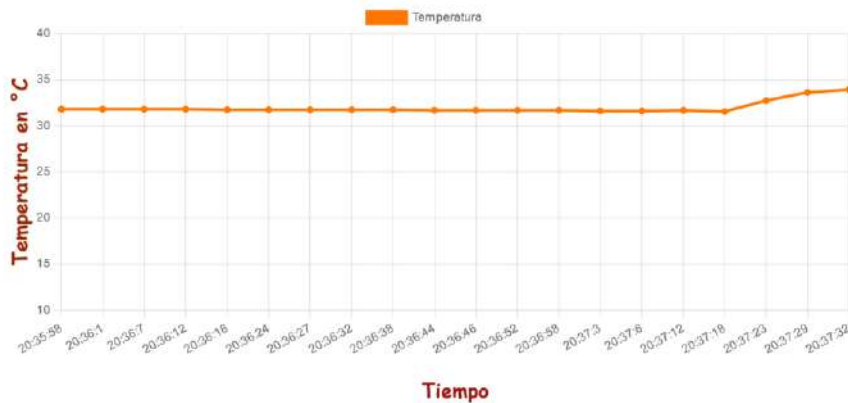


Figura 18: Interfaz de visualización de gráfico de temperatura vs. Tiempo

En el anexo se encuentran el código fuente de cada archivo que conforma la aplicación web.

Incluir Archivos de HTML, CSS, JAVASCRIPT, PHP, MYSQL

### 5.8. Funcionamiento y programación del hardware

El hardware del sistema juega un papel crucial en la recopilación de datos y la ejecución de acciones basadas en esos datos.

Los sensores recogen información sobre las condiciones del estanque. Miden los valores de temperatura del agua, el nivel de oxígeno, el pH y demás variables involucradas en el proceso, luego las envía al controlador. El controlador recibe información de los sensores, las procesa y luego toma decisiones basadas en la información recibida y los parámetros preestablecidos.

El controlador actúa sobre el proceso mediante los actuadores, que son los dispositivos que realizan acciones físicas basadas en las señales de control enviadas por el controlador. Por ejemplo, si un sensor informa que la temperatura del agua es demasiado baja, el controlador puede decidir activar un calentador para subir la temperatura.

El controlador intercambia información con la aplicación web (a través de una red local o Internet). Este último permite al usuario monitorear y controlar el proceso ajustando los parámetros o configurando alertas.

A continuación, se muestran los esquemas de conexión entre el Arduino y cada uno de los sensores y actuadores. Así también en el Anexo se presentan los códigos asociados para la lectura de los datos de los sensores en lenguaje c y c++ utilizado en el entorno de programación de Arduino.

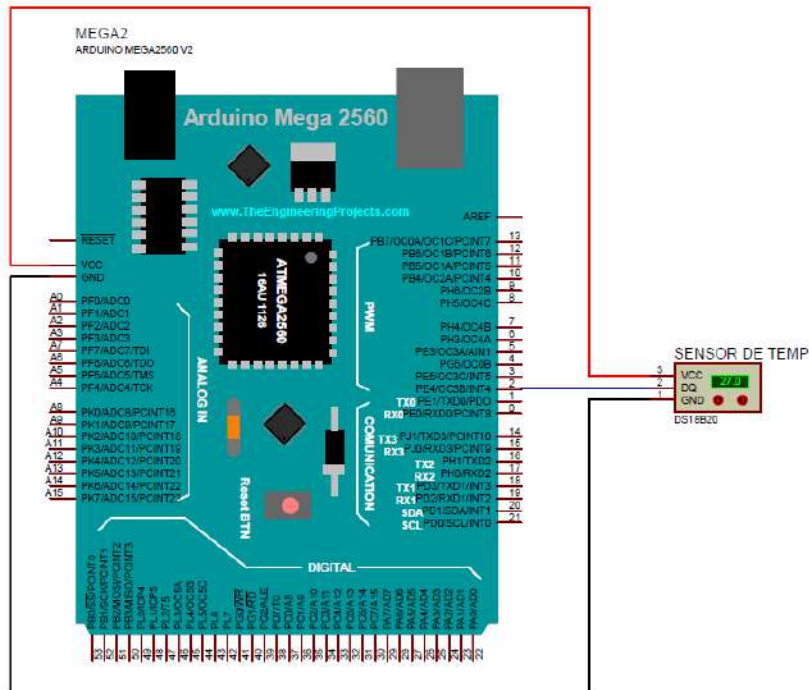


Figura 19: Diagrama de conexión entre Arduino Mega y sensor de temperatura.

El diagrama de la figura muestra la conexión entre el sensor de temperatura y el Arduino Mega, los datos enviados por el sensor entran a través del pín digital número dos y este a su vez está configurado como entrada. Para la lectura y procesamiento de sus valores se utiliza una librería llamada DallasTemperature [23] dentro del entorno de programación del Arduino.

Código para lectura de datos del sensor de temperatura en el Anexo.

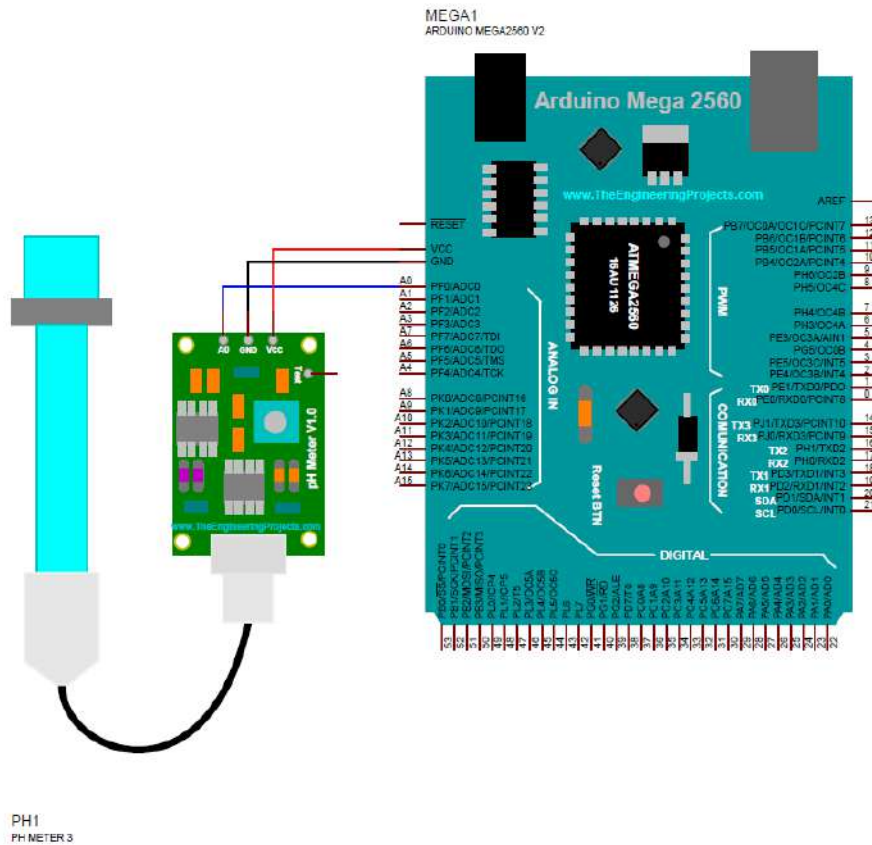


Figura 20: Diagrama de conexión entre Arduino Mega y sensor de PH del agua.

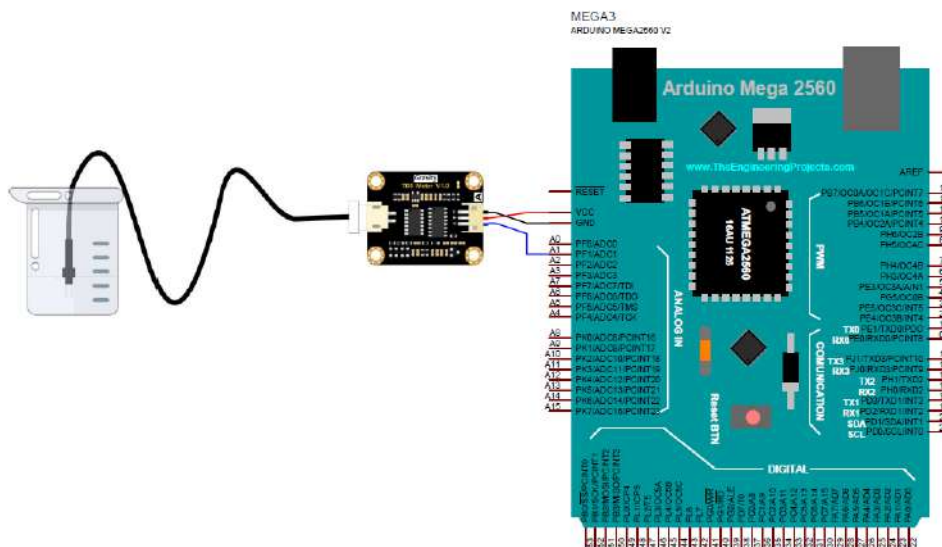


Figura 21: Diagrama de conexión entre Arduino Mega y sensor de Turbidez del agua.

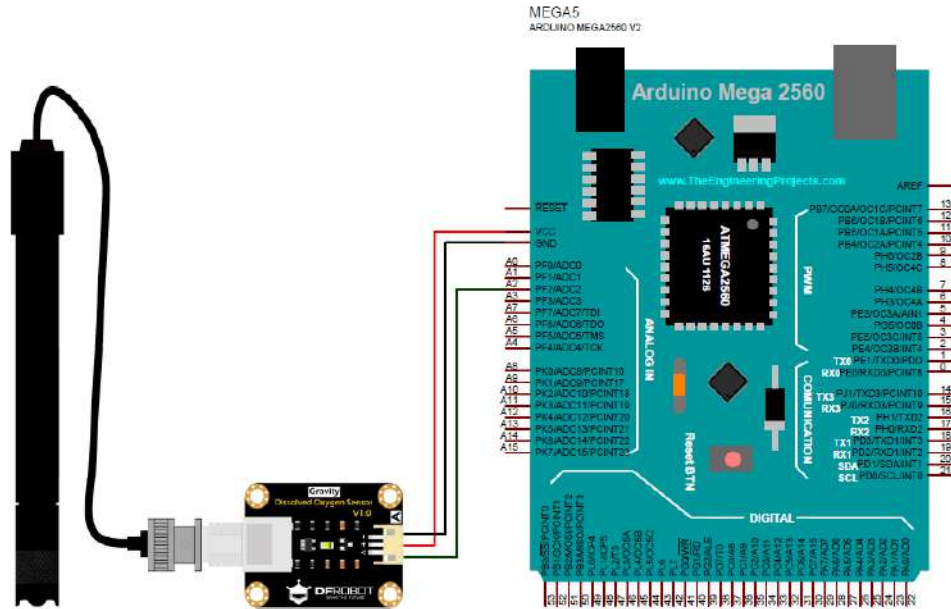


Figura 22: Diagrama de conexión entre Arduino Mega y sensor de oxígeno disuelto en el agua.

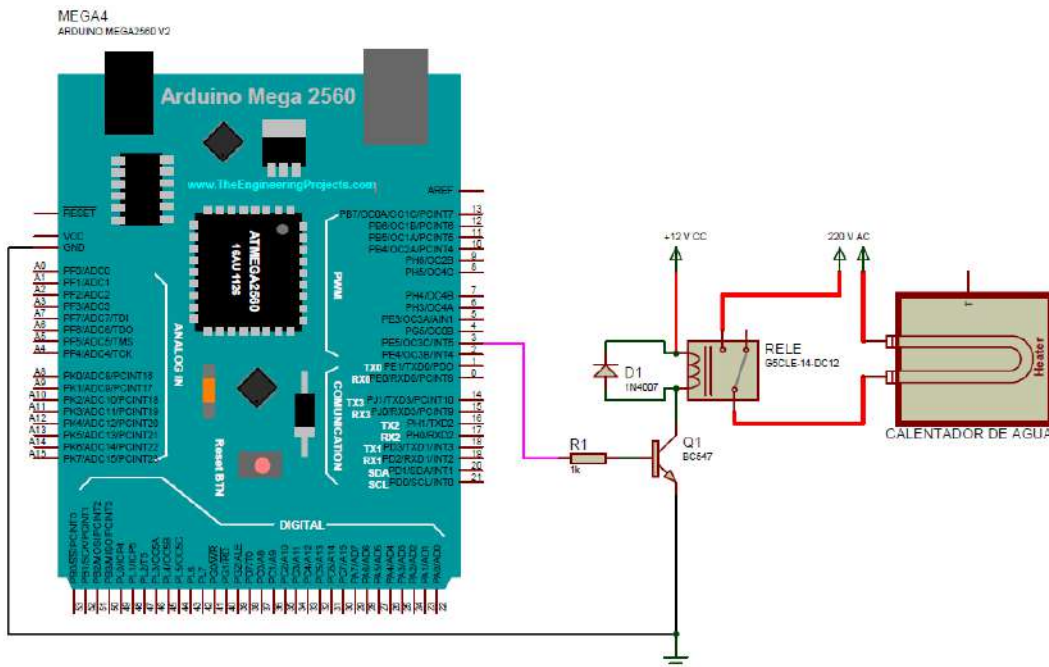


Figura 23: Diagrama de conexión entre Arduino Mega y calentador de agua.

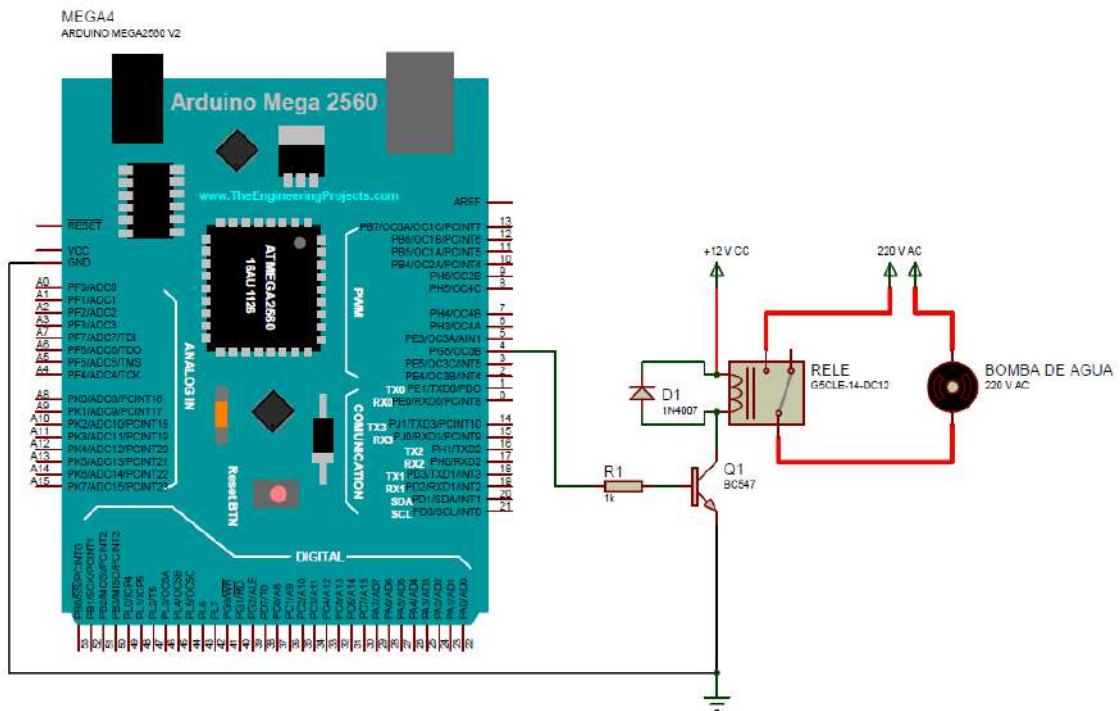


Figura 24: Diagrama de conexión entre Arduino Mega y bomba de agua.

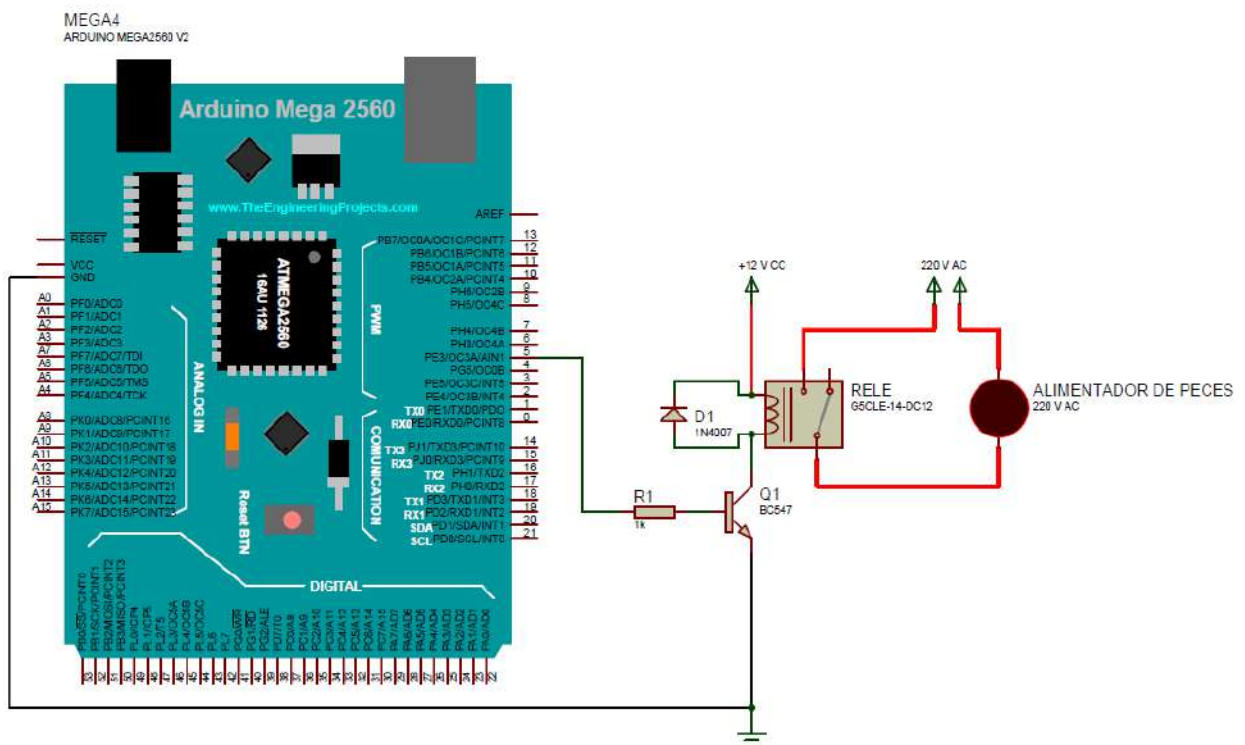


Figura 25: Diagrama de conexión entre Arduino Mega y alimentador de peces.

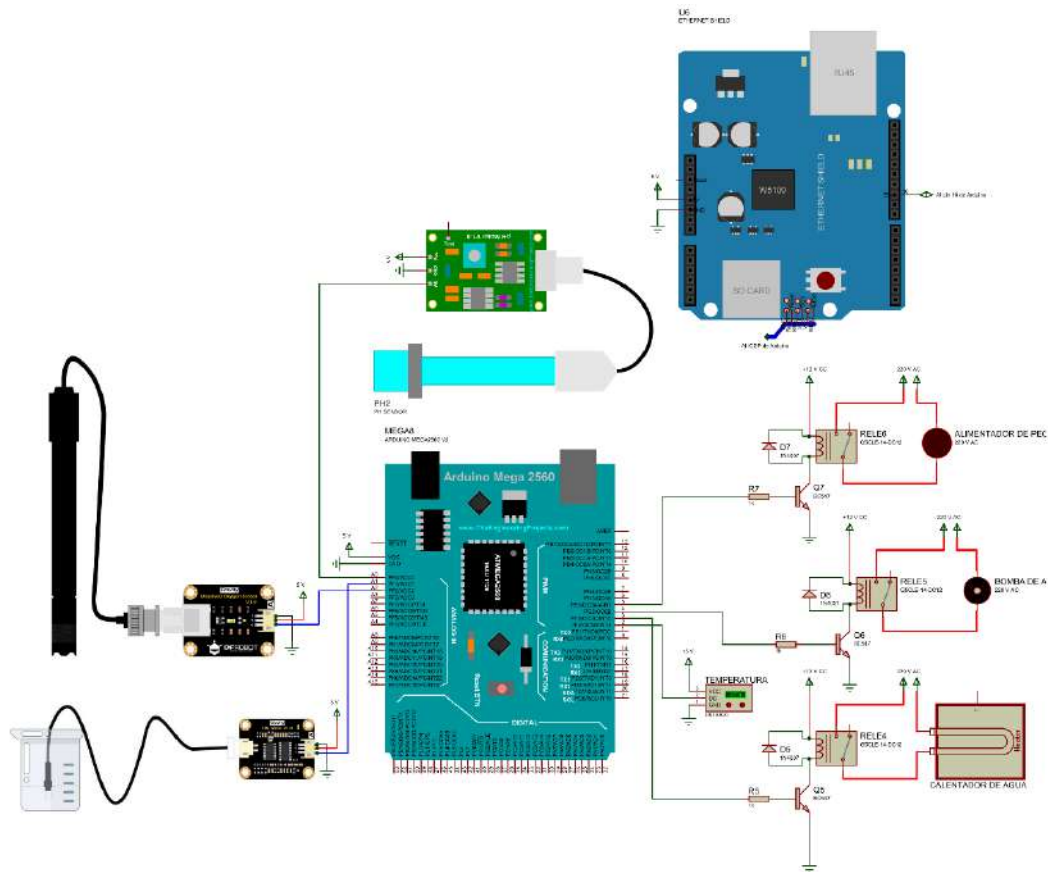


Figura 26: Esquema completo de la propuesta

## V. ANÁLISIS ECONÓMICO

### 6. CAPITULO 6

#### 6.1. Lista de materiales y costos

A continuación, se detalla la lista de materiales empleados en la construcción del prototipo, junto con sus respectivos costos, para facilitar la evaluación de los indicadores de viabilidad financiera.

Descripción	Costo
Arduino Mega 2560	150.000
Ethernet Shield W5100	100.000
Módulo Relé de 4 canales	65.000
Sensor de temperatura	75.000
Sensor de Ph	135.000
Sensor de turbidez	105.000
Sensor de Oxígeno disuelto	1.350.000
Sensor de nivel	50.000
Alimentador de peces	120.000
Calentador de agua	95.000
Bomba de agua sumergible	60.000
Router	250.000
Computadora de escritorio	2.400.000
Fuente de 9 V 2 A	30.000
Costo total de materiales	4.955.000

Tabla 4: Lista de materiales

Al costo de materiales se ha agregado un 10% por gasto en materiales diversos relacionados a la puesta en funcionamiento de los mismos, tales como: cables, protoboard, cinta aisladora, pistola de silicona, recipiente de plástico, electroválvulas, cajas, etc.

El costo por el desarrollo del sistema y la mano de obra por el montaje del prototipo se estimó en 20.000.000 de guaraníes considerando un salario mensual mínimo razonable

de 6.000.000 para un ingeniero electrónico y otros gastos relacionados a la utilización de equipos especializados, pagos de honorario a personales, viáticos, combustibles, etc.

El tiempo estimado para la implementación y operación del sistema es de aproximadamente dos meses como se muestra en el cronograma de la siguiente tabla.

Descripción	Tiempo	
	mes 1	mes 2
Desarrollo de la aplicación web	■	■
Programación del Arduino	■	■
Pruebas de funcionamiento individual de componentes electrónicos	■	■
Integración entre app web y el controlador	■	■
Calibración de sensores	■	■
Montaje del prototipo	■	■

Tabla 5: Costo del sistema

## 6.2. Indicadores financieros

Para realizar el análisis financiero del proyecto, primeramente, se debe conocer la tasa rendimiento anual de producción de alevines. Para ello se presenta la siguiente tabla.

Descripción	Cantidad
Huevos incubados	1000000
Alevines	200000
Porcentaje de alevines sobrevivientes	20%

Tabla 6: Resumen de rendimiento anual de alevines incubados

La tabla anterior muestra que de un total de 1000000 de huevos incubados solamente el 20% nacen. De esta cantidad existe un 70% al 100% de perdidas lo que significa que solamente el 30% en el mejor de los casos sobreviven para ser comercializados. Dicha información se resume en la tabla a continuación.

Descripción	Cantidad anual	Cantidad mensual
Huevos en incubación	1000000	16667
Huevos incubados	200000	83333
Pérdidas de alevines por mes (70%)		11667
Alevines sobrevivientes para comercialización (30%)		5000

Tabla 7: Detalles de alevines

La tabla presenta la cantidad de alevines disponibles para su comercialización al final del periodo de cría correspondiente a 2 meses por cada grupo incubado. De los 5000 alevines sobrevivientes se han dividido en tres grupos iguales para una mayor comodidad de cálculo y teniendo en cuenta que no se aleja mucho de la realidad.

Para la estimación de los montos se consideraron los precios para cada variedad de alevines que también se muestran en la tabla.

	Tipo de alevín	Cantidad por mes	Precio en el mercado	Venta al final del periodo (2 meses)
Para comercialización	Tilapia	1667	₡ 200	₡ 333.333
	Pacu	1667	₡ 500	₡ 833.333
	Surubi	1667	₡ 5000	₡ 8.333.333
			<b>Total:</b>	<b>₡ 9.500.000</b>

Tabla 8: Cantidad de alevines para comercialización

De manera similar, se estiman las pérdidas correspondientes al 70% de los alevines que no sobrevivieron al final del periodo. Que, para fines prácticos, también se agruparon en tres grupos con cantidades iguales.

	Tipo de alevín	Cantidad por mes	Precio en el mercado	No vendidas al final del periodo (2 meses)
Pérdidas	Tilapia	3889	₡ 200	₡ 777.778
	Pacu	3889	₡ 500	₡ 1.944.444
	Surubi	3889	₡ 5000	₡ 19.444.444
			<b>Total:</b>	<b>₡ 22.166.667</b>

Tabla 9: Detalles de pérdidas de alevines

Se aprecia una pérdida de ₡ 22.166.667 por cada 2 meses, considerando que el periodo para conseguir los alevines es de septiembre a marzo, son 7 meses, correspondería a 3 grupos de incubación. Por lo tanto, serían:

$$Pérdida\ anual = 3 * 22.166.667 = 66.500.000$$

### 6.3. Flujo de caja

Seguidamente se muestra una proyección a mediano plazo del proyecto en un escenario moderado como puede observarse en el flujo de caja de la tabla.

Se observa que, se tiene una tasa interna de retorno del 83%. Por otro lado, también se observa que el periodo de recuperación de la inversión es a partir del segundo año posterior a la implementación del proyecto.

Se estimaron algunos gastos anuales relacionados con la compra de repuestos y el mantenimiento del sistema.

AÑO	INVERSIÓN EN MATERIALES	COSTO DE IMPLEMENTACIÓN, OPERACIÓN Y MANTENIMIENTO	TOTAL EGRESO	AHORRO POR IMPLEMENTACIÓN DEL PROYECTO	TOTAL INGRESOS	FLUJO DE CAJA PROYECTADO	VAN
0	¢ 4.955.000	¢ 20.000.000	¢ 24.955.000	¢ 0	¢ 0	-¢ 24.955.000	
1	¢ 1.500.000	¢ 10.000.000	¢ 11.500.000	¢ 33.250.000	¢ 33.250.000	¢ 21.750.000	-¢ 5.360.405
2	¢ 1.500.000	¢ 10.000.000	¢ 11.500.000	¢ 33.250.000	¢ 33.250.000	¢ 21.750.000	¢ 12.292.383
3	¢ 1.500.000	¢ 10.000.000	¢ 11.500.000	¢ 33.250.000	¢ 33.250.000	¢ 21.750.000	¢ 28.195.795
4	¢ 1.500.000	¢ 10.000.000	¢ 11.500.000	¢ 33.250.000	¢ 33.250.000	¢ 21.750.000	¢ 42.523.194
5	¢ 1.500.000	¢ 10.000.000	¢ 11.500.000	¢ 33.250.000	¢ 33.250.000	¢ 21.750.000	¢ 55.430.760
TIR =						83%	

Tabla 10: Flujo de caja en el escenario moderado

## 6.4. Análisis de sensibilidad

### 6.4.1. Escenario optimista

El escenario optimista arrojó los siguientes resultados durante el análisis.

AÑO	INVERSIÓN EN MATERIALES	COSTO DE IMPLEMENTACIÓN, OPERACIÓN Y MANTENIMIENTO	TOTAL EGRESO	AHORRO POR IMPLEMENTACIÓN DEL PROYECTO	TOTAL INGRESOS	FLUJO DE CAJA PROYECTADO	VAN
0	¢ 4.955.000	¢ 20.000.000	¢ 24.955.000	¢ 0	¢ 0	-¢ 24.955.000	
1	¢ 1.500.000	¢ 10.000.000	¢ 11.500.000	¢ 46.550.000	¢ 46.550.000	¢ 35.050.000	¢ 6.621.577
2	¢ 1.500.000	¢ 10.000.000	¢ 11.500.000	¢ 46.550.000	¢ 46.550.000	¢ 35.050.000	¢ 35.068.943
3	¢ 1.500.000	¢ 10.000.000	¢ 11.500.000	¢ 46.550.000	¢ 46.550.000	¢ 35.050.000	¢ 60.697.201
4	¢ 1.500.000	¢ 10.000.000	¢ 11.500.000	¢ 46.550.000	¢ 46.550.000	¢ 35.050.000	¢ 83.785.721
5	¢ 1.500.000	¢ 10.000.000	¢ 11.500.000	¢ 46.550.000	¢ 46.550.000	¢ 35.050.000	¢ 104.586.190

TIR =	139%
-------	------

Tabla 11: Flujo de caja en el escenario optimista

### 6.4.2. Escenario pesimista

Finalmente, en el escenario pesimista, se puede apreciar que los resultados indican un proyecto viable.

AÑO	INVERSIÓN EN MATERIALES	COSTO DE IMPLEMENTACIÓN, OPERACIÓN Y MANTENIMIENTO	TOTAL EGRESO	AHORRO POR IMPLEMENTACIÓN DEL PROYECTO	TOTAL INGRESOS	FLUJO DE CAJA PROYECTADO	VAN
0	₡ 4.955.000	₡ 20.000.000	₡ 24.955.000	₡ 0	₡ 0	-₡ 24.955.000	
1	₡ 1.500.000	₡ 10.000.000	₡ 11.500.000	₡ 19.950.000	₡ 19.950.000	₡ 8.450.000	-₡ 17.342.387
2	₡ 1.500.000	₡ 10.000.000	₡ 11.500.000	₡ 19.950.000	₡ 19.950.000	₡ 8.450.000	-₡ 10.484.178
3	₡ 1.500.000	₡ 10.000.000	₡ 11.500.000	₡ 19.950.000	₡ 19.950.000	₡ 8.450.000	-₡ 4.305.611
4	₡ 1.500.000	₡ 10.000.000	₡ 11.500.000	₡ 19.950.000	₡ 19.950.000	₡ 8.450.000	₡ 1.260.666
5	₡ 1.500.000	₡ 10.000.000	₡ 11.500.000	₡ 19.950.000	₡ 19.950.000	₡ 8.450.000	₡ 6.275.330
						TIR =	21%

Tabla 12: Flujo de caja en el escenario pesimista

### 6.5. Conclusión de factibilidad

Tras el análisis financiero del proyecto, se concluye que es viable desde el punto de vista económico, teniendo en cuenta los resultados de los indicadores financieros evaluados. Este proyecto aportará beneficios al centro piscícola al incrementar su tasa de rendimiento en la producción de alevines y puede ser implementado tanto en centros pequeños como grandes. El prototipo propuesto está diseñado para un solo estanque, pero dada su factibilidad a corto plazo e incluso en el peor de los escenarios, puede ser implementado en más de un estanque con los ajustes necesarios o la incorporación de más sensores y actuadores.

## **VI. CONCLUSIONES Y RECOMENDACIONES**

Al término del presente proyecto se han cumplido satisfactoriamente los objetivos planteados. Se han identificado las variables críticas que intervienen en la producción de alevines, lo que ha permitido diseñar un sistema automatizado capaz de monitorear y controlar constantemente dichas variables. Además, se ha desarrollado una interfaz web que facilita la observación y el control en tiempo real de todas las variables críticas del proceso. Finalmente, se ha desarrollado un prototipo y se han realizado pruebas para garantizar su alta confiabilidad. Todo esto contribuye a la eficiencia y efectividad del proceso de producción de alevines, lo que a su vez puede tener un impacto positivo en la productividad y rentabilidad del Centro piscícola nacional de Eusebio Ayala.

Por otro lado, los resultados del análisis económico arrojaron valores que indican que el proyecto es viable económicamente.

Como recomendaciones se consideran las siguientes:

- Se recomienda la implementación del sistema de manera individual a los estanques del centro piscícola, es decir, un sistema por cada estanque para evitar grandes modificaciones del prototipo presentado.
- Se recomienda instalar el hardware de monitoreo y control en un recinto ambientado para el efecto.
- Respetar las normas eléctricas y de seguridad nacionales e internacionales en la instalación de los elementos del sistema.
- Instalar dispositivos y sistemas de protección contra descargas eléctricas atmosféricas para proteger los elementos del sistema.
- Elaborar un plan de mantenimiento periódico del sistema.

## VII. BIBLIOGRAFÍA

- [1] FAO, Acuicultura, "Organización de las Naciones Unidas para la Alimentación y la Acuicultura.," [Online]. Available: [http://www.fao.org/aquaculture/es/..](http://www.fao.org/aquaculture/es/) [Accessed 09 abril 2022].
- [2] Guerrero, Alfredo R. Polomino Ramos y Paul M. Baltazar, "Manual de cultivo de tilapias," Lima, 2004.
- [3] Ministerio de Agricultura y Ganadería - ONUAA, Manual Básico de Piscicultura para Paraguay, 2011.
- [4] García, Pablo Andrés Bejarano, Diseño de un sistema automatizado para producción de trucha, Bogotá, 2021.
- [5] Lisbeth Haydee Rivera Betancur, Solución IoT para la optimización del proceso de piscicultura en el Centro de Desarrollo, Bucaramanga, 2020.
- [6] A. Brunete, P. San Segundo and R. Herrero, Introducción a la Automatización Industrial, Madrid: Universidad Politécnica de Madrid, 2020.
- [7] K. Ogata, Ingeniería de control moderna, Madrid: PEARSON EDUCACIÓN, 2010.
- [8] J. M. Alfonso, ELECTRONICA INDUSTRIAL AVANZADA, Bogota: Centro, Bogota: centro nacional del medio para el aprendizaje , 2008.
- [9] W. Bolton, Mecatrónica Alfaomega, segunda edicion , 2001.
- [10] S. M. J. Park, Practical Data Acquisition for Instrumentation and Control, 2003.
- [11] B. Pavan, "hipertextual," 28 Mayo 2013. [Online]. Available: <https://hipertextual.com/2013/05/entendiendo-html5-guia-para-principiantes>. [Accessed 27 Junio 2023].
- [12] w3schools, "w3schools," [Online]. Available: <https://www.w3schools.com/css/>. [Accessed 28 Junio 2023].
- [13] World Wide Web Consortium, "W3C," 2023. [Online]. Available: <https://www.w3.org/Style/CSS/>. [Accessed 28 Junio 2023].
- [14] oregoom, "OG-oregoom," [Online]. Available: <https://oregoom.com/css/>. [Accessed 28 Junio 2023].
- [15] ictea, "ictea," [Online]. Available: <https://www.ictea.com/cs/index.php?rp=/knowledgebase/8801/iQue-es-el-lenguaje-de-programacion-JAVASCRIPT.html>. [Accessed 2 Julio 2023].
- [16] oregoom, "OG-oregoom," [Online]. Available: <https://oregoom.com/javascript/>. [Accessed 28 Junio 2023].
- [17] IBM, "IBM Business Automation Workflow," 7 Junio 2022. [Online]. Available: <https://www.ibm.com/docs/es/baw/20.x?topic=formats-javascript-object-notation-json-format>. [Accessed 2 Julio 2023].
- [18] oregoom, "OG-oregoom," [Online]. Available: <https://oregoom.com/php/>. [Accessed 2 Julio 2023].

- 
- [19] EcuRed, "Bootstrap (framework)," [Online]. Available: [https://www.ecured.cu/Bootstrap\\_\(framework\)](https://www.ecured.cu/Bootstrap_(framework)). [Accessed 2 Julio 2023].
- [20] IONOS, "Digital Guide IONOS," [Online]. Available: <https://www.ionos.es/digitalguide/servidores/know-how/que-es-un-servidor-un-concepto-dos-definiciones/>. [Accessed 2 Julio 2023].
- [21] R. Elmasri and S. B. Navathe, Fundamentals of database systems, Boston: Addison-Wesley, 2011.
- [22] M. Hassan, "Investigación Aplicada – Tipos, Métodos y Ejemplos," 1 Octubre 2023. [Online]. Available: <https://researchmethod.net/applied-research/>. [Accessed 6 Octubre 2023].
- [23] www.arduino.cc, [Online]. Available: <https://www.arduino.cc/reference/en/libraries/dallastemperature/>. [Accessed 20 Noviembre 2023].
- [24] "PISCICULTURA AMAZANICA CON ESPECIES NATIVAS," [Online]. Available: <http://www4.congreso.gob.pe/comisiones/1999/ciencias/cd/iiap1/texto.htm>. [Accessed 9 abril 2022].
- [25] "URUGUAY. Dirección Nacional de Recursos Acuáticos. 2010. Manual básico de," [Online]. [Accessed 13 abril 2022].
- [26] "wikipedia/controlador de dispositivos," [Online]. Available: [https://es.wikipedia.org/wiki/Controlador\\_de\\_dispositivo](https://es.wikipedia.org/wiki/Controlador_de_dispositivo). [Accessed 14 07 2021].
- [27] MARIO ENRIQUE AVILA PARADA, ASISTENCIA REMOTA PARA EL CONTROL DE DISPOSITIVOS EN PREDIOS, A TRAVÉS DEL SISTEMA OPERATIVO ANDROID, Bogota, 2014.
- [28] Controles Digitales, "Introduccion a los microcontroladores," [Online]. Available: <http://www.controladoresdigitales.com>. [Accessed 17 abril 2022].
- [29] tecnología+informática, [Online]. Available: <https://www.tecnologia-informatica.com/lenguaje-de-alto-nivel-que-es-ejemplos/>. [Accessed 21 Junio 2023].
- [30] Stackscale, 2022. [Online]. Available: <https://www.stackscale.com/es/blog/lenguajes-programacion-mas-populares/>. [Accessed 22 Junio 2023].
- [31] Pansy. [Online]. Available: <https://techlib.net/techedu/lenguaje-de-bajo-nivel/>. [Accessed 22 Junio 2023].
- [32] T. Statler. [Online]. Available: <https://compscicentral.com/programming-language-levels/>. [Accessed 22 Junio 2023].
- [33] BBC, [Online]. Available: <https://www.bbc.co.uk/bitesize/guides/zkck2p/revision/1>. [Accessed 23 Junio 2023].
- [34] J. Calvo, "European Valley," 17 Abril 2018. [Online]. Available: <https://www.europeanvalley.es/noticias/que-es-un-compiler-en-programacion/>. [Accessed 22 Junio 2023].
- [35] Instituto de Computación – Facultad de Ingeniería, Universidad de la República, "Persistencia en Sistemas O.O.," [Online]. Available: <https://www.fing.edu.uy/tecnoinf/mvd/cursos/progapl/material/teo/PA-03-Persistencia.pdf>. [Accessed 24 Junio 2023].

- [36] O. Margalida Noguera, C. López-Polín Hernanz and J. Salinas Ibáñez, "Depósito de Investigación. Universidad de Sevilla," [Online]. Available: [https://idus.us.es/bitstream/handle/11441/62460/El\\_interfaz\\_de\\_usuario\\_el\\_caso\\_de\\_campus\\_extens.pdf](https://idus.us.es/bitstream/handle/11441/62460/El_interfaz_de_usuario_el_caso_de_campus_extens.pdf). [Accessed 27 Junio 2023].
- [37] R. Cárdenas C, "La interfaz Hombre-Máquina en el Mundo de Internet," *Ingeniería*, vol. V, no. 2, pp. 88-94, 1999.
- [38] C. Date, *An Introduction to Database Systems*, Pearson Education, 2004.
- [39] S. Flores and D. Aracena, SISTEMA DE MONITOREO REMOTO DE ACUICULTURA EN ESTANQUES PARA LA CRIANZA DE CAMARONES, Tarapacá: Revista Chilena de Ingeniería, 2018.
- [40] A. F. Estrella Quispe, J. A. Cárdenas Carrión, J. O. Obregón Gutiérrez, G. P. Segura Núñez and C. F. Mendoza Vélez, "DISEÑO DE UN SISTEMA AUTOMATIZADO PARA LA PRODUCCIÓN PISCÍCOLA EN EL INSTITUTO SUPERIOR TECNOLÓGICO TSA´CHILA," *Ciencia Latina*, vol. VII, no. 5, p. 24, 2023.
- [41] V. A. Sánchez Villacis, IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO PARA EL CONTROL DE TEMPERATURA DEL AGUA Y ALIMENTACIÓN DE ALEVINES UTILIZANDO UNA PLATAFORMA CON SOFTWARE LIBRE, Ecuador: Universidad Regional Autónoma de los Andes, 2017.
- [42] P. D. Contreras Aristibal and M. A. Pérez Vado, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO DE OXIGENACIÓN DEL AGUA PARA EL CRIADERO ACUÍCOLA VALLE DEL MAR UBICADO EN SANTA MARTA, Colombia: Facultad de Ingeniería de Santa Marta, 2019.
- [43] L. H. Rivera Betancur, SOLUCIÓN IoT (EL INTERNET DE LAS COSAS) PARA LA OPTIMIZACIÓN DEL PROCESO DE PISCICULTURA EN EL CENTRO DE DESARROLLO AGROALIMENTARIO EL LIMONAL, Bucaramanga: Universidad Santo Tomás, 2020.
- [44] H. Díaz Lopez and Y. Vargas Gómez, DISEÑO DE UN MÓDULO ELECTRÓNICO PARA LA CRIANZA AUTOMATIZADA DE PECES MEDIANTE MODELAMIENTO MATEMÁTICO MULTIPARAMÉTRICO QUE SIMULE LAS CONDICIONES BÁSICAS NECESARIAS PARA LA CRIANZA, EN ESTANQUES ARTIFICIALES EN FUNCIÓN DE PARÁMETROS FÍSICOQUÍMICOS, Colombia: Revista UIS Ingenierías, 2018.

## VIII. ANEXO

### Aplicación web

#### Código HTML/PHP

#### Interfaz de inicio de sesión – login.php

```
<?php
session_start();
require_once 'conexion.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $usuario = $_POST['usuario'];
    $contrasena = $_POST['contrasena'];

    $sql = "SELECT * FROM usuarios WHERE usuario = ?";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([$usuario]);
    $user = $stmt->fetch();

    if ($user) {
        if (password_verify($contrasena, $user['contrasena'])) {
            $_SESSION['usuario_id'] = $user['id'];
            $_SESSION['usuario_nombre'] = $user['nombre'];
            header('Location: index.php');
        } else {
            echo "Contraseña incorrecta.";
            header('Location: login.php?error=1');
            exit;
        }
    } else {
        echo "Usuario no encontrado.";
        header('Location: login.php?error=1');
        exit;
    }
} /* else {
    echo "No se recibió una solicitud POST.";
} */
?>

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```



```
<?php
session_start();

if (!isset($_SESSION['usuario_id'])) {
    header('Location: login.php');
    exit();
}
?>

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Estanque de cría de peces</title>
    <link rel="stylesheet" href="css/styles.css">
</head>
<body>
    <!-- Barra de Navegación -->
    <div class="navbar">
        <a href="index.php">CENTRO PISCÍCOLA NACIONAL - Eusebio Ayala - Cordillera</a>
        <a href="diagrama.php">Ver diagrama</a>
        <a href="logout.php">Cerrar sesion</a>
    </div>
    <div class="header">
        <h1>Sistema de monitoreo y control automatizado para crianza de alevines</h1>
    </div>
    <div class="container">
        <div class="card">
            <h2>Sensor de temperatura</h2>
            <p>Temperatura: <span id="temperatura">----</span> °C</p>
            <br>
            <a href='http://localhost/piscicultura/temperatura.php' style='text-decoration: none; padding: 10px 20px; background-color: blue; color: white; border-radius: 5px;'>Ver gráfico</a>
        </div>
        <div class="card">
            <h2>Sensor de Turbidez</h2>
            <p>Valor: <span id="tds">----</span> ppm</p>
            <br>
            <a href='http://localhost/piscicultura/tds.php' style='text-decoration: none; padding: 10px 20px; background-color: blue; color: white; border-radius: 5px;'>Ver gráfico</a>
        </div>
        <div class="card">
            <h2>Sensor de pH</h2>
        </div>
    </div>
</body>
</html>
```

```
<p>Valor de pH: <span id="ph">----</span></p>
<br><br>
<a href='http://localhost/piscicultura/ph.php' style='text-decoration: none; padding: 10px 20px;
background-color: blue; color: white; border-radius: 5px;'>Ver gráfico</a>
</div>
<div class="card">
  <h2>Sensor de oxígeno disuelto</h2>
  <p>Oxígeno disuelto: <span id="oxigeno">----</span> ug/L</p>
  <br>
  <a href='http://localhost/piscicultura/oxigeno.php' style='text-decoration: none; padding: 10px
20px; background-color: blue; color: white; border-radius: 5px;'>Ver gráfico</a>
</div>
<div class="card">
  <h2>Sensor de nivel de agua</h2>
  <p>Nivel de agua: <span id="nivel">----</span></p>
  <p id="nivelalerta">Aviso</p>
  <button id="onled" onclick="actagua(1)">Cargar</button>
  <button id="offled" onclick="actagua(0)">Detener carga</button>
</div>
<div class="card">
  <h2>Calentador de agua</h2>
  <p>Estado: <span id="calentEstado">OFF</span></p>
  <p id="calentadoralerta">Aviso</p>
  <button id="onled" onclick="actcalentador(1)">Encender</button>
  <button id="offled" onclick="actcalentador(0)">Apagar</button>
</div>
<div class="card">
  <h2>Bomba de oxigenación</h2>
  <p>Estado: <span id="bombaEstado">OFF</span></p>
  <p id="oxigenacionalerta">Aviso</p>
  <button id="onled" onclick="actbomba(1)">Encender</button>
  <button id="offled" onclick="actbomba(0)">Apagar</button>
</div>
<div class="card">
  <h2>Alimentador de peces</h2>
  <p>Estado: <span id="alimEstado">OFF</span></p>
  <p id="alimentadoralerta">Aviso</p>
  <button id="onled" onclick="actalim(1)">Encender</button>
  <button id="offled" onclick="actalim(0)">Apagar</button>
</div>
</div>
<script src="scripts.js"></script>
<h2>Desarrollado por Cesar R. Rojas Mereles</h2>
<h2>Ultimos registros, actualice la página</h2>
```

```
<p id="registros"></p>
</body>
</html>
```

## Interfaz de gráfico de temperatura

```
<?php
session_start();

if (!isset($_SESSION['usuario_id'])) {
    header('Location: login.php');
    exit();
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>Monitor de Temperatura</title>
    <link rel="stylesheet" href="css/styles.css">
    <script src="http://localhost/piscicultura/chart.js"></script>
    <style>
        body {
            display: flex;
            flex-direction: column;
            height: 100vh;
        }
        .header {
            text-align: center;
        }
        #myChart {
            flex-grow: 1;
            max-height: 70%;
        }
    </style>
</head>
<body>
    <!-- Barra de Navegación -->
    <div class="navbar">
        <a href="index.php">INICIO</a>
    </div>
    <div class="header">
        <h1>Gráfico de Temperatura</h1>
```

```
<h3>Los valores se actualizan cada 5 segundos</h3>
</div>
<div class="grafico">
<canvas id="myChart"></canvas>
</div>
<script>
var ctx = document.getElementById('myChart').getContext('2d');
var chart = new Chart(ctx, {
  type: 'line',
  data: {
    labels: [],
    datasets: [{
      label: 'Temperatura',
      backgroundColor: 'rgb(255, 119, 0)',
      borderColor: 'rgb(255, 119, 0)',
      data: []
    }]
  },
  options: {
    responsive: true,
    scales: {
      y: { // escala y del gráfico

        title: {
          display: true,
          text: 'Temperatura en °C',
          color: '#920',
          font: {
            family: 'Comic Sans MS',
            size: 20,
            weight: 'bold',
            lineHeight: 1.2,
          },
        },
        padding: {top: 20, left: 0, right: 0, bottom: 0}
      },
      beginAtZero: false,
      min: 10,
      max: 40,
    },
    x: { // escala x del gráfico
      ticks: {
        autoSkip: true,
        maxTicksLimit: 20
      },
    },
  },
}
```

```
        title: {
display: true,
text: 'Tiempo',
color: '#911',
font: {
family: 'Comic Sans MS',
size: 20,
weight: 'bold',
lineHeight: 1.2,
},
padding: {top: 20, left: 0, right: 0, bottom: 0}
}
}
}
});

async function fetchTemperature() {
const response = await fetch('http://192.168.100.177'); // La IP del Arduino
const text = await response.text();
console.log(text);
const temp = parseFloat(text.split("Temperatura: ")[1]);
console.log(temp);
const now = new Date();
chart.data.labels.push(now.getHours() + ":" + now.getMinutes() + ":" + now.getSeconds());
chart.data.datasets[0].data.push(temp);
if (chart.data.labels.length > 20) { // por ejemplo, limitar a 20 puntos de datos
chart.data.labels.shift(); // Elimina la etiqueta más antigua
chart.data.datasets[0].data.shift(); // Elimina el dato más antiguo
}

chart.update();
}

setInterval(fetchTemperature, 5000); // Actualiza cada 5 segundos
</script>
</body>
</html>
```

## Código CSS

### Estilo de inicio de sesión

```
body {
position: relative;
```

```
font-family: 'Roboto', sans-serif;
background-color: #f5f5f5;
min-height: 100vh;
}
body::before {
content: "";
position: absolute;
top: 0; right: 0; bottom: 0; left: 0;
background-image: url('http://localhost/piscicultura/img/fondo.png');
background-size: cover;
background-repeat: no-repeat;
background-position: center;
z-index: -1;
}
.card {
box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
}
.form-group label {
font-weight: bold;
}
.btn-primary {
background-color: #00b51b;
border-color: #00b51b;
}
.btn-primary:hover {
background-color: #56c918;
border-color: #56c918;
}
```

## Estilo principal – styles.css

```
body {
background-color: rgb(255, 255, 255);
display: flex;
flex-direction: column;
font-family: Arial, sans-serif;
}
h1, h2 {
text-align: center;
margin-bottom: 20px;
color: black;
}
.container {
```

```
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
    gap: 20px;
}
.card {
    background-color: #fff;
    border-radius: 10px;
    box-shadow: 0 0 5px rgba(0, 0, 0, 0.1);
    padding: 20px;
    width: 200px;
    text-align: center;
}
button {
    background-color: #4CAF50;
    border: none;
    border-radius: 5px;
    color: white;
    cursor: pointer;
    padding: 10px 20px;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 10px 2px;
    transition-duration: 0.4s;
}
#offled {
    background-color: #ff1100;
    border: none;
    border-radius: 5px;
    color: white;
    cursor: pointer;
    padding: 10px 20px;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 10px 2px;
    transition-duration: 0.4s;
}
button:hover {
    background-color: #45a049;
}
/* Estilos de la Barra de Navegación */
.navbar {
```

```
    background-color:rgb(0, 0, 0); /* Color de fondo */
}
.navbar a {
    float: left;
    display: block;
    color: rgb(255, 255, 255); /* Color del texto */
    text-align: center;
    padding: 14px 20px;
    text-decoration: none;
}
/* Cambia el color de los enlaces al pasar el mouse */
.navbar a:hover {
    background-color: #4CAF50;
    color: white;
}

.diagrama {
    text-align: center;
}
.desdiagrama h2{
    text-align: center;
}
.grafico {
    display: flex;
    align-self: center;
    align-items: center;
    background-color: #fff;
    min-height: 80vh;
    min-width: 70%;
    max-height: 90vh;
    max-width: 80%;
}
```

## Código Javascript – scripts.js

```
function actagua(valor) {
    // Código para interactuar con el renovador de agua
    if (valor==1){
        fetch('http://192.168.100.177/VAL=ON')
        .then(response => {
            if(response.ok) {
                console.log("Se esta cargando el agua");
            } else {

```

```
        console.error("Error al intentar cargar agua");
        document.getElementById('registros').textContent='Error al intentar cargar agua';
    }
})
.catch(error => {
    console.error("Error en la red o servidor no accesible: ", error);
    document.getElementById('registros').textContent='Error en la red o servidor no accesible';
});
document.getElementById('nivelalerta').textContent='Cargando agua potable al estanque';
}
if (valor==0){
    fetch('http://192.168.100.177/VAL=OFF')
    .then(response => {
        if(response.ok) {
            console.log("Se detuvo la carga de agua al estanque");
        } else {
            console.error("Error al intentar detener la carga de agua");
            document.getElementById('registros').textContent='Error al intentar detener la carga de
agua';
        }
    })
    .catch(error => {
        console.error("Error en la red o servidor no accesible: ", error);
        document.getElementById('registros').textContent='Error en la red o servidor no accesible';
    });
    document.getElementById('nivelalerta').textContent='Se detuvo la carga de agua potable al
estanque';
}
}
function actcalentador(valor) {
    // Código para interactuar con el calentador de agua
    if (valor==1){
        fetch('http://192.168.100.177/CAL=ON')
        .then(response => {
            if(response.ok) {
                console.log("Calentador encendido");
            } else {
                console.error("Error al intentar encender el Calentador");
                document.getElementById('registros').textContent='Error al intentar encender el
Calentador';
            }
        })
        .catch(error => {
```

```
        console.error("Error en la red o servidor no accesible: ", error);
        document.getElementById('registros').textContent='Error en la red o servidor no accesible';
    });
    document.getElementById('calentadoralerta').textContent='Calentador de agua encendido';
}
if (valor==0){
    fetch('http://192.168.100.177/CAL=OFF')
    .then(response => {
        if(response.ok) {
            console.log("Calentador apagado");
        } else {
            console.error("Error al intentar apagar el Calentador");
            document.getElementById('registros').textContent='Error al intentar apagar el Calentador';
        }
    })
    .catch(error => {
        console.error("Error en la red o servidor no accesible: ", error);
        document.getElementById('registros').textContent='Error en la red o servidor no accesible';
    });
    document.getElementById('calentadoralerta').textContent='Calentador de agua apagado';
}
}
function actbomba(valor) {
    // código para interactuar con la bomba de agua real
    if (valor==1){
        fetch('http://192.168.100.177/BOM=ON')
        .then(response => {
            if(response.ok) {
                console.log("Oxigenador de agua encendido");
            } else {
                console.error("Error al intentar encender el Oxigenador de agua");
                document.getElementById('registros').textContent='Error al intentar encender el Oxigenador
de agua';
            }
        })
        .catch(error => {
            console.error("Error en la red o servidor no accesible: ", error);
            document.getElementById('registros').textContent='Error en la red o servidor no accesible';
        });
        document.getElementById('oxigenacionalerta').textContent='Oxigenador de agua encendida';
    }
    if (valor==0){
        fetch('http://192.168.100.177/BOM=OFF')
```

```
.then(response => {
  if(response.ok) {
    console.log("Oxygenador de agua apagado");
  } else {
    console.error("Error al intentar apagar el Oxygenador de agua");
    document.getElementById('registros').textContent='Error al intentar apagar el Oxygenador de agua';
  }
})
.catch(error => {
  console.error("Error en la red o servidor no accesible: ", error);
  document.getElementById('registros').textContent='Error en la red o servidor no accesible';
});
document.getElementById('oxigenacionalerta').textContent='Oxygenador de agua apagado';
}
}
function actalim(valor) {
  // Aquí puede agregar el código para activar el alimentador de peces
  if (valor==1){
    fetch('http://192.168.100.177/ALIM=ON')
    .then(response => {
      if(response.ok) {
        console.log("Alimentador encendido");
      } else {
        console.error("Error al intentar encender el Alimentador");
        document.getElementById('registros').textContent='Error al intentar encender el Alimentador';
      }
    })
    .catch(error => {
      console.error("Error en la red o servidor no accesible: ", error);
      document.getElementById('registros').textContent='Error en la red o servidor no accesible';
    });
    document.getElementById('alimentadoralerta').textContent='Alimentador de peces encendido';
  }
  if (valor==0){
    fetch('http://192.168.100.177/ALIM=OFF')
    .then(response => {
      if(response.ok) {
        console.log("Alimentador apagado");
      } else {
        console.error("Error al intentar apagar el Alimentador");
        document.getElementById('registros').textContent='Error al intentar apagar el Alimentador';
      }
    })
  }
}
```

```
    })
    .catch(error => {
      console.error("Error en la red o servidor no accesible: ", error);
      document.getElementById('registros').textContent='Error en la red o servidor no accesible';
    });
    document.getElementById('alimentadoralerta').textContent='Alimentador de peces apagado';
  }
}
// función para obtener los datos del Arduino
async function fetchDataFromArduino() {
  const response = await fetch('http://192.168.100.177');
  const text = await response.text();
  console.log("Separador inicial");
  const temp = parseFloat(text.split("Temperatura: ")[1]);
  console.log(temp);
  document.getElementById('temperatura').innerHTML = temp;
  let temperatura = parseFloat(document.getElementById('temperatura').innerHTML);
  console.log(temperatura);
  const ph = parseFloat(text.split("Valor de pH: ")[1]);
  console.log(ph);
  document.getElementById('ph').textContent = ph;
  const tds = parseFloat(text.split("Valor de tds: ")[1]);
  console.log(tds);
  document.getElementById('tds').textContent = tds;
  const oxi = parseFloat(text.split("Oxigeno disuelto: ")[1]);
  console.log(oxi);
  document.getElementById('oxigeno').textContent = oxi;
  const calentador = parseFloat(text.split("Calentador: ")[1]);
  console.log(calentador);
  if (calentador==1){
    document.getElementById('calentEstado').textContent = 'ENCENDIDO';
    console.log("calentador encendido");
  }
  if(calentador==0){
    document.getElementById('calentEstado').textContent = 'APAGADO';
    console.log("calentador apagado");
  }
  const bomba = parseFloat(text.split("Bomba de agua: ")[1]);
  console.log(bomba);
  if (bomba==1){
    document.getElementById('bombaEstado').textContent = 'ENCENDIDO';
    console.log("bomba encendida");
  }
  if(bomba==0){
```

```
document.getElementById('bombaEstado').textContent = 'APAGADO';
console.log("bomba apagada");
}
const alimentador = parseFloat(text.split("Alimentador de peces: ")[1]);
console.log(alimentador);
if (alimentador==1){
document.getElementById('alimEstado').textContent = 'ENCENDIDO';
console.log("alimentador encendido");
}
if(alimentador==0){
document.getElementById('alimEstado').textContent = 'APAGADO';
console.log("alimentador apagado");
}
const nivel = parseFloat(text.split("Nivel de agua: ")[1]);
console.log(nivel);
if (nivel==1){
document.getElementById('nivel').textContent = 'BIEN';
console.log("nivel ok");
}
if(nivel==0){
document.getElementById('nivel').textContent = 'BAJO';
console.log("nivel de agua bajo");
}
console.log("FIN Separador inicial");
// valores de los sensores
let oxigenoDisuelto = parseFloat(document.getElementById('oxigeno').innerText);
// Lógica para el calentador
if (temperatura < 20) {
actcalentador(1); // Enciende el calentador
} else if (temperatura > 28) {
actcalentador(0); // Apaga el calentador
}
// Lógica para la bomba de oxigenación
if (oxigenoDisuelto < 5) {
actbomba(1); // Enciende la bomba de oxigenación
} else {
actbomba(0); // Apaga la bomba de oxigenación
}
// Lógica para la renovación de agua
if (tds > 1000 || ph < 6.5 || ph > 8.0) {
actagua(1); // Abre la válvula para renovar agua
} else {
actagua(0); // Cierra la válvula
}
}
```

```
// Lógica para el alimentador de peces
// Ejemplo: Alimentar una vez al día a una hora específica
let horaActual = new Date().getHours();
console.log(horaActual);
if (horaActual === 12) { // Suponiendo que se alimenta a las 12 PM
  actalim(1);
}
}
// Llama a la función fetchDataFromArduino cada 5 segundos para actualizar la información en la
interfaz
setInterval(fetchDataFromArduino, 5000);
```

## Código PHP

### Para la conexión a la base de datos

```
<?php
$host = "localhost";
$dbname = "piscicultura";
$user = "root";
$pass = "";

try {
  $pdo = new PDO("mysql:host=$host;dbname=$dbname", $user, $pass);
  $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
  echo "Error en la conexión: " . $e->getMessage();
  exit();
}
```

### Para guardar contraseñas de manera cifrada en la base de datos

```
<?php
require_once 'conexion.php';

$sql = "SELECT * FROM usuarios";
$stmt = $pdo->prepare($sql);
$stmt->execute();
$users = $stmt->fetchAll(PDO::FETCH_ASSOC);

foreach ($users as $user) {
```

```
$id = $user['id'];
$contrasena = $user['contrasena'];
$contrasena_hash = password_hash($contrasena, PASSWORD_DEFAULT);

$update_sql = "UPDATE usuarios SET contrasena = ? WHERE id = ?";
$update_stmt = $pdo->prepare($update_sql);
$update_stmt->execute([$contrasena_hash, $id]);
}

echo "Las contraseñas de todos los usuarios se han actualizado con éxito.";
?>
```

## Código MySQL

### Código SQL para usuarios de la base de datos

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Servidor: 127.0.0.1
-- Tiempo de generación: 18-12-2023 a las 02:15:43
-- Versión del servidor: 10.4.32-MariaDB
-- Versión de PHP: 8.0.30

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de datos: `piscicultura`
--

-----

--
-- Estructura de tabla para la tabla `usuarios`
```

```
--  
  
CREATE TABLE `usuarios` (  
  `id` int(11) NOT NULL,  
  `usuario` varchar(255) NOT NULL,  
  `nombre` varchar(255) NOT NULL,  
  `contrasena` varchar(255) NOT NULL,  
  `acceso` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;  
  
--  
-- Volcado de datos para la tabla `usuarios`  
--  
  
INSERT INTO `usuarios` (`id`, `usuario`, `nombre`, `contrasena`, `acceso`) VALUES  
(1, 'Cesar', 'Cesar Rojas',  
'$2y$10$/y3jpYeZ3MKk4QCjuYxpughrHINUBpekz9enrG2XIG7CoFEi7qhC', 'Administrador');  
  
--  
-- Índices para tablas volcadas  
--  
  
--  
-- Indices de la tabla `usuarios`  
--  
ALTER TABLE `usuarios`  
  ADD PRIMARY KEY (`id`);  
  
--  
-- AUTO_INCREMENT de las tablas volcadas  
--  
  
--  
-- AUTO_INCREMENT de la tabla `usuarios`  
--  
ALTER TABLE `usuarios`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;  
COMMIT;  
  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

## Controlador Arduino Mega 2560

## Código para lectura del sensor de temperatura

```
#include <OneWire.h>
#include <DallasTemperature.h>
// Pin al que está conectado el sensor
#define ONE_WIRE_BUS 2
// Configuración de la comunicación OneWire
OneWire oneWire(ONE_WIRE_BUS);
// Pasar la referencia oneWire al Dallas Temperature
DallasTemperature sensors(&oneWire);
void setup(void) {
  // Iniciar comunicaciones
  Serial.begin(9600);
  sensors.begin();
}
void loop(void) {
  // Solicitar temperaturas (envía el comando para tomar lecturas de temperatura)
  sensors.requestTemperatures();
  // Leer temperatura en grados Celsius
  float tempC = sensors.getTempCByIndex(0);
  // Controlar un error de lectura del sensor
  if(tempC == DEVICE_DISCONNECTED_C) {
    Serial.println("Error: No se puede leer la temperatura del sensor");
  } else {
    Serial.print("Temperatura: ");
    Serial.print(tempC);
    Serial.println("°C");
  }
  // Esperar un segundo antes de leer nuevamente
  delay(1000);
}
```

## Código para lectura del sensor de ph

```
// Pin al que está conectado el sensor de pH
const int phPin = A0;
int buf[10];
int temp=0;
unsigned long int inValue;

void setup() {
  Serial.begin(9600);
}
/*para evitar demasiadas variaciones se calcula el promedio*/
```

```
for(int i=0; i<10;i++){
  buf[i]= analogRead(phPin);
  delay(10);
}
/* se descartan los valores demasiado elevados y bajos*/
for(int i=0; i<9; i++){
  for(int j=i+1;j<10;j++){
    temp= buf[i];
    buf[i]=buf[j];
    buf[j]=temp;
  }
}
/*calculo del promedio y la conversion a voltaje en mv*/
inValue=0;
for(int i=2; i<8; i++){
  inValue= inValue + buf[i];
}
float PHVol= (float)inValue*100*5/1024/6;
Serial.print("Voltaje mv =");
Serial.println(PHVol);
/*ecuacion del sensor*/
float PH= -0.0554*PHVol +30;
Serial.print("PH =");
Serial.println(PH);
delay(100);
```

### Código para lectura del sensor de turbidez de agua

```
#include "GravityTDS.h"
#define TdsSensorPin A1
GravityTDS gravityTds;
float temperatura = 25,tdsValue = 0;
void setup()
{
  Serial.begin(9600);
  gravityTds.setPin(TdsSensorPin);
  gravityTds.setAref(5.0); // voltaje de referencia en ADC, predeterminado 5.0V en Arduino UNO
  gravityTds.setAdcRange(1024); //1024 para ADC de 10 bits; 4096 para ADC de 12 bits
  gravityTds.begin(); //inicialización
}
void loop()
{
  //temperatura = readTemperature(); // agregar y leer sensor de temperatura
```

```
gravityTds.setTemperature(temperatura); // establecer y ejecutar la compensación de
temperatura
gravityTds.update(); // muestrear y calcular
tdsValue = gravityTds.getTdsValue(); // luego se obtiene el valor
Serial.print(tdsValue,0);
Serial.println("ppm");
delay(1000);
}
```

## Código para lectura del sensor de oxígeno disuelto en el agua

### Código para calibración del sensor

```
#include <Arduino.h>
#define VREF 5000//VREF(mv)
#define ADC_RES 1024//ADC Resolution
uint32_t raw;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  raw=analogRead(A1);
  Serial.println("raw:\t"+String(raw)+"\tVoltage(mv)"+String(raw*VREF/ADC_RES));
  delay(1000);
}
```

### Código principal

```
#include <Arduino.h>
#define DO_PIN A2
#define VREF 5000 //VREF (mv)
#define ADC_RES 1024 //ADC Resolution
//Single-point calibration Mode=0
//Two-point calibration Mode=1
#define TWO_POINT_CALIBRATION 0
#define READ_TEMP (25) //Current water temperature °C, Or temperature sensor function
//Single point calibration needs to be filled CAL1_V and CAL1_T
#define CAL1_V (1600) //mv
#define CAL1_T (25) //°C
//Two-point calibration needs to be filled CAL2_V and CAL2_T
//CAL1 High temperature point, CAL2 Low temperature point
```

```
#define CAL2_V (1300) //mv
#define CAL2_T (15) //°C
const uint16_t DO_Table[41] = {
    14460, 14220, 13820, 13440, 13090, 12740, 12420, 12110, 11810, 11530,
    11260, 11010, 10770, 10530, 10300, 10080, 9860, 9660, 9460, 9270,
    9080, 8900, 8730, 8570, 8410, 8250, 8110, 7960, 7820, 7690,
    7560, 7430, 7300, 7180, 7070, 6950, 6840, 6730, 6630, 6530, 6410};
uint8_t Temperaturet;
uint16_t ADC_Raw;
uint16_t ADC_Voltage;
uint16_t DO;
int16_t readDO(uint32_t voltage_mv, uint8_t temperature_c)
{
    #if TWO_POINT_CALIBRATION == 0
        uint16_t V_saturation = (uint32_t)CAL1_V + (uint32_t)35 * temperature_c - (uint32_t)CAL1_T * 35;
        return (voltage_mv * DO_Table[temperature_c] / V_saturation);
    #else
        uint16_t V_saturation = (int16_t)((int8_t)temperature_c - CAL2_T) * ((uint16_t)CAL1_V - CAL2_V) /
        ((uint8_t)CAL1_T - CAL2_T) + CAL2_V;
        return (voltage_mv * DO_Table[temperature_c] / V_saturation);
    #endif
}

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    Temperaturet = (uint8_t)READ_TEMP;
    ADC_Raw = analogRead(DO_PIN);
    ADC_Voltage = uint32_t(VREF) * ADC_Raw / ADC_RES;
    Serial.print("Temperaturet:\t" + String(Temperaturet) + "\t");
    Serial.print("ADC RAW:\t" + String(ADC_Raw) + "\t");
    Serial.print("ADC Voltage:\t" + String(ADC_Voltage) + "\t");
    Serial.println("DO:\t" + String(readDO(ADC_Voltage, Temperaturet)) + "\t");
    delay(1000);
}
```

### Código para accionar un actuador

```
// Código de ejemplo para activar y desactivar un actuador cada 5 minutos
// Pin al que está conectado el relé
const int relayPin = 8;
```

```
// Intervalo de tiempo (5 minutos en milisegundos)
const unsigned long interval = 300000; // 5 minutos * 60 segundos * 1000 milisegundos
unsigned long previousMillis = 0; // Almacena la última vez que se actualizó el relé
void setup() {
  pinMode(relayPin, OUTPUT); // Establece el pin del relé como salida
  digitalWrite(relayPin, LOW); // Inicia con el relé desactivado (suponiendo que LOW desactiva el
  relé)
}
void loop() {
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    // Guarda la última vez que se actualizó el relé
    previousMillis = currentMillis;
    // Si el relé está apagado, enciéndelo; si está encendido, apágalo
    if (digitalRead(relayPin) == LOW) {
      digitalWrite(relayPin, HIGH);
    } else {
      digitalWrite(relayPin, LOW);
    }
  }
}
```

### Código completo del sistema de control automático de estanque para cría de alevines

```
#include <Ethernet.h>
#include <SPI.h>
#include "GravityTDS.h"
#include <OneWire.h>
#include <DallasTemperature.h>
// Pin al que está conectado el sensor de temperatura
#define ONE_WIRE_BUS 2
// Configuración de la comunicación OneWire
OneWire oneWire(ONE_WIRE_BUS);
// Pasar la referencia oneWire al Dallas Temperature
DallasTemperature sensors(&oneWire);
// Pin al que está conectado el sensor de pH
const int phPin = A0;
int buf[10];
int temp=0;
unsigned long int inValue;
// Pin al que está conectado el sensor TDS
#define TdsSensorPin A1
GravityTDS gravityTds;
```

```
float tds = 0;
// Configuración de la interfaz Ethernet
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 100, 177); // IP del Arduino en tu red local
EthernetServer server(80);
//Sensor de oxigeno
#include <Arduino.h>
#define DO_PIN A2
#define VREF 5000 //VREF (mv)
#define ADC_RES 1024 //ADC Resolution
//Single-point calibration Mode=0
//Two-point calibration Mode=1
#define TWO_POINT_CALIBRATION 0
#define READ_TEMP (25) //Current water temperature °C, Or temperature sensor function
//Single point calibration needs to be filled CAL1_V and CAL1_T
#define CAL1_V (1600) //mv
#define CAL1_T (25) //°C
//Two-point calibration needs to be filled CAL2_V and CAL2_T
//CAL1 High temperature point, CAL2 Low temperature point
#define CAL2_V (1300) //mv
#define CAL2_T (15) //°C
const uint16_t DO_Table[41] = {
    14460, 14220, 13820, 13440, 13090, 12740, 12420, 12110, 11810, 11530,
    11260, 11010, 10770, 10530, 10300, 10080, 9860, 9660, 9460, 9270,
    9080, 8900, 8730, 8570, 8410, 8250, 8110, 7960, 7820, 7690,
    7560, 7430, 7300, 7180, 7070, 6950, 6840, 6730, 6630, 6530, 6410};
uint8_t Temperature;
uint16_t ADC_Raw;
uint16_t ADC_Voltage;
uint16_t DO;
float DOmg;
int16_t readDO(uint32_t voltage_mv, uint8_t temperature_c)
{
    #if TWO_POINT_CALIBRATION == 0
        uint16_t V_saturation = (uint32_t)CAL1_V + (uint32_t)35 * temperature_c - (uint32_t)CAL1_T * 35;
        return (voltage_mv * DO_Table[temperature_c] / V_saturation);
    #else
        uint16_t V_saturation = (int16_t)((int8_t)temperature_c - CAL2_T) * ((uint16_t)CAL1_V - CAL2_V) /
        ((uint8_t)CAL1_T - CAL2_T) + CAL2_V;
        return (voltage_mv * DO_Table[temperature_c] / V_saturation);
    #endif
}
//Fin sensor de oxigeno
const int calentPin = 3; // El pin donde está conectado el calentador
```

```
const int bombaPin = 4; // El pin donde está conectada la bomba
const int aguaextPin = 5; // El pin donde está conectada la bomba de agua potable para renovacion
de agua del estanque
const int alimPin = 13; // El pin donde está conectado el alimentador
int calentador;
int bomba;
int alimentador;
int nivel;
int valvula;
void setup() {
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.begin(9600);
  sensors.begin();
  gravityTds.setPin(TdsSensorPin);
  gravityTds.setAref(5.0); //reference voltage on ADC, default 5.0V on Arduino UNO
  gravityTds.setAdcRange(1024); //1024 for 10bit ADC;4096 for 12bit ADC
  gravityTds.begin(); //initialization
  pinMode(calentPin, OUTPUT); // Inicializa el pin del calentador como salida
  digitalWrite(calentPin, LOW); // Asegúrate de que el calentador esté apagado al inicio
  pinMode(bombaPin, OUTPUT); // Inicializa el pin de la bomba oxigenadora como salida
  digitalWrite(bombaPin, LOW); // Asegúrate de que la bomba oxigenadora esté apagado al inicio
  pinMode(aguaextPin, OUTPUT); // Inicializa el pin del agua externa como salida
  digitalWrite(aguaextPin, LOW); // Asegúrate de que el rele o valvula del agua externa esté
apagado al inicio
  pinMode(alimPin, OUTPUT); // Inicializa el pin del alimentador como salida
  digitalWrite(alimPin, LOW); // Asegúrate de que el alimentador esté apagado al inicio
}
void loop() {
//Valores de temperatura
sensors.requestTemperatures();
// Leer temperatura en grados Celsius
float tempC = sensors.getTempCByIndex(0);
// Controlar un error de lectura del sensor
if(tempC == DEVICE_DISCONNECTED_C) {
  Serial.println("Error: No se puede leer la temperatura del sensor");
} else {
  Serial.print("Temperatura: ");
  Serial.print(tempC);
  Serial.println("°C");
}
// Esperar un segundo antes de leer nuevamente
delay(1000);
//Fin Valores de temperatura
```

```
//Valores pH
for(int i=0; i<10;i++){
  buff[i]= analogRead(phPin);
  delay(10);
}
for(int i=0; i<9; i++){
  for(int j=i+1;j<10;j++){
    temp= buff[i];
    buff[i]=buff[j];
    buff[j]=temp;
  }
}
inValue=0;
for(int i=2; i<8; i++){
  inValue= inValue + buff[i];
}
float PHVol= (float)inValue*100*5/1024/6;
Serial.print("Voltaje mv =");
Serial.println(PHVol);
float PH= -0.0554*PHVol +30;
Serial.print("PH =");
Serial.println(PH);
delay(100);
//Fin Valores pH
//Inicio valores de turbidez
gravityTds.setTemperature(sensors.getTempCByIndex(0)); // set the temperature and execute
temperature compensation
gravityTds.update(); //sample and calculate
tds = gravityTds.getTdsValue(); // then get the value
Serial.print(tds,0);
Serial.println(" ppm");
Serial.print("Temperature is: ");
Serial.println(sensors.getTempCByIndex(0));
//Fin valores de turbidez
//Inicio valores de oxigeno disuelto
Temperaturet = tempC;
ADC_Raw = analogRead(DO_PIN);
ADC_Voltage = uint32_t(VREF) * ADC_Raw / ADC_RES;
DO = readDO(ADC_Voltage, Temperaturet);
DOMg = DO/1000;
Serial.print("Temperaturet:\t" + String(Temperaturet) + "\t");
Serial.print("ADC RAW:\t" + String(ADC_Raw) + "\t");
Serial.print("ADC Voltage:\t" + String(ADC_Voltage) + "\t");
Serial.println("DO:\t" + String(DO) + " ug/L\t");
```

```
delay(1000);
//Fin valores de oxigeno disuelto
// Escucha a los clientes entrantes
EthernetClient client = server.available();
if (client) {
  // Una petición HTTP termina con una línea en blanco
  boolean currentLineIsBlank = true;
  while (client.connected()) {
    if (client.available()) {
      char c = client.read();
      // Si se llegó al final de la línea (recibe un salto de línea)
      // si la línea está en blanco, es el fin de la petición HTTP
      if (c == '\n' && currentLineIsBlank) {
        // Envía una respuesta HTTP estándar
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println("Access-Control-Allow-Origin: *"); // Permite solicitudes de cualquier origen
        client.println("Connection: close");
        client.println();
        // página web
        client.println("<!DOCTYPE html>");
        client.println("<html lang='es'>");
        client.println("<head>");
        client.println(" <meta charset='UTF-8'>");
        client.println(" <meta name='viewport' content='width=device-width, initial-scale=1.0'>");
        client.println(" <title>Estanque de cría de peces</title>");
        client.println(" <link rel='stylesheet' href='http://localhost/piscicultura/styles.css'>");
        client.println("</head>");
        client.println("<body>");
        client.println(" <h1>Monitoreo de estanque para cría de peces</h1>");
        client.print("Calentador: ");
        client.print(calentador);
        client.print("Temperatura: ");
        client.print(tempC);
        client.print("°C");
        client.print("Nivel de agua: ");
        client.print(nivel);
        client.print("Bomba de agua: ");
        client.print(bomba);
        client.print("Valor de pH: ");
        client.print(PH);
        client.print("Valor de tds: ");
        client.print(tds);
        client.print("Oxigeno disuelto: ");
```

```
client.print(DO);
client.print("ug/L");
client.print("Alimentador de peces: ");
client.print(alimentador);
client.println("<h1>Alimentador Control</h1>");
client.println("<a href='/ALIM=ON'>Encender Alimentador</a><br>");
client.println("<a href='/ALIM=OFF'>Apagar Alimentador</a>");
client.println("</body>");
client.println("</html>");
break;
}
if (c == '\n') {
    // Estás al comienzo de una nueva línea
    currentLineIsBlank = true;
} else if (c != '\r') {
    // Tienes un carácter en la línea
    currentLineIsBlank = false;
}
}
}
// Verifica si la petición es para encender el alimentador
if (client.available()) {
    String request = client.readStringUntil('\r');
    Serial.println(request);
    client.flush();
    if (request.indexOf("/ALIM=ON") != -1) {
        digitalWrite(alimPin, HIGH); // Enciende el LED
        alimentador=1;
        Serial.println("Alimentador encendido");
    }
    if (request.indexOf("/ALIM=OFF") != -1) {
        digitalWrite(alimPin, LOW); // Apaga el LED
        alimentador=0;
        Serial.println("Alimentador apagado");
    }
    if (request.indexOf("/BOM=ON") != -1) {
        digitalWrite(bombaPin, HIGH); // Enciende el LED
        bomba=1;
        Serial.println("Bomba encendida");
    }
    if (request.indexOf("/BOM=OFF") != -1) {
        digitalWrite(bombaPin, LOW); // Apaga el LED
        bomba=0;
        Serial.println("Bomba apagada");
    }
}
```

```
if (request.indexOf("/CAL=ON") != -1) {
  digitalWrite(calentPin, HIGH); // Enciende el LED
  calentador=1;
  Serial.println("Calentador encendido");
}
if (request.indexOf("/CAL=OFF") != -1) {
  digitalWrite(calentPin, LOW); // Apaga el LED
  calentador=0;
  Serial.println("Calentador apagado");
}
if (request.indexOf("/VAL=ON") != -1) {
  digitalWrite(aguaextPin, HIGH); // Enciende el LED
  valvula=1;
  Serial.println("Agua externa encendida");
}
if (request.indexOf("/VAL=OFF") != -1) {
  digitalWrite(aguaextPin, LOW); // Apaga el LED
  valvula=0;
  Serial.println("Agua externa apagada");
}
}
}
// Da tiempo para que el navegador reciba los datos
delay(1);
client.stop();
}
}
```

## Instrumento de recolección de datos

### **¿Cuáles son las dificultades que se aprecian en el centro?**

Falta más recurso humano para realizar toda la actividad debido a que se cuenta con varias piletas que se deben controlar los parámetros del agua a diario y eso muchas veces no se logran con precisión

### **¿Cuáles son las variables que se controlan en la cría de alevines?**

Las variables que se controlan son el ph del agua, oxígeno, la temperatura, la densidad de siembra de matrices, turbidez del agua

### **¿Cantidad de equipos con la que cuentan?**

Cuentan con 8 equipos

### **¿Cuáles son esos equipos?**

Calentadores de agua, compresor, generador, aireadores de paletas, aireadores sumergibles, disco de secchi, microscopio, oxímetro, phmetro o Peachimetro, balanza

### **¿Desde cuándo cuentan con esos equipos?**

Desde que se inició en 2010

### **¿Cuáles son los instrumentos de medición de las variables?**

Los instrumentos de medición de dichas variables son Oxímetro, Peachimetro, termómetro, disco de secchi

### **¿Las mediciones de las variables se hacen en forma manual o automática?**

Se realizan en forma manual El control de los parámetros se realiza introduciendo los equipos en el agua los cuales indican en número los valores de las variables mencionadas y en caso necesario se toman las acciones requeridas para su corrección.

### **¿Cuáles son los mecanismos de control de los parámetros de las variables y la periodicidad de los mismos**

La temperatura con termómetro, disco de secchi la turbidez del agua, oxígeno con oxímetro, pH con peachimetro

### **¿De estos parámetros que ustedes controlan qué es lo que más le produce dificultad para obtener un control preciso y que está produciendo la muerte de sus alevines?**

La temperatura es uno de los parámetros físicos más importante en el agua pues por lo general influye en el retardo o aceleración de la actividad biológicas. El rango óptimo es de 25°-30 °c. en la que existe mayor desove de tilapias. Cuando disminuye por debajo de los 24°C, los peces dejan de comer y pueden presentar algunas enfermedades

### **¿Cómo ustedes corrigen o realizan esa corrección si hay cuando detectan el parámetro fuera de rango?**

Lo recomendable es que el ph del agua esté entre el promedio de 6 y 8 de alcalinidad en este caso el agua es alcalina por tanto solo se procura mantener a este rango de alcalinidad es decir no hay preocupación con el tema de acidez del agua, para corregir esto se le agrega 150 gramo de cal en metro cuadrado.

El oxígeno agregando más oxigenadores a través de aire de compresor y aireadores sumergibles, los compresores son utilizados para generar oxígeno.

La temperatura con calentador de agua, se coloca invernaderos para mantener a un valor deseado de un rango de 25°-30°C.

La turbidez del agua se corrige haciendo que fluye más cantidad de agua

**¿Como se llama el equipo utilizado para corregir la anomalía?**

Peachimetro, calentador de agua, disco de secchi, compresor

**¿Cuáles serían algunas de las alternativas para mejorar la producción?**

Motor de agua, calentador de agua para obtener la temperatura deseada que es de 25°-30° Celsius, incorporar más técnicos.

**¿Para la cría de alevines cuales son las variables que se desea controlar**

El ph del agua, el oxígeno, la turbidez del agua, la temperatura

**¿Cuál es la mecánica que se aplica para obtener la información**

Se realizan diariamente con observación directa, introduciendo los equipos en el agua los cuales indican en número los valores de las variables mencionadas, la temperatura debe estar en un promedio de 25°C.

El nivel de ph en un rango de 6 a 8 de alcalinidad

La turbidez del agua se mide con el disco de secchi

El oxígeno con el oxímetro

**¿Cómo se mide la densidad de siembra de matrices (que acción se realiza) y cuál es su valor óptimo?**

Se realiza en forma manual normalmente 4 matrices en 1 metro cuadrado

**¿Cuál es la unidad de medida del disco secchi, las horas del día en que se debe medir y el valor deseado de la turbidez del agua?**

La unidad de medida es en centímetros, se realiza una vez a la semana antes del mediodía y el valor deseado es de 30 cm