



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

**TITULO DEL TRABAJO DE TITULACIÓN**

**SISTEMA WEB PARA MONITOREO Y GESTIÓN DE ALIMENTACIÓN DE  
LARVAS DE CAMARÓN**

**AUTOR**

**Vera Pozo, Samuel Angelo**

**EXAMEN COMPLEXIVO**

**Previo a la obtención del grado académico en  
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN**

**TUTOR**

**Ing. Sánchez León Carlos, Mgt.**

**Santa Elena, Ecuador**

**Año 2025**



**UPSE**

**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

**TRIBUNAL DE SUSTENTACIÓN**



---

Ing. José Sánchez Aquino, Mgt.  
**DIRECTOR DE LA CARRERA**



---

Ing. Carlos Sánchez León, Mgt  
**TUTOR**



---

Ing. Mónica Jaramillo Infante, Mgt  
**DOCENTE ESPECIALISTA**



---

Ing. Marjorie Coronel Suárez, Mgt.  
**DOCENTE GUÍA UIC**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

**CERTIFICACIÓN**

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por SAMUEL ANGELO VERA POZO, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

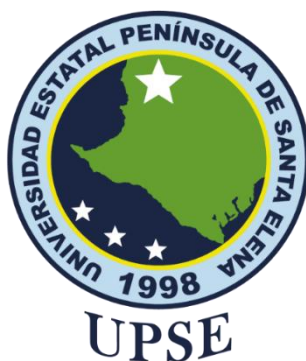
La Libertad, a los 20 días del mes de Junio del año 2025

**TUTOR**



---

**Ing. Carlos Sánchez León, Mgt.**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA**

**FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

**DECLARACIÓN DE RESPONSABILIDAD**

**Yo, SAMUEL ANGELO VERA POZO**

**DECLARO QUE:**

El trabajo de Titulación, Sistema Web para monitoreo y gestión de Alimentación de Larvas de Camarón previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

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

La Libertad, a los 20 días del mes de Junio del año 2025

**EL AUTOR**

A handwritten signature in black ink, appearing to read "Samuel Angelo Vera Pozo", is written over a horizontal line.

**Samuel Angelo Vera Pozo**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA**

**FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

**CERTIFICACIÓN DE ANTIPLAGIO**

Certifico que después de revisar el documento final del trabajo de titulación denominado **SISTEMA WEB PARA MONITOREO Y GESTIÓN DE ALIMENTACIÓN DE LARVAS DE CAMARÓN**, presentado por el estudiante, VERA POZO SAMUEL ANGELO fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 5%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



CERTIFICADO DE ANÁLISIS  
magister

VERAPOZO\_SAMUELANGEL  
O

**5%**  
Textos  
sospechosos

**2% Similitudes**  
< 1% similitudes entre comillas  
< 1% entre las fuentes mencionadas

**3% Idiomas no reconocidos**

**10% Textos potencialmente generados por la IA (ignorado)**

Nombre del documento: VERAPOZO_SAMUELANGELO.docx	Depositante: CARLOS EFRAIN SANCHEZ LEON	Número de palabras: 13.158
ID del documento: 73a28262e0075c65f8822cf70c248fcb85560755	Fecha de depósito: 20/6/2025	Número de caracteres: 91.137
Tamaño del documento original: 4,34 MB	Tipo de carga: interface	
	fecha de fin de análisis: 20/6/2025	

**TUTOR**



Firmado electrónicamente por:  
**CARLOS EFRAIN  
SANCHEZ LEON**

Validar únicamente con FirmaRC

**Ing. Carlos Sánchez León, Mgt.**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

**AUTORIZACIÓN**

**Yo, VERA POZO SAMUEL ANGELO**

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

Cedo los derechos en línea patrimoniales del presente trabajo de titulación con fines de difusión pública, además apruebo la reproducción dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 20 días del mes de Junio del año 2025

**EL AUTOR**

A handwritten signature in black ink, appearing to read "Samuel Angelo Vera Pozo", is written over a horizontal line.

**Samuel Angelo Vera Pozo**

## AGRADECIMIENTO

Con la ayuda de Dios, finalizar esta etapa universitaria ha sido el resultado de años de trabajo constante, decisiones difíciles y aprendizajes acumulados. Agradezco a quienes hicieron posible que pudiera concentrarme en lo importante: avanzar. A mi familia, especialmente a mis padres Eladio Vera y Concepción Pozo, por su respaldo práctico y por confiar en mi capacidad para terminar lo que empecé. A los profesores que aportaron con claridad, exigencia y criterio.

Extiendo también mi agradecimiento al Ing. Carlos Sánchez León, tutor en este proceso, por su orientación y compromiso durante el desarrollo del proyecto. A mis compañeros, con quienes compartí más que clases: ideas, errores y soluciones. También a quienes, desde fuera, me dieron una mano cuando se necesitaba. Este cierre es el reflejo de un esfuerzo colectivo, aunque la responsabilidad siempre fue mía.

*Samuel Angelo, Vera Pozo*

## DEDICATORIA

A Dios, por darme la fuerza, la paciencia y la claridad en los momentos en que el camino se volvió incierto. Nada de esto hubiera sido posible sin Su guía y respaldo constante. A mi familia, especialmente a mis padres Eladio Vera y Concepción Pozo, por su apoyo firme, su fe inquebrantable en mí y por enseñarme, con hechos, el valor del compromiso, el trabajo honesto y la perseverancia.

A quienes no siempre estuvieron en primer plano, pero sí aportaron con palabras, gestos o simplemente con su presencia cuando más se necesitaba. Esta etapa también se construyó con esas pequeñas acciones que, sumadas, hicieron una gran diferencia. Y me la dedico a mí mismo, porque supe sostener el ritmo, tomar decisiones difíciles y continuar aun cuando el cansancio o la duda quisieron imponerse. Este logro no es casualidad: es el resultado de mantenerse firme y avanzar, un paso a la vez.

*Samuel Angelo Vera Pozo*

## ÍNDICE GENERAL

TRIBUNAL DE SUSTENTACIÓN	II
CERTIFICACIÓN	III
DECLARACIÓN DE RESPONSABILIDAD	IV
CERTIFICACIÓN DE ANTIPLAGIO	V
AUTORIZACIÓN	VI
AGRADECIMIENTO	VII
DEDICATORIA	VIII
ÍNDICE DE TABLAS	XII
ÍNDICE DE FIGURAS	XIII
RESUMEN	XV
ABSTRACT	XVI
INTRODUCCIÓN	2
CAPÍTULO 1. FUNDAMENTACIÓN	4
1.1 Antecedentes	4
1.2 Descripción del Proyecto	7
1.3 Objetivos del Proyecto	9
1.4 Justificación del Proyecto	9
1.5 Alcance del Proyecto	10
CAPÍTULO 2. MARCO TEÓRICO Y METODOLOGÍA DEL PROYECTO	13
2.1 Marco Conceptual	13

2.1.1 Cultivo de larvas de camarón	13
2.1.2 Parámetros ambientales	13
2.1.3 Aplicación Web	14
2.1.4 Angular	14
2.1.5 Angular Material	15
2.1.6 Base de datos	15
2.1.7 Lenguajes de programación	16
2.1.8 Visual Studio Code	16
2.1.9 XAMPP	17
2.1.10 PhpMyAdmin	17
2.1.11 Apache HTTP	17
2.1.12 MySQL	18
2.1.13 NodeJS	18
2.1.14 API-REST	19
2.1.15 EmailJS	19
2.1.16 Bootstrap	20
2.1.17 Postman	21
2.2 Marco Teórico	21
2.2.1 Impacto de la Nutrición en el Cultivo de Larvas de Camarón	21
2.2.2 Importancia de los Sistemas de Gestión y Monitoreo en Acuicultura	22
2.2.3 Angular como framework en el desarrollo de aplicaciones web	23

2.3 Metodología del Proyecto	24
2.3.1 Metodología de la Investigación	24
2.3.2 Técnicas e instrumentos de recolección de datos	25
2.3.3 Metodología de desarrollo	25
CAPÍTULO 3. PROPUESTA	28
3.1 Requerimientos	28
3.1.1 Requerimientos Funcionales	28
3.1.2 Requerimientos No Funcionales	32
3.2 Componentes de la Propuesta	32
3.2.1 Arquitectura del Sistema	32
3.2.2 Diagramas de caso de Uso	33
3.2.3 Modelado de Datos	43
3.3 Diseño de Interfaces	45
3.4 Pruebas	56
CONCLUSIONES	66
RECOMENDACIONES	67
ANEXO	73

## ÍNDICE DE TABLAS

<i>Tabla 1: Requerimientos Funcionales.....</i>	<i>31</i>
<i>Tabla 2: Requerimientos No Funcionales.....</i>	<i>32</i>
<i>Tabla 3: Caso de Uso. Autenticación &amp; Acceso.....</i>	<i>35</i>
<i>Tabla 4: Caso de Uso. Gestión de Usuarios.....</i>	<i>36</i>
<i>Tabla 5: Caso de Uso. Módulos &amp; Tanques.....</i>	<i>37</i>
<i>Tabla 6: Caso de Uso. Corridas.....</i>	<i>38</i>
<i>Tabla 7: Caso de Uso. Gestión de Siembras.....</i>	<i>39</i>
<i>Tabla 8: Caso de Uso. Cosechas.....</i>	<i>40</i>
<i>Tabla 9: Caso de Uso. Monitoreo.....</i>	<i>41</i>
<i>Tabla 10: Caso de Uso. Planes Alimenticios.....</i>	<i>42</i>
<i>Tabla 11: Caso de Uso. Reportes &amp; Análisis.....</i>	<i>43</i>
<i>Tabla 12: Prueba - Inicio Sesión.....</i>	<i>57</i>
<i>Tabla 13: Prueba - Gestión de Usuarios.....</i>	<i>58</i>
<i>Tabla 14: Prueba - Módulos y Tanques.....</i>	<i>59</i>
<i>Tabla 15: Prueba - Corridas.....</i>	<i>60</i>
<i>Tabla 16: Prueba - Siembras.....</i>	<i>61</i>
<i>Tabla 17: Prueba - Cosechas.....</i>	<i>62</i>
<i>Tabla 18: Prueba - Monitoreo.....</i>	<i>63</i>
<i>Tabla 19: Prueba - Planes Alimenticios.....</i>	<i>64</i>
<i>Tabla 20: Prueba - Reportes y Análisis.....</i>	<i>65</i>

## ÍNDICE DE FIGURAS

<i>Figura 1: Árbol de Problema.....</i>	<i>5</i>
<i>Figura 2: Metodología de Desarrollo.....</i>	<i>27</i>
<i>Figura 3: Arquitectura del Sistema.....</i>	<i>33</i>
<i>Figura 4: Caso de Uso. Sistema .....</i>	<i>33</i>
<i>Figura 5: Caso de Uso. Autenticación Acceso .....</i>	<i>34</i>
<i>Figura 6: Caso de Uso. Gestión de Usuarios .....</i>	<i>35</i>
<i>Figura 7: Caso de Uso. Módulos &amp; Tanques.....</i>	<i>36</i>
<i>Figura 8: Caso de Uso. Corridas .....</i>	<i>37</i>
<i>Figura 9: Caso de Uso. Gestión de Siembras.....</i>	<i>38</i>
<i>Figura 10: Caso de Uso. Gestión de Cosechas .....</i>	<i>39</i>
<i>Figura 11: Caso de Uso. Monitoreo .....</i>	<i>40</i>
<i>Figura 12: Caso de Uso. Planes Alimenticios .....</i>	<i>41</i>
<i>Figura 13: Caso de Uso. Reportes &amp; Análisis.....</i>	<i>42</i>
<i>Figura 14: Modelado de Base de Datos .....</i>	<i>44</i>
<i>Figura 15: Interfaz – Inicio de Sesión .....</i>	<i>45</i>
<i>Figura 16: Interfaz - Restablecimiento de Contraseña .....</i>	<i>46</i>
<i>Figura 17: Interfaz - Nueva Contraseña.....</i>	<i>46</i>
<i>Figura 18: Interfaz – Dashboard Biólogo .....</i>	<i>47</i>
<i>Figura 19: Interfaz - Dashboard Gerente.....</i>	<i>47</i>
<i>Figura 20: Interfaz - Gestión de Usuarios.....</i>	<i>48</i>
<i>Figura 21: Interfaz - Registro de Usuario .....</i>	<i>48</i>
<i>Figura 22: Interfaz - Módulos y Tanque .....</i>	<i>49</i>
<i>Figura 23: Interfaz – Corridas .....</i>	<i>49</i>

<i>Figura 24: Interfaz – Siembras</i> .....	50
<i>Figura 25: Interfaz – Cosechas</i> .....	50
<i>Figura 26: Interfaz - Registro Monitoreo</i> .....	51
<i>Figura 27: Interfaz - Reporte Monitoreo</i> .....	51
<i>Figura 28: Interfaz - Generar Plan Alimenticio</i> .....	52
<i>Figura 29: Interfaz - Modal Registro Plan Alimenticio</i> .....	52
<i>Figura 30: Interfaz - Reporte Plan Operativo</i> .....	53
<i>Figura 31: Interfaz - Historial de Planes</i> .....	53
<i>Figura 32: Interfaz - Reporte Técnico de Alimentación</i> .....	54
<i>Figura 33: Interfaz - Reporte de Deficiencias</i> .....	54
<i>Figura 34: Interfaz - Reporte Rendimiento Biólogo</i> .....	55
<i>Figura 35: Interfaz Reporte Rendimiento Gerente</i> .....	55
<i>Figura 36: Interfaz - Reporte Costo Alimentación Cantidad</i> .....	56

## RESUMEN

El presente proyecto se centra en la fase crítica de cría de larvas de *Litopenaeus vannamei* en Ecuador, donde el control de temperatura, salinidad, oxígeno disuelto y pH, junto con una dieta ajustada a cada subestadio, resulta vital para maximizar la supervivencia y el rendimiento. Hoy estos registros y cálculos se realizan manualmente, lo que sobrecarga al personal, retrasa los ajustes nutricionales y eleva el riesgo de mortalidad y brotes de enfermedad.

Para solucionarlo, se propone una plataforma web integral que centralice el ingreso de parámetros y genere automáticamente planes alimenticios por subestadio. El sistema incluirá módulos de gestión de usuarios, corridas de producción, monitoreo diario, elaboración de dietas, generación de reportes en PDF y alertas ante desviaciones críticas. Con ello se optimizarán los procesos operativos, se mejorará la toma de decisiones en tiempo real para futuras mejoras tecnológicas, como la integración de sensores IoT y algoritmos predictivos., elevando la supervivencia larval y la eficiencia productiva.

**Palabras clave:** Acuicultura, Larvas de camarón, Monitoreo ambiental, Planes alimenticios, Plataforma web integrada

## ABSTRACT

The present project focuses on the critical phase of rearing *Litopenaeus vannamei* larvae in Ecuador, where control of temperature, salinity, dissolved oxygen and pH, together with a diet adjusted to each substage, is vital to maximize survival and performance. Today, these records and calculations are done manually—with logs and spreadsheets—which overburdens staff, delays nutritional adjustments, and increases the risk of mortality and disease outbreaks.

To solve this, a comprehensive web platform is proposed that centralizes the entry of parameters and automatically generates meal plans by substage. The system will include modules for user management, production runs, daily monitoring, diet preparation, PDF report generation and alerts in the event of critical deviations. This will optimize operational processes, improve real-time decision-making for future technological improvements, such as the integration of IoT sensors and predictive algorithms, increasing larval survival and production efficiency.

**Keywords:** Aquaculture, Shrimp larvae, Environmental monitoring, Feeding plans, Integrated web platform

## INTRODUCCIÓN

La acuicultura con un crecimiento exponencial en las últimas décadas consolidándose como una de las principales fuentes de proteína animal a nivel mundial. En Ecuador gracias a su geografía y condiciones climáticas lo ha posicionado como líder en la producción de camarón: (*Litopenaeus vannamei*), la etapa larvaria juega un rol importante para dar rentabilidad a la industria. Sin embargo, durante esta etapa crítica, la supervivencia de las larvas depende de un control preciso de variables ambientales: temperatura, salinidad, oxígeno disuelto, pH y de una planificación nutricional ajustada a cada subestadio de desarrollo.

En muchos laboratorios o centros de producción el registro de parámetros y la preparación de planes de alimentación se realizan aún de manera manual acudiendo a bitácoras escritas. Esta metodología dificulta la correlación inmediata entre las condiciones del agua y las raciones suministradas, lo que puede resultar en: desnutrición o sobrealimentación, crecimiento irregular, mayor mortalidad y retoños de enfermedades. La dispersión de datos y la falta de visibilidad integrada impiden ajustes oportunos y obstaculizan la toma de decisiones basada en evidencia.

Frente a este panorama, el proyecto propone el desarrollo de una plataforma web integral que centralice el registro de parámetros ambientales y la planificación técnica de dietas para larvas de camarón. Mediante módulos especializados (gestión de usuarios, corridas de producción, monitoreo diario, planes alimenticios y generación de reportes), el sistema facilitará el ingreso organizado de datos, aplicará cálculos automáticos de dosis y frecuencias de alimentación, y emitirá alertas o recomendaciones ante desviaciones. De este modo, se busca optimizar procesos operativos y reducir la carga de trabajo manual del personal.

La implementación de esta herramienta tecnológica no solo impulsa una transformación digital en la acuicultura larval, sino que también contribuye a el desempeño productivo de los cultivos. Al proporcionar información clara y oportuna, el sistema fortalece la capacidad de los biólogos y administradores para

tomar decisiones basadas en datos reales, disminuye la probabilidad de errores humanos y sienta las bases para futuras ampliaciones como: integración de sensores IoT o algoritmos predictivos que promuevan buenas prácticas y aseguren la competitividad del sector camaronero.

# CAPÍTULO 1. FUNDAMENTACIÓN

## 1.1 Antecedentes

La acuicultura se ha consolidado como una de las industrias alimentarias de mayor crecimiento a nivel global, Ecuador con el camarón ocupando un lugar destacado por su alta demanda en mercados internacionales [1]. Inicialmente la industria se basó en la captura de post-larvas silvestres hasta que en 1969 se establecieron las primeras unidades de cultivo intensivo en El Oro [2]. Para garantizar un suministro continuo de semilla, en 1981 abrió la hatchery Semacua, seguido por el laboratorio de la ESPOL, lo que permitió formar especialistas y desplazar la recolección silvestre [3]. Con esos avances, los centros de producción evolucionaron a complejos con sistemas de recirculación y monitoreo de parámetros ambientales, base de la eficiencia y sostenibilidad del sector camaronero ecuatoriano [3].

Durante la fase de cría de larvas de camarón, procesos como el registro de parámetros ambientales y la planificación de la alimentación adquieren particular relevancia para asegurar tasas óptimas de supervivencia [4]. En muchos laboratorios y centros de producción, se basan en métodos manuales y observaciones dificultando la identificación de variaciones en la calidad del agua y la adaptación de las raciones a las necesidades específicas de las larvas, situación puede derivar en deficiencias nutricionales, crecimiento irregular y mayor susceptibilidad a enfermedades [5].

Se realizó una observación en un centro de producción local de larvas de camarón (Anexo 1), se realiza la planificación alimenticia en experiencia tradicional y el monitoreo de parámetros como temperatura, salinidad y oxígeno disuelto se realiza a través de hojas de cálculo y bitácoras impresas. Si bien esta práctica puede funcionar para operaciones de menor escala, no siempre permite una correlación precisa entre el estado de los parámetros ambientales y la alimentación suministrada. Además, la carencia de un método unificado para registrar, comparar y analizar estos datos dificulta los ajustes nutricionales en el momento oportuno [6].

Entre las causas con mayor relevancia se ha identificado el registro manual de los datos, la ausencia de correlación directa con los planes alimenticios y la falta de visibilidad inmediata de los cambios en la calidad del agua. Como consecuencia, se observan efectos como el riesgo de subalimentación o sobrealimentación, el aumento en la mortalidad de las larvas y la posible aparición de enfermedades debido a condiciones ambientales inadecuadas o retrasos en la adaptación de raciones. En la (Figura 1) se presenta el árbol de problema, que ilustra cómo estas causas se relacionan y generan impactos negativos en la producción de larvas de camarón, enfatizando la importancia de vincular de forma integrada el monitoreo de parámetros ambientales con la planificación alimenticia.

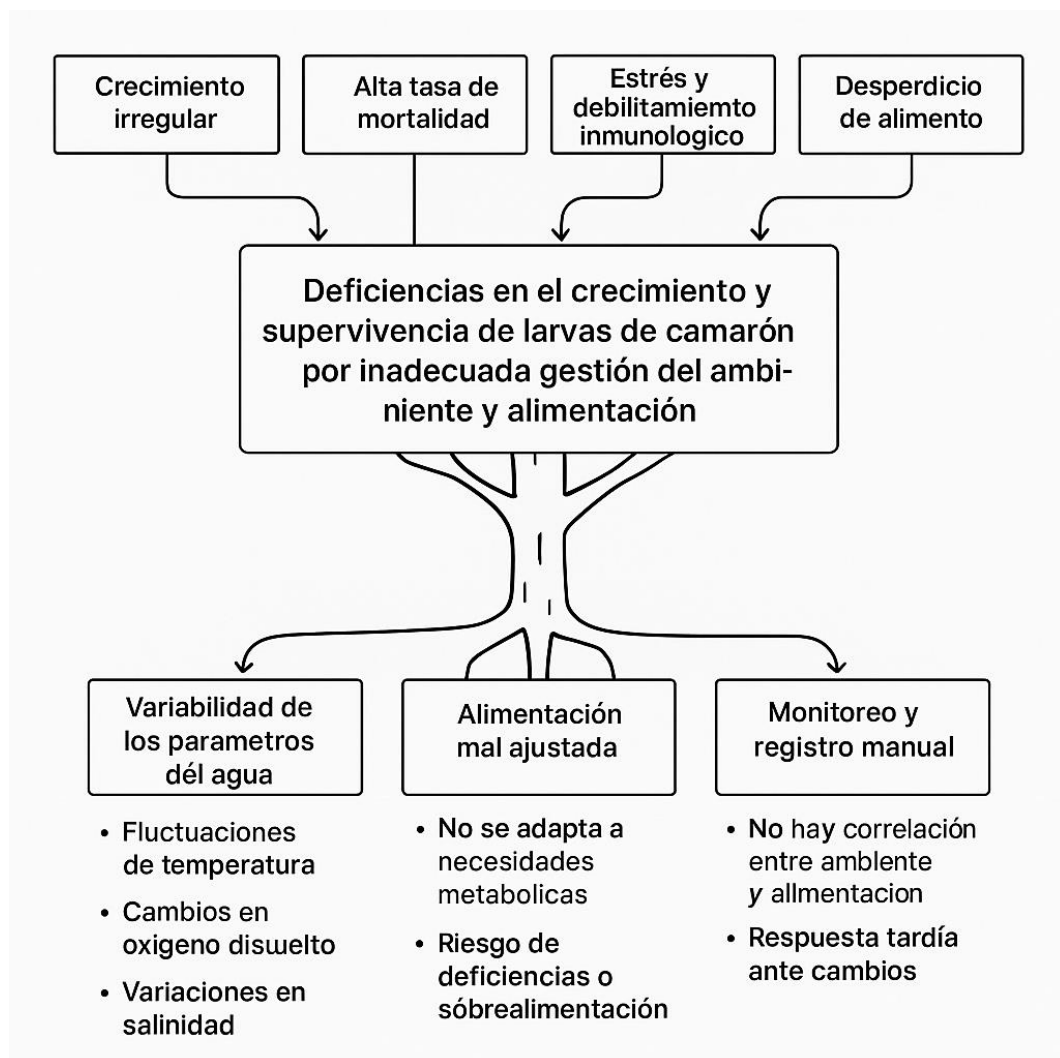


Figura 1: Árbol de Problema

Se han analizado varios documentos y proyectos provenientes de diferentes bases de datos, repositorios y artículos, que abordan el desarrollo de sistemas digitales aplicados a la acuicultura y, en particular, a la producción de larvas de camarón:

A nivel internacional, en Indonesia una investigación realizada por Heri Ariadi, 2Mohamad Fadjar, 3Mohammad Mahmudi, 3Supriatna “The relationships between water quality parameters and the growth rate of white shrimp (*Litopenaeus vannamei*) in intensive ponds” observó que los cambios en el oxígeno disuelto (OD) se relacionan con diversos factores de la calidad del agua, como temperatura, fosfato, nitrito y materia orgánica [7]. Cada estanque tenía un tamaño de 3,200 m<sup>2</sup>, una densidad de 110 postlarvas por metro cuadrado y utilizaba aireadores con 16 HP de potencia [7]. Al registrar de manera periódica tanto el OD como el nitrito, el estudio concluyó que el nitrito afecta en mayor medida el crecimiento de los camarones (75.2%) en comparación con otras variables. [7]

También en el contexto latinoamericano, destaca el trabajo de Susana Flores Mollo y Diego Aracena Pizarro, de la Universidad de Tarapacá en Chile, quienes diseñaron un sistema de monitoreo remoto para la crianza de camarones [8]. La información recolectada se podía consultar en tiempo real desde un dispositivo con acceso a internet, evitando así la dependencia de mediciones totalmente manuales [8]. Este enfoque permitió detectar variaciones en la calidad del agua de manera oportuna y, por ende, prevenir situaciones críticas para el cultivo [8]. Tales resultados refuerzan la idea de que la automatización y la centralización de datos son elementos clave para optimizar la producción camaronera en entornos de pequeña o mediana escala [8].

Mientras que, a nivel nacional, destaca el trabajo de Vera Rivera Christian Jazmani en la Universidad Estatal Península de Santa Elena, titulado “*Desarrollo de un sistema para medir y monitorear la calidad del agua de los tanques de producción de Larvas de camarón caso laboratorio de Larvas El Perla Negra del Mar* [9].” En esta propuesta se implementó un prototipo basado en tecnología IoT, con sensores capaces de medir la temperatura, el pH y la turbidez del agua en tanques utilizados para la cría de larvas de camarón [9]. Dichos datos se gestionan a través de un

sistema web, que permite la visualización en tiempo real y el envío de mensajes de alerta cuando los parámetros superan los rangos óptimos [9].

En síntesis, la dependencia de procedimientos manuales para el monitoreo de parámetros y la planificación de la alimentación obstaculiza la productividad y eleva el riesgo de complicaciones en la cría de larvas de camarón. Los antecedentes revisados, junto con la observación directa del laboratorio local, dejan en evidencia la necesidad de sistemas integrados que correlacionen datos ambientales y nutricionales en tiempo real. Mediante el diseño de una plataforma web capaz de gestionar planes alimenticios y monitorear parámetros críticos, se espera no solo optimizar la toma de decisiones y reducir la mortalidad.

## **1.2 Descripción del Proyecto**

El proyecto propuesto, es el desarrollo de un sistema web técnico para facilitar la gestión y monitoreo en la producción de larvas de camarón, enfocándose en la etapa crítica de desarrollo de las larvas. El sistema pretende en el desarrollo integral para la gestión eficiente de planes alimenticios basados en datos ingresados manualmente y el monitoreo diario dentro del laboratorio. El sistema permitirá registrar parámetros nutricionales, identificar desviaciones y generar planes de alimentación personalizados, con el fin de mejorar el control del proceso y apoyar la toma de decisiones técnicas sobre la alimentación de las larvas. Para el desarrollo de este proyecto se implementarán los siguientes módulos:

- **Gestión de Usuarios**

Este módulo permite la administración de los usuarios que interactúan con el sistema según su rol específico (biólogo, gerente, administrador).

Funciones Principales:

- Registro y Modificación de Usuarios.
- Asignación de Roles.
- Control de Acceso.

- **Gestión de Corridas (Siembras-Cosechas)**

Este módulo administra el ciclo completo desde la siembra de nauplios hasta la cosecha.

Funciones Principales:

- Registro de corridas
- Seguimiento de la clasificación del rendimiento (mala, regular, buena).

- **Monitoreo**

Este módulo apoya técnicamente al biólogo en registro y seguimiento de parámetros nutricionales críticos, utilizando fórmulas y cálculos basados en las observaciones.

Funciones Principales:

- Registro de condiciones diarias de parámetros ambientales del agua.
- Recomendaciones.
- Generación de observaciones.

- **Planes Alimenticios**

Este módulo gestiona la planificación técnica diaria de la alimentación para las larvas.

Funciones Principales:

- Planificación de dietas diarias según subestadios.
- Inclusión de suplementos y probióticos.
- Generación de un plan detallado en PDF para el operario

- **Reportes y Análisis (Informes)**

Genera reportes operativos específicos basados en los datos recolectados.

Funciones Principales:

- Reporte de Alimentación.
- Reporte de Deficiencias.
- Reporte de Rendimiento.
- Reporte de Cosechas.

### **1.3 Objetivos del Proyecto**

#### **Objetivo General**

Implementar un sistema web mediante herramientas tecnológicas Open Source que permita gestionar el proceso de alimentación de larvas de camarón basado en el monitoreo de parámetros ambientales y la planificación técnica de dietas.

#### **Objetivos Específicos:**

- Definir la arquitectura y el modelo de datos relacional necesarios para almacenar corridas de producción, parámetros ambientales y planes de alimentación.
- Desarrollar el módulo de monitoreo que permita el registro de parámetros ambientales y generación de acciones recomendadas ante desviaciones de parámetros.
- Desarrollar el módulo de planificación de dietas que permita el cálculo de dosis y frecuencias de alimentación para cada subestadio larval.
- Elaborar el módulo de reportes que proporcione información detallada del historial de corridas (siembras-cosechas) producidas.

### **1.4 Justificación del Proyecto**

El desarrollo de un sistema web para la gestión de planes alimenticios y el monitoreo nutricional de larvas de camarón surge de la necesidad de contar con una herramienta clara y práctica, que permita llevar un registro ordenado de parámetros ambientales, de suma importancia en el manejo de parámetros, como: (temperatura-salinidad-oxígeno disuelto-pH) y de los ajustes diarios en la alimentación [10], [11]. Esta propuesta basada en la tendencia global de impulsar una transformación digital que simplifique varios procesos y permita una mejora en cuanto a la exactitud de los datos relevantes para el éxito de los cultivos acuícolas [8].

Actualmente, muchos centros de producción de larvas de camarón aún dependen de métodos manuales, lo que dificulta la correlación precisa entre la información ambiental y la planificación de la alimentación. Esta falta de integración puede derivar en retrasos al momento de ajustar las raciones, con consecuencias negativas en el crecimiento y supervivencia de las larvas [7]. En respuesta a estos desafíos, la adopción de sistemas automatizados y plataformas web optimiza el registro de parámetros críticos y reduce la probabilidad de errores humanos [9].

La herramienta tecnológica que se propone no solo representa una solución concreta para la industria larvera, sino que también contribuye a mejorar la eficiencia operativa y reducir costos asociados con la toma de decisiones tardías [8]. Investigaciones recientes señalan el uso de un monitoreo inteligente disminuye el riesgo de errores en la administración de la alimentación, al tiempo que ofrece un mejor control sobre la calidad del agua, esto a su vez aumenta la productividad y fortalece la competitividad del sector [11], [12].

El proyecto busca impulsar un cambio positivo en la forma de gestionar la producción de larvas de camarón, centralizando y automatizando el acceso a datos críticos. Este paso no solo beneficia al personal técnico y administrativo, sino que también sienta las bases para futuras implementaciones de mayor alcance, fomentando buenas prácticas y procesos sostenibles en la industria [9]. Con ello, se refuerza la eficiencia de la cadena productiva y se promueve un crecimiento más ordenado y seguro en el sector camaronero.

### **1.5 Alcance del Proyecto**

El presente proyecto consiste en desarrollar un sistema web enfocado en la gestión de planes alimenticios, el mismo se basa en el ingreso manual y continuo de datos de monitoreo dentro de un laboratorio. Los datos abarcan parámetros ambientales y nutricionales (temperatura-oxígeno-disuelto-salinidad), la planificación y ajuste de las dietas diarias destinadas a las larvas de camarón. A través de esta herramienta, se busca centralizar la información en un solo entorno digital, optimizar la manera en que se registran las condiciones del cultivo y facilitar la generación de

recomendaciones oportunas, contribuyendo a una mejor toma de decisiones en el ámbito acuícola.

El proyecto contempla los siguientes módulos:

**Módulo de Gestión de Usuarios:** en este módulo se considera el acceso al sistema para los siguientes roles administrador, gerente y biólogo, se validan que los datos sean correctos para la correcta autenticación. Se permite el restablecimiento de contraseña en caso de olvido o cambio. Diseñado para la creación de nuevos usuarios, el registro de datos requeridos, se validan correos que no se repitan, se confirman mediante notificación por email la creación de la cuenta, permite también la modificación de datos de usuario y la eliminación de cuenta. Sin embargo, el registro se limitará a este método tradicional, **no** se implementará registro mediante redes sociales u otros servicios externos, ni se contemplará la verificación de identidad a través de documentos oficiales.

**Módulo de Gestión de Corridas (Siembras-Cosechas):** el módulo está diseñado para la administración del ciclo completo de producción de larvas de camarón, desde la siembra hasta la cosecha. Facilita el registro de cada corrida y la clasificación del rendimiento (mala, regular o buena), brindando un historial organizado para su seguimiento.

**Módulo de Monitoreo:** este modulo registra de manera diaria los parámetros de calidad del agua (temperatura, oxígeno disuelto, salinidad, entre otros), con la posibilidad de ofrecer recomendaciones para ajustar la alimentación y suplementación según las necesidades nutricionales de las larvas. Adicional un listado de monitoreos mediante filtros con opción a generación de un pdf.

**Módulo de Planes Alimenticios:** módulo diseñado para la elaboración de planes de alimentación, basándose en la información recabada en el monitoreo. Permite programar dietas específicas para cada etapa de desarrollo, además de generar documentos en PDF para uso del operario.

**Módulo de Reportes y Análisis:** el módulo genera reportes operativos y de rendimiento en función de los datos recolectados. Contempla informes sobre alimentación, deficiencias nutricionales, clasificación de corridas y resultados de cosechas, contribuyendo a una visión clara del proceso productivo.

## **CAPÍTULO 2. MARCO TEÓRICO Y METODOLOGÍA DEL PROYECTO**

### **2.1 Marco Conceptual**

#### **2.1.1 Cultivo de larvas de camarón**

El cultivo de larvas de camarón (*Litopenaeus vannamei*) empieza con la incubación del huevo este se completa aproximadamente 16 horas después del desove y la fertilización, iniciando la primera etapa larval denominada nauplio [13]. El nauplio se transforma en zoea después de unos días pasando a un subestadio en los que ya puede filtrar fitoplancton y ampliar su capacidad natatoria mediante movimiento circular constante [14]. En el estadio mysis las larvas nadan hacia atrás con sacudidas de la cola y comienzan a alimentarse también de zooplancton, antes de llegar al estado postlarval adquiriendo una forma muy similar al camarón juvenil y migrar hacia estanques de engorde [15].

La cría larvaria en laboratorios requiere condiciones controladas de densidad, calidad de agua y alimentación [16]. Estudios han demostrado que incorporar huevos de *Artemia* en dieta mejora significativamente la tasa de supervivencia con el crecimiento y la resistencia al estrés de las larvas en comparación con dietas compuestas exclusivamente de pienso artificial [17]. Además, mantener densidades moderadas (50-100larvas/mL) reduce la competencia por alimento y la acumulación de desechos tóxicos, contribuyendo a una mayor eficiencia alimenticia y menor mortalidad [18].

#### **2.1.2 Parámetros ambientales**

La temperatura de (26-33°C) maximiza la actividad metabólica con el consumo de alimento, fuera del rango se reduce la eficiencia alimenticia y aumenta el estrés [18]. La salinidad usada es 22 a 33ppt: favorecen al equilibrio iónico evitando caídas drásticas de oxígeno disuelto, valores elevados ayudan a disminuir la concentración de O<sub>2</sub> en el agua. El pH del agua (7.5-9) rangos inferiores afectan en supervivencia y la reproducción se ven comprometidas, las concentraciones de oxígeno disuelto superiores a 5 mg/L son necesarias para prevenir el estrés

hipoxémico y reducir la incidencia de enfermedades en cambio concentraciones bajas de oxígeno disuelto provocan estrés oxidativo, disminuyen el consumo de alimento y elevan la susceptibilidad a patógenos [18].

Variaciones de estos parámetros afectan directo en la fisiología del camarón, cambios bruscos de temperatura o salinidad alteran la función enzimática y el balance osmótico afectando el crecimiento y la supervivencia de las larvas [19]. PH fuera de estos valores puede dañar las branquias y alterar la absorción de nutrientes repercutiendo negativamente en la tasa de metamorfosis y el desarrollo de la etapa postlarval [19].

### **2.1.3 Aplicación Web**

Una aplicación web es un programa de aplicación de una página o conjunto de páginas y que se aloja en un servidor remoto para ser accedido a través de Internet mediante un navegador [20]. Los servicios web son aplicaciones por definición y la mayoría de los sitios web, aunque no todos incluyen aplicaciones en línea. Los desarrolladores de diseño crean aplicaciones web para diferentes usuarios y propósitos, desde una organización hasta un individuo por numerosas razones [20].

Las aplicaciones web de uso común pueden incluir correo web, calculadoras en línea y tiendas de comercio electrónico son ejemplos de aplicaciones web populares. Aunque sólo se puede acceder a algunas aplicaciones web a través de un navegador en particular la mayoría son accesibles independientemente del navegador utilizado [20].

### **2.1.4 Angular**

Angular, uno de los framework de desarrollo JavaScript más dinámico y utilizado en la actualidad [21]. Se trata de un framework “MVC” (Modelo – Vista - Controlador) [22], desarrollado por Google LLC que da facilidad a la creación y programación de apps web de una sola página o también conocidas como webs “SPA” [21]. Se entiende por framework de una estructura de soporte conceptual y tecnológica definida con artefactos o módulos de software específicos. En otras

palabras, un tipo de estructura o modelo fundamentado en tecnología que facilita el trabajo y previene errores de programación [21].

Angular es una colección de librerías con muchas características como gestión de formularios, comunicación entre cliente y servidor, enrutamiento, entre otros. Esto permite al programador el desarrollo, pruebas, compilación y actualización del código fuente de una aplicación. Como lenguaje principal de programación de Angular se tiene Typescript [21].

### **2.1.5 Angular Material**

Angular Material es una colección de componentes de interfaz de usuario (UI) diseñada para programadores de Angular [23]. Fue desarrollado por el equipo de Angular para funcionar de manera ideal con Angular [23]. Angular Material sigue las directrices de Material Design de Google al ofrecer más de 30 componentes de interfaz de usuario para la creación de páginas web y aplicaciones web atractivas y funcionales, manteniendo los principios de diseño-web-moderno [23]. Angular Material se incorpora perfectamente a las aplicaciones de Angular. Aprovecha los beneficios del ecosistema de Angular, que incluye herramientas como la CLI de Angular, RxJS Observables y el nuevo HttpClient de Angular [23]. Igualmente, a través del CDK de Angular Material, los desarrolladores pueden crear sus propias creaciones con patrones de interacción comunes, permitiendo una mayor flexibilidad sin comprometer el rendimiento y la accesibilidad que Angular Material ofrece [23].

### **2.1.6 Base de datos**

Una base de datos es un conjunto organizado de información estructurada, o de datos generalmente almacenados electrónicamente en un sistema informático, suele estar controlada por un sistema gestor de base de datos (DBMS) [24]. Entre los datos y el gestor de BD, contando con las aplicaciones que están asociadas a ellos, se denominan un sistema de base de datos o abreviado solo como base de datos [24].

Normalmente, la información de los tipos más populares utilizados en la actualidad se representa en filas y columnas en múltiples tablas para facilitar el procesamiento y la búsqueda de datos [24]. De este modo, es posible que la información sea accesible para administrarla, cambiarla, renovarla, supervisarla y ordenarla con facilidad. Casi mayoría de las bases de datos existentes hacen uso del lenguaje de consulta estructurado (SQL) para escribir y consultar datos [24].

### **2.1.7 Lenguajes de programación**

Un lenguaje de programación es un sistema de símbolos y reglas diseñado con el fin de expresar instrucciones para ser ejecutadas por un computador [25]. Permiten a los programadores la escritura de algoritmos, creación de funciones que ayudan que sea comprensible tanto para humanos y maquinas. Cada tipo de lenguaje que existe tiene su propia sintaxis de programación lo cual la define como se estructura las instrucciones para la interpretación correcta por un compilador [25]. La existencia de diferentes tipos de lenguajes varía según el nivel de abstracción y el propósito con el cual fue diseñado como: lenguajes de bajo nivel (lenguaje de maquina) y los de alto nivel (para desarrolladores) [25].

En desarrollo de software estos lenguajes son de mucha importancia, permiten la transformación de ideas hacia aplicaciones funcionales como unos simples script hasta el uso de sistemas complejos [25]. Cada lenguaje está diseñado para tareas específicas según el creador como: JavaScript su uso es para desarrollo web, Python se creó para ciencias de datos, y otros más simples como Java o C++ usados para desarrollo de escritorio [25].

### **2.1.8 Visual Studio Code**

Visual Studio Code es un editor de código fuente gratuito y de código abierto ligero, pero con la capacidad de ejecutar en el escritorio y es compatible con varios SO como: Windows, MacOS y Linux [26]. Es ideal porque incluye soporte para lenguajes de programación entre ellos: JavaScript, TypeScript y Node.JS, adicional a eso posee un apartado con extensiones para otros tipos de lenguajes [26]. También, ofrece herramientas personalizables con interfaz de fácil manejo, para los

desarrolladores se puede acceder a tutoriales, atajos con uso del teclado y consejos que ayudan a mejorar la productividad [26].

### **2.1.9 XAMPP**

XAMPP es un entorno de desarrollo web gratuito y de código abierto que permite configurar un pequeño servidor local en el computador [27]. Su nombre proviene del inicio de cada letra de las siguientes tecnologías: Apache, MySQL, PHP y Perl además de otras herramientas [27]. Es usado principalmente por desarrolladores por su uso ideal en proyectos para pruebas locales, que se hacen antes del traslado hacia servidores de producción [27]. Como principal ventaja destacan la facilidad de uso por ser compatible con varios SO: Windows, Linux y MacOS. Además, incluye phpMyAdmin interfaz de gestión sencilla de base de datos [27].

### **2.1.10 PhpMyAdmin**

PhpMyAdmin es una herramienta de uso libre gratuita para desarrolladores en entorno local, está diseñada en “PHP” para facilitar la gestión de base de datos: MySQL y MariaDB contando con una interfaz intuitiva [28]. Los usuarios pueden realizar operaciones sencillas como: gestión de bases de datos, tablas, columnas, índices, relaciones y los permisos de usuarios esto sin necesidad de codificar o escribir líneas de consultas SQL [28]. Además, ofrece la capacidad de ejecución de consultas SQL directamente que permite una mejor administración. Es usado tanto como para principiantes y desarrolladores con mayor experiencia [28].

### **2.1.11 Apache HTTP**

Apache HTTP Server es un servidor web gratuito de código abierto, facilita la gestión y distribución de contenido web mediante el uso del protocolo HTTP [29]. Cuenta con una arquitectura que es modular, permite la integración de extensiones que personaliza el funcionamiento según las necesidades de cada tipo de proyecto [29]. La capacidad que presenta para manejar sitios web ya sean estáticos o dinámicos lo han consolidado como una opción versátil y ampliamente utilizada, formando parte esencial de configuraciones como el stack LAMP (Linux, Apache,

MySQL, PHP). En el contexto en desarrollo web apache es clave al proporcionar una plataforma para el alojamiento de apps web [29].

### **2.1.12 MySQL**

MySQL es un sistema de gestión de base de datos relacional de código abierto utilizado para: almacenar, gestionar y organizar una gran cantidad de datos, está basado en lenguaje SQL permitiendo a desarrolladores realizar: consultas, agregar, actualizar y la eliminación de datos [30]. Su arquitectura se basa en tablas organizadas en esquemas que facilita la relación entre datos y su manipulación haciéndola adecuada para proyectos personales o para sistemas grandes. Además, es conocido por ser robusto, fiable y escalable como características que le permiten el soporte de un gran volumen de conexiones simultaneas y la gestión de transacciones consistente y a la vez segura haciendo uso de la propiedad ACID [30].

### **2.1.13 NodeJS**

NodeJS es un entorno de ejecución en JavaScript asincrónico controlado por eventos permitiendo el desarrollo de aplicaciones de red escalable. Utiliza múltiples conexiones simultaneas a través de un buble de eventos evitando el bloqueo del proceso y mejorando en escalabilidad [31]. Es ideal para el uso de aplicaciones que demandan de una alta capacidad de respuesta como: servidores web en tiempo real. Además, es flexible permitiendo aprovechar varios núcleos del procesador por medio de la generación de procesos secundarios lo que facilita el balanceo de carga. Esto lo convierte en una opción potente para el desarrollo de sistemas distribuidos y aplicaciones de alto rendimiento [31].

En el desarrollo del Backend, se utilizan diversas librerías que optimizan la funcionalidad y seguridad de la aplicación. Estas son algunas de las más relevantes:

- Express: Framework ligero para la creación de APIs y manejo de rutas.
- Sequelize: ORM que facilita la interacción con bases de datos relacionales.
- bcrypt: Librería utilizada para el cifrado de contraseñas.

- cors: Permite configurar políticas de acceso para mejorar la seguridad.
- dotenv: Facilita la gestión de variables de entorno.
- MySQL/MySQL2 y mssql: Conectores para bases de datos.
- nodemon: Herramienta que permite la recarga automática del servidor durante el desarrollo.
- TypeScript y TSLint: Utilizados para tipar el código y mantener un control de calidad en el desarrollo

#### **2.1.14 API-REST**

Una Interfaz de programación de Aplicaciones Representacional (API-REST) es un tipo de interfaz web, sus principios son de estándar REST (Trasferencia de Estado Representacional) [32]. Tiene como objetivo principal facilitar la comunicación entre diferentes sistemas contando con: acceso, modificación o eliminación de datos haciendo uso de solicitudes HTTP. Los principales métodos que se utiliza son: GET, POST, PUT y DELETE los cuales son operaciones que se pueden realizar mediante una URL. Ofrece simplicidad y flexibilidad lo que la ha convertido en una de las interfaces más populares para el uso en servicios web y la comunicación en aplicaciones [32].

Su estructura estandarizada y la utilización de tecnologías como: HTTP y JSON les permite que su integración entre sistemas sea rápida y escalable, independientemente del tipo de lenguaje de programación o las plataformas usadas. Las API REST son empleadas en numerosos contextos, desde servicios web como PayPal y OpenWeatherMap hasta dispositivos IoT y aplicaciones empresariales [32].

#### **2.1.15 EmailJS**

EmailJS es un servicio que ofrece el envío de correos electrónicos directo usando client-side technologies, eliminando el uso de un servidor Backend que gestione el

envío de correos [33]. Su funcionamiento hace uso de la conexión con proveedores de email, cuenta con plantillas predefinidas y soporta: Gmail, Outlook, entre otros. Las plantillas pueden contener variables dinámicas que son configuradas desde la aplicación para su uso en el código, lo que permite la personalización del contenido sea: asunto o destinatarios, adaptándose a las necesidades del desarrollador [33].

Además, tiene otras funcionalidades avanzadas como: integración de archivos adjuntos, validación por método CAPTCHA, revisión del historial de solicitudes de envío de correos. También Esta herramienta es ideal para desarrolladores que buscan una solución rápida y eficiente para enviar correos electrónicos desde el frontend sin la complejidad de implementar un servidor [33].

### **2.1.16 Bootstrap**

Bootstrap es un framework gratuito y openSource, diseñado para el desarrollo en frontend de sitios y aplicaciones web [34]. Esta creado con un enfoque mobile-first permitiendo a los desarrolladores la creación de sitios web responsivos que son adaptados automáticamente al tamaño y orientación de pantalla del usuario. Incluye una gran colección de componentes: interfaz de usuario, sistema de cuadrícula flexible de 12 columnas y herramientas JavaScript precompiladas, ayudando a la fácil creación de las interfaces web son tener que escribir desde cero comandos básicos como: HTML, CSS o JavaScript [34].

Lo más destacado es la capacidad de simplificar y acelerar el desarrollo web por el uso de plantillas y estilos prediseñados que ofrece, lo que garantiza la consistencia y compatibilidad entre navegadores y dispositivos. Fue creado por desarrolladores de Twitter y es ampliamente utilizado por su enfoque modular y personalizable, que permite a los desarrolladores adaptar y extender el framework según las necesidades de cada proyecto. Además, es ideal para quienes buscan optimizar el desarrollo de sitios web y aplicaciones con una estructura clara y eficiente [34].

### **2.1.17 Postman**

Postman es una plataforma que permite la construcción y uso de APIs que simplifica cada etapa del ciclo de una API, partiendo desde: el diseño, pruebas y documentación [35]. Los desarrolladores hacen uso para: almacenar, gestionar y colaborar en torno a las especificaciones de las APIs como: casos de pruebas, resultados o documentación, todo en una sola plataforma centralizada. Ofrece espacios de trabajos personalizados tanto para: individuos, equipos y socios, permitiendo una mejor organización y compartir el trabajo de manera eficiente. También utiliza el protocolo HTTP, los métodos que se utilizan son: GET, POST, PUT y DELETE los cuales son operaciones que se pueden realizar mediante una URL [35].

## **2.2 Marco Teórico**

### **2.2.1 Impacto de la Nutrición en el Cultivo de Larvas de Camarón**

La acuicultura y su constante evolución a lo largo del tiempo representa en una de las principales fuentes de producción de proteína a nivel mundial [36]. La cría de camarón actividad de alto impacto económico-productivo, la fase larval considerada una de las más críticas para garantizar la eficiencia del cultivo de larvas de camarón [15]. Durante esta etapa inicial estos organismos son altamente susceptibles a variaciones ambientales y deficiencias nutricionales, lo que hace indispensable una gestión adecuada de su alimentación y monitoreo [16].

La producción de larvas de camarón basa su éxito en el tipo de manejo nutricional preciso el mismo que considere las necesidades metabólicas específicas de los organismos en cada fase de desarrollo [15]. Factores ambientales (temperatura, salinidad, oxígeno disuelto y pH del agua), parámetros determinantes en la absorción de nutrientes y la supervivencia de las larvas, las condiciones ambientales subóptimas pueden causar estrés y afectar el crecimiento, incrementando la tasa de mortalidad y disminuyendo la rentabilidad del cultivo [37].

En la fase larval: la dieta combina alimento vivo con piensos formulados, el uso de artemia destaca por su aporte de vitaminas, proteínas y ácidos grasos omega-3, esenciales para el crecimiento [15]. Aun así, detectar a tiempo deficiencias nutricionales sigue siendo un reto: los efectos de una mala alimentación suelen notarse solo en etapas posteriores. Estudios señalan que la calidad y tamaño del pellet influyen de forma directa en la absorción de nutrientes y, por ende, en la tasa de crecimiento y la mortalidad de *Litopenaeus vannamei* [16].

Esto ha permitido que los sistemas de monitoreo cobren importancia en la industria, registrar temperatura, salinidad u oxígeno disuelto en tiempo real, es posible ajustar la dieta según las condiciones reales del agua y el estado fisiológico de las larvas [10]. Los métodos tradicionales —observar coloración, crecimiento o comportamiento de alimentación— resultan útiles, pero son subjetivos y propensos a errores.

El uso de herramientas tecnológicas como: (sensores - plataformas web y/o análisis de datos) ofrece un seguimiento más preciso de la calidad del agua y la nutrición [38]. Digitalizar estos procesos mejora la toma de decisiones, optimiza el uso de insumos y disminuye el impacto ambiental asociado al desperdicio de alimento y al deterioro de la calidad del agua [12].

### **2.2.2 Importancia de los Sistemas de Gestión y Monitoreo en Acuicultura**

El desarrollo de la acuicultura ha estado acompañado de la implementación de sistemas de gestión que buscan optimizar la producción y mejorar la sostenibilidad del sector [10]. Tradicionalmente, el monitoreo en cultivos de larvas de camarón se ha realizado de manera manual, con registros en hojas de cálculo o en bitácoras físicas que requieren la intervención constante del personal. Sin embargo, este enfoque presenta limitaciones, como la dificultad para acceder a la información en tiempo real y la posibilidad de errores en la recolección de datos.

Actualmente, existen sistemas de monitoreo inteligente que integran sensores, análisis de datos y herramientas de monitoreo remoto, permitiendo rastrear y gestionar diversos parámetros dentro de las granjas acuícolas [9]. Entre las

soluciones más utilizadas se encuentran los sensores inteligentes, que permiten medir parámetros como temperatura, oxígeno disuelto y salinidad con alta precisión, así como los sistemas de alimentación automatizada, que optimizan la administración de nutrientes y reducen el desperdicio [10]. Además, el uso de plataformas basadas en la nube facilita la integración y análisis de grandes volúmenes de datos, permitiendo la toma de decisiones en tiempo real y la implementación de estrategias de alimentación personalizadas.

La digitalización en la acuicultura no solo contribuye a la eficiencia operativa, sino que también permite mejorar la trazabilidad y sostenibilidad del cultivo [36]. Al automatizar la recopilación y análisis de datos, los productores pueden identificar tendencias, anticipar problemas y ajustar sus estrategias en función de datos concretos, reduciendo así el margen de error y optimizando los recursos disponibles. Esto representa una ventaja significativa frente a los métodos tradicionales, donde la toma de decisiones se basa en la experiencia empírica y no en evidencia cuantificable.

### **2.2.3 Angular como framework en el desarrollo de aplicaciones web para gestión de datos**

Angular, desarrollado por Google, es presentado como un framework de los más populares para la creación de aplicaciones web dinámicas de una sola página. Proporciona una arquitectura basada en componentes simplificando el desarrollo y mantenimiento de apps web. Una de las ventajas clave de “angular”, es su capacidad para admitir el enlace de datos bidireccional permitiendo la sincronización en tiempo real entre el modelo y vista [39]. Haciéndolo ideal para el desarrollo de aplicaciones interactivas, donde los usuarios pueden ver actualizaciones al instante como: paneles, sistemas de chat o de soporte técnico en vivo [40].

Además, la inyección de dependencias que incorpora y el diseño modular que ofrece, lo convierte en una herramienta poderosa para la creación de aplicaciones escalables garantizando que los proyectos estén organizados y administrables a medida que crecen [40]. Angular también integra a la perfección con otras

herramientas y frameworks haciéndola flexible para las diversas necesidades de desarrollo [39]. Se puede utilizar para desarrollar la interfaz frontend, conectándose con un sistema Backend impulsado por una API.

En el contexto del presente proyecto, Angular se utilizará para desarrollar una interfaz intuitiva y dinámica que permitirá al personal del laboratorio gestionar corridas de cultivo, monitorear parámetros críticos, y planificar la alimentación de manera eficiente. La capacidad de optimización del rendimiento de Angular, mediante técnicas como la compilación anticipada (Ahead-of-Time - AOT) y tree shaking, garantizará tiempos de carga rápidos, brindando una experiencia fluida y eficiente a los usuarios del sistema [40]

## **2.3 Metodología del Proyecto**

### **2.3.1 Metodología de la Investigación**

En este proyecto se emplea una metodología cualitativa basada en la observación no participante en un laboratorio de cultivo de larvas de camarón. Durante jornadas programadas, el investigador registró sistemáticamente los parámetros ambientales (temperatura, pH, oxígeno disuelto, salinidad) y las prácticas de alimentación cada cuatro horas, utilizando una guía de observación diseñada ad hoc. Las notas de campo incluyeron descripciones detalladas de los procedimientos manuales de registro, la secuencia de ciclos de alimentación y las condiciones de los tanques de cultivo. Esta técnica permitió obtener información rica en contexto sobre las rutinas diarias y las posibles incidencias en el manejo de la alimentación larvaria.

Para asegurar la consistencia en la recolección de datos, se utilizó un instrumento estandarizado —guía de observación estructurada— con categorías explícitas para anotar eventos relacionados con cada parámetro medido y cada acción de alimentación. Cada sesión de observación fue documentada en formato de tabla de registro, indicando hora, valor del parámetro y descripción de la práctica alimenticia ejecutada. Posteriormente, las entradas de la guía serán sometidas a un análisis de contenido mediante codificación abierta, con el fin de identificar patrones y áreas de mejora en el proceso observado.

### **2.3.2 Técnicas e instrumentos de recolección de datos**

A continuación, se detalla de manera puntual las técnicas e instrumentos para la recolección de información que se llevara a cabo en el presente proyecto.

#### **Técnicas cualitativas:**

Se llevó a cabo una visita a un laboratorio local de producción de larvas de camarón, donde se observaron las prácticas habituales de registro de parámetros ambientales y la planificación alimenticia. Durante esta observación, se tomaron notas de campo que permitieron identificar desafíos como la dispersión de la información, la falta de correlación entre los datos ambientales y la alimentación, y la dependencia de métodos manuales. Los hallazgos recabados se detallan en el **Anexo 1**.

#### **Instrumentos:**

Fuentes Bibliográficas y Documentales: Se utilizaron libros, artículos científicos, tesis, informes técnicos y publicaciones digitales como principales fuentes de información. Estas permitieron obtener datos precisos y actualizados.

### **2.3.3 Metodología de desarrollo**

La metodología de desarrollo de la aplicación se basará en la modelo incremental permitiendo que se pueda construir y entregar el sistema en módulos funcionales siendo iterativa o etapas. Los módulos que contara serán desarrollado y probado de manera independiente antes de la integración en el sistema completo con esta metodología podemos realizar las fases de: análisis, diseño, codificación y pruebas [41].

Esta metodología se basa en las siguientes fases:

- **Fase de análisis:** en esta fase la información recopilada para el desarrollo de los módulos de la aplicación web ayudaran a identificar los requerimientos funcionales y no funcionales.

- **Fase de diseño:** esta etapa se diseña la arquitectura y la interfaz de la aplicación web que lograra presentarse en cada tipo de usuario de manera amigable y de fácil entendimiento.
- **Fase de codificación:** en esta fase procede al desarrollo de los módulos y sus respectivas interfaces y resalta la configuración de los servicios de conexión de base de datos para poder ejecutarlo y verificar su funcionalidad.
- **Fase de prueba:** esta etapa al finalizar cada uno de los módulos se verifica el correcto funcionamiento realizando pruebas de rendimiento del sistema que componen la aplicación web.

### **Incrementos del desarrollo**

El sistema se desarrollará en cinco incrementos, cada uno correspondiente a un módulo funcional del sistema:

#### **1. Primer incremento: Gestión de Usuarios**

- Implementación del módulo de autenticación y control de acceso.
- Registro, modificación y asignación de roles a usuarios.
- Pruebas de acceso y validación de permisos.

#### **2. Segundo incremento: Módulos y Tanques**

- Registro de Módulos y tanques del laboratorio.
- Seguimiento de Uso de los respectivos módulos con sus tanques asociados mediante estados.

#### **3. Tercer incremento: Gestión de Corridas (Siembras-Cosechas)**

- Registro y seguimiento de corridas de producción.
- Clasificación del rendimiento (mala, regular, buena).
- Historial de producción y pruebas de funcionalidad.

#### 4. Cuarto incremento: Monitoreo

- Registro diario de parámetros ambientales (temperatura, salinidad, oxígeno disuelto, pH).
- Generación automática de observaciones.
- Propuesta de acciones recomendadas cuando algún valor supere los rangos críticos.

#### 5. Quinto incremento: Planes de Alimenticios

- Cálculo de dosis y frecuencia de alimento según subestadio larval.
- Inclusión de suplementos y probióticos.
- Exportación del plan en PDF para el operario.

#### 6. Sexto incremento: Reportes y Análisis

- Implementación del módulo de generación de reportes técnicos.
- Análisis de alimentación, rendimiento y cosechas.
- Pruebas finales de integración y optimización del sistema.

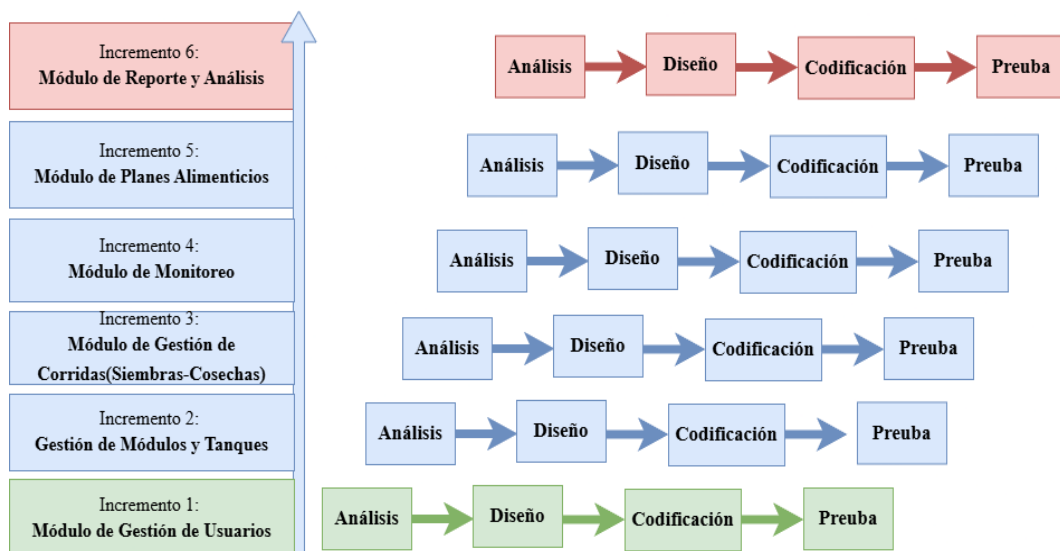


Figura 2: Metodología de Desarrollo

## CAPÍTULO 3. PROPUESTA

### 3.1 Requerimientos

#### 3.1.1 Requerimientos Funcionales

TIPO	CÓDIGO	DESCRIPCIÓN
Perfil y Roles	RF-01	Definir tres roles en el sistema: <ul style="list-style-type: none"><li>• Administrador: acceso total a todas las secciones.</li><li>• Biólogo: acceso a monitoreo e informes técnicos.</li><li>• Gerente: acceso a reportes de gestión y estadísticas.</li></ul>
Autenticación	RF-02	El usuario ingresa correo y contraseña. Si las credenciales son válidas y la cuenta está activa, recibe un token y su menú personalizado.
	RF-03	En la pantalla “¿Olvidaste tu contraseña?”, el usuario introduce su correo y recibe un enlace con token para restablecerla. Al abrir el enlace, el usuario ve un formulario para ingresar y confirmar la nueva contraseña.
Gestión de Usuarios	RF-04	Crear un nuevo usuario con nombre, correo, contraseña y rol; generar token de verificación. Comprobar si un correo ya está registrado en el sistema.
	RF-05	Recuperar el token de verificación asociado a un correo. Validar el token de verificación y marcar la cuenta como confirmada.
	RF-06	Obtener usuarios filtrando por nombre, correo o rol, sin exponer la contraseña. Recuperar datos completos de un usuario específico, incluyendo su rol.

	<b>RF-07</b>	Modificar nombre, correo, rol o estado de un usuario existente. Borrar un usuario del sistema por su identificador. Cambiar el estado (activo/inactivo) de un usuario.
<b>Módulos</b>	<b>RF-08</b>	Listar todos los módulos disponibles (estado “disponible”), con filtro por nombre. Listar todos los módulos, con sus tanques asociados, sin filtro de estado.
	<b>RF-09</b>	Crear un nuevo módulo proporcionando nombre y descripción. Actualizar nombre y descripción de un módulo existente. Eliminar un módulo sólo si no tiene tanques asociados.
<b>Tanques</b>	<b>RF-10</b>	Crear un nuevo tanque en un módulo existente. Actualizar nombre, capacidad, estado o descripción de un tanque. Eliminar un tanque por su identificador. Listar tanques “disponible” o “en uso”, opcionalmente filtrados por módulo.
<b>Corridas</b>	<b>RF-11</b>	Listar todas las corridas, con filtros opcionales por nombre, módulo y estado. Listar sólo las corridas activas, con filtros opcionales por nombre y módulo.
	<b>RF-12</b>	Crear una nueva corrida; al guardarla, marcar el módulo asociado como “en uso” dentro de la misma transacción. Actualizar los datos de una corrida existente (nombre, fechas, módulo, estado). Eliminar una corrida por su identificador.
<b>Siembras</b>	<b>RF-13</b>	Registrar una nueva siembra: validar capacidad del tanque, ajustar zona horaria, fijar cantidad actual. Actualizar una siembra existente, incluyendo fecha (con hora local) y demás campos. Eliminar una siembra: dentro de la misma transacción, liberar el tanque a “disponible”.

	<b>RF-14</b>	Listar siembras activas de una corrida (estado “activa” y cantidad > 0), con sus tanques y estadios. Listar siembras finalizadas de una corrida, con sus tanques y estadios.
<b>Cosechas</b>	<b>RF-15</b>	Obtener todos los estadios de cultivo, ordenados o sin orden específico. Listar todas las cosechas de una corrida, incluyendo datos de siembra y tanque relacionados.
	<b>RF-16</b>	Crear una nueva cosecha: validar cantidad contra la siembra, actualizar estado de siembra y módulo/tanque. Actualizar los campos de una cosecha existente, sin alterar la corrida ni la siembra original.
<b>Monitoreo</b>	<b>RF-17</b>	Listar todos los registros de monitoreo, con filtros opcionales por corrida, tanque y rango de fechas. Obtener los datos completos de un monitoreo por su identificador. Calcular y devolver el subestadio vigente para cada siembra activa en una corrida.
	<b>RF-18</b>	Crear un nuevo registro de monitoreo: validar siembra, calcular mortalidad y observaciones/acciones. Modificar un monitoreo existente: recalcular observaciones, acciones y ajustar siembra. Borrar un registro de monitoreo por su identificador.
	<b>RF-19</b>	Generar un PDF con logo, tabla de monitoreo y subtítulo (“Corrida X — Fecha Y”).

<b>Planes Alimenticios</b>	<b>RF-20</b>	Obtener las configuraciones de dosis (dosis_base, base_larvas) para un subestadio dado, incluyendo datos de producto. Para una corrida, listar los tanques con su subestadio actual y calcular la dosis recomendada por producto. Guardar un plan maestro con su fecha y todos sus detalles (siembra, subestadio, producto, hora y cantidad).
	<b>RF-21</b>	Generar un PDF organizado por tanques y productos, con logos y tabla de dosis, para un plan existente.
	<b>RF-22</b>	Marcar un plan como “cancelado” (soft-delete) sin eliminar registros históricos. Listar todos los planes (filtrables por corrida), indicando subestadio, fecha, estado y si es el último plan.
<b>Reportes</b>	<b>RF-23</b>	Listar en PDF los datos de cada detalle de plan: fecha, hora, dosis aplicada vs recomendada, desviación y observaciones.
	<b>RF-24</b>	En PDF, mostrar solo registros con mortalidad o parámetros fuera de rango, incluyendo “desviaciones” concatenadas.
	<b>RF-25</b>	Incluir en un único PDF línea de mortalidad (%) vs fecha y tabla detallada de mortalidad absoluta, porcentaje y observaciones.
	<b>RF-26</b>	Sanitizar y rechazar cadenas vacías o "null" para filtros opcionales, antes de ejecutar cualquier consulta.

*Tabla 1: Requerimientos Funcionales*

### 3.1.2 Requerimientos No Funcionales

CÓDIGO	DESCRIPCIÓN
<b>RNF-01</b>	Las contraseñas y tokens se gestionan de forma segura (hash y expiración cada 1 h).
<b>RNF-02</b>	Los endpoints de autenticación, CRUD y filtros responden bajo carga normal.
<b>RNF-03</b>	Listados grandes usan paginación o límite.
<b>RNF-04</b>	Todas las modificaciones (usuarios, corridas, monitoreos, planes) usan transacciones y rollback ante cualquier error.
<b>RNF-05</b>	Las fechas se manejan con zona horaria local (UTC-5) y validan formato ISO 8601.
<b>RNF-06</b>	Errores y validaciones devuelven mensajes claros.
<b>RNF-07</b>	La API documenta parámetros de filtro (nombre, módulo, estado, fechas) y valores permitidos.
<b>RNF-08</b>	La UI es responsive y muestra mensajes de error legibles.
<b>RNF-09</b>	Todos los PDFs incluyen logos, títulos y subtítulos dinámicos, respetan márgenes y no desbordan en A4.
<b>RNF-10</b>	Las gráficas se generan en memoria (buffers) sin archivos temporales y se incrustan online para iframes.

*Tabla 2: Requerimientos No Funcionales*

## 3.2 Componentes de la Propuesta

### 3.2.1 Arquitectura del Sistema

El sistema utilizado para la arquitectura sigue el patrón Modelo-Vista-Controlador (MVC) para el diseño que divide la aplicación en tres componentes conectados entre sí, pero a la vez independientes [22]. Se desarrolló para estandarizar el diseño de las aplicaciones y la reducción del esfuerzo de programación, esta separación

permite que cada componente se implemente por separado dando facilidad ha actualizaciones futuras y mejorar la eficiencia en el desarrollo. El **modelo** se encarga de gestionar los datos y la lógica de negocio, la **vista** por otro lado es la encargada de la presentación de información al usuario, y por último el **controlador** es el que actúa como intermediario dando el manejo de interacciones del usuario y dando actualización tanto al modelo como la vista según se requiera [22].

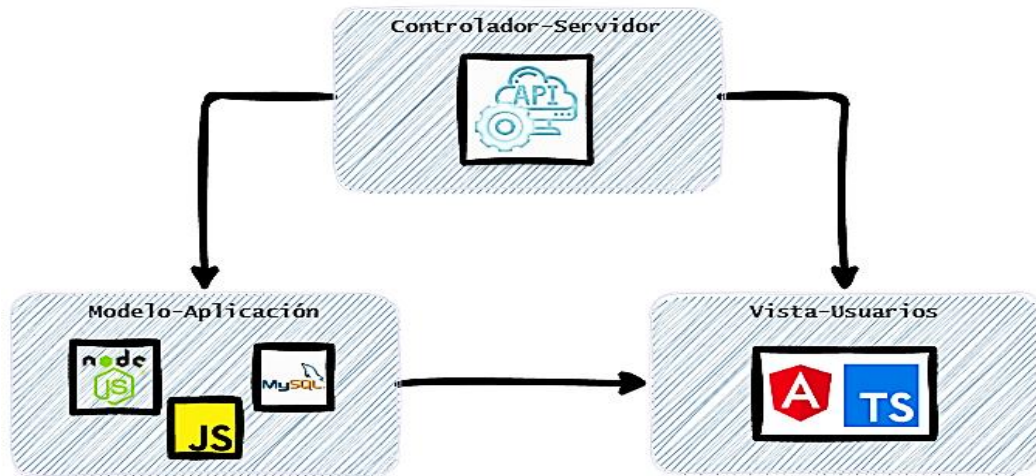


Figura 3: Arquitectura del Sistema

### 3.2.2 Diagramas de caso de Uso

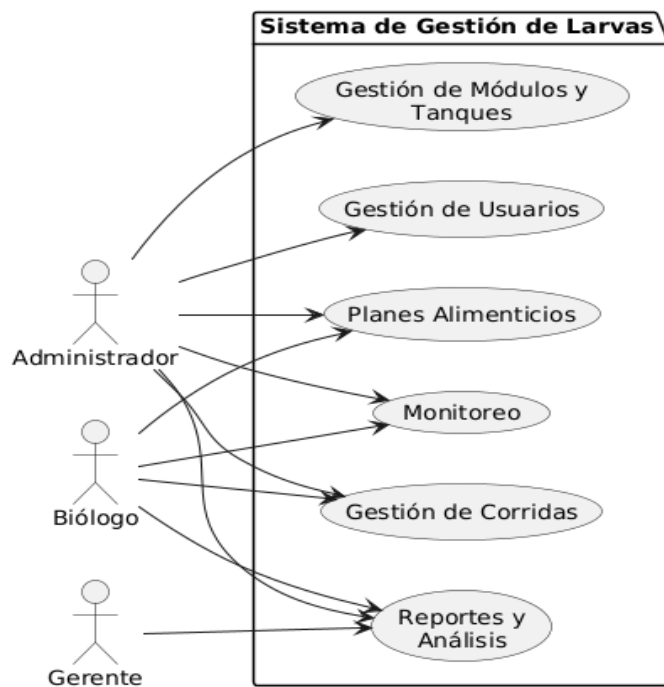


Figura 4: Caso de Uso. Sistema

## Caso de Uso: Autenticación & Acceso

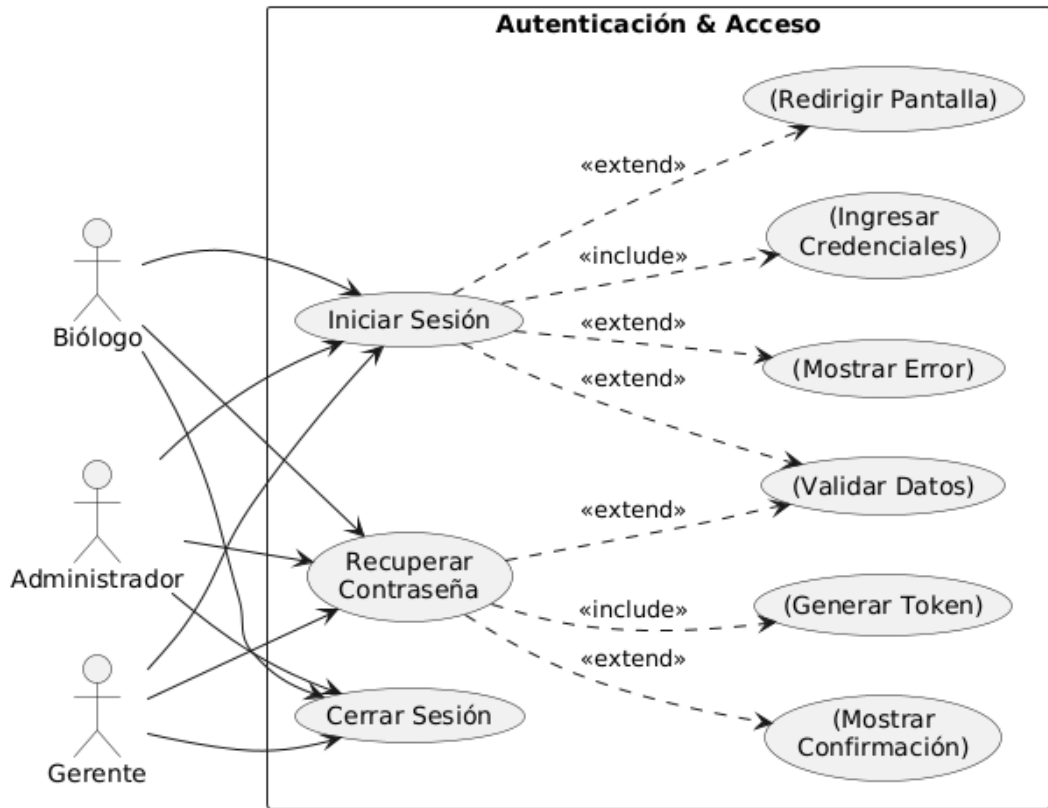


Figura 5: Caso de Uso. Autenticación Acceso

### Caso de Uso Autenticación & Acceso

<b>Actor(es)</b>	Administrador, Biólogo, Gerente
<b>Objetivo</b>	Permitir al usuario iniciar sesión, recuperar contraseña y cerrar sesión de forma segura.
<b>Descripción</b>	Conjunto de funciones que validan credenciales, gestionan tokens y garantizan la integridad de la sesión.
<b>Flujo Principal</b>	<ul style="list-style-type: none"> <li>• El actor abre la pantalla de inicio.</li> <li>• Ingresa correo y contraseña.</li> <li>• El sistema valida datos y genera token.</li> <li>• Muestra el menú según rol.</li> <li>• El actor puede cerrar sesión cuando lo requiera.</li> </ul>

<b>Subflujos</b>	<ul style="list-style-type: none"> <li>• Recuperar contraseña: el actor solicita enlace, recibe email, define nueva clave.</li> <li>• Renovar token: sesión prolongada renueva token antes de expirar.</li> <li>• Salir: invalida token y redirige a login.</li> </ul>
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>• Credenciales inválidas ⇒ mensaje de error.</li> <li>• Cuenta inactiva ⇒ bloqueo y aviso.</li> <li>• Token expirado ⇒ forzar re-login.</li> </ul>

Tabla 3: Caso de Uso. Autenticación & Acceso

### Caso de Uso: Gestión de Usuarios

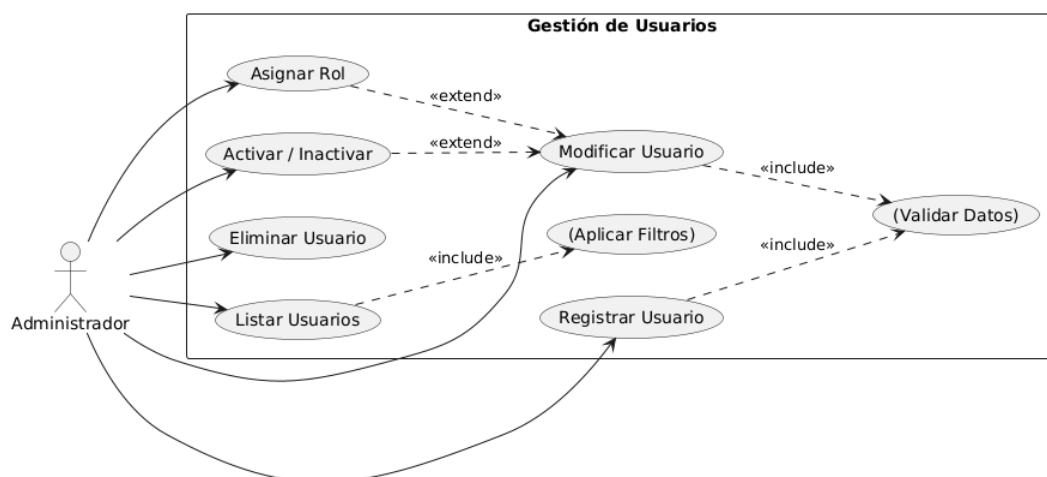


Figura 6: Caso de Uso. Gestión de Usuarios

### Caso de Uso Gestión de Usuarios

<b>Actor(es)</b>	Administrador
<b>Objetivo</b>	Administrar el ciclo de vida de las cuentas (crear, editar, asignar rol, activar/inactivar, eliminar).
<b>Descripción</b>	Permite al administrador mantener la base de usuarios y sus permisos, garantizando unicidad de correo y trazabilidad de cambios.

<b>Flujo Principal</b>	<ul style="list-style-type: none"> <li>• Abre “Usuarios”.</li> <li>• Aplica filtros opcionales (nombre, rol, estado).</li> <li>• Selecciona “Nuevo” o elige un usuario existente.</li> <li>• Completa/edita formulario y guarda.</li> <li>• El sistema envía correo de confirmación o actualiza registro.</li> </ul>
<b>Subflujos</b>	<ul style="list-style-type: none"> <li>• Asignar/Cambiar rol (extiende “Modificar”).</li> <li>• Activar/Inactivar (soft-delete).</li> <li>• Validar datos (incluye verificación de correo único y formato).</li> </ul>
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>• Email duplicado ⇒ bloquea creación.</li> <li>• Usuario con dependencias críticas (p. ej. registros de monitoreo) ⇒ impide eliminación física.</li> <li>• Rol inexistente ⇒ error de configuración.</li> </ul>

Tabla 4: Caso de Uso. Gestión de Usuarios

### Caso de Uso: Módulos & Tanques

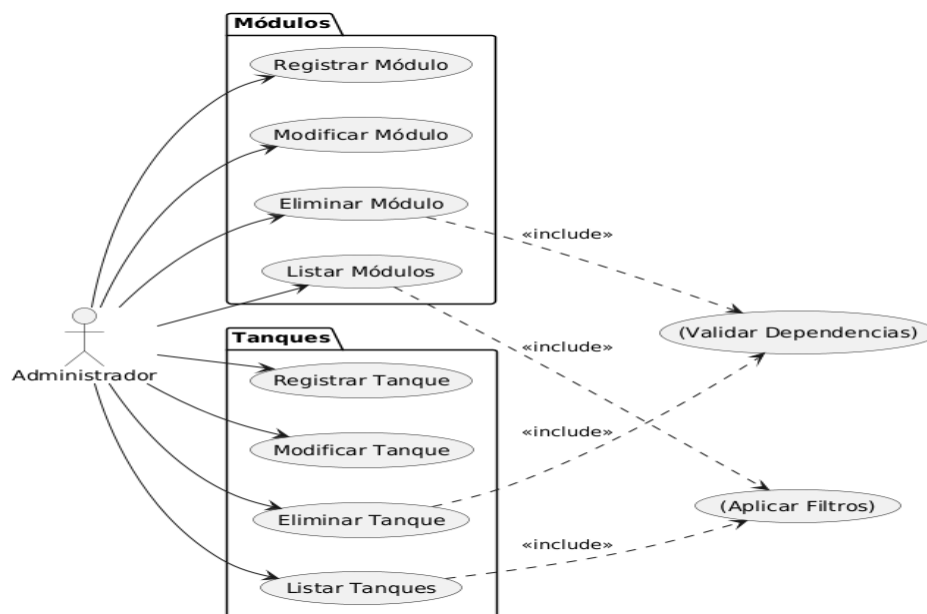


Figura 7: Caso de Uso. Módulos & Tanques

## Caso de Uso Módulos y Tanques

<b>Actor(es)</b>	Administrador
<b>Objetivo</b>	Crear, editar, listar y eliminar módulos de cultivo y sus tanques asociados.
<b>Descripción</b>	Mantiene la infraestructura física del laboratorio, controlando disponibilidad y capacidad de cada tanque.
<b>Flujo Principal</b>	<ul style="list-style-type: none"> <li>• Accede a “Módulos”.</li> <li>• Registra módulo (nombre, descripción).</li> <li>• Dentro del módulo, añade tanques con capacidad y estado.</li> <li>• Modifica o elimina registros según políticas.</li> </ul>
<b>Subflujos</b>	<ul style="list-style-type: none"> <li>• Validar dependencias: evita borrar módulo con tanques.</li> <li>• Listar tanques: filtra por estado (disponible/en uso).</li> </ul>
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>• Nombre de módulo duplicado.</li> <li>• Tanque en uso (corrida activa) ⇒ bloqueo de eliminación.</li> <li>• Capacidad inválida (<math>\leq 0</math>).</li> </ul>

Tabla 5: Caso de Uso. Módulos & Tanques

## Caso de Uso: Corridas

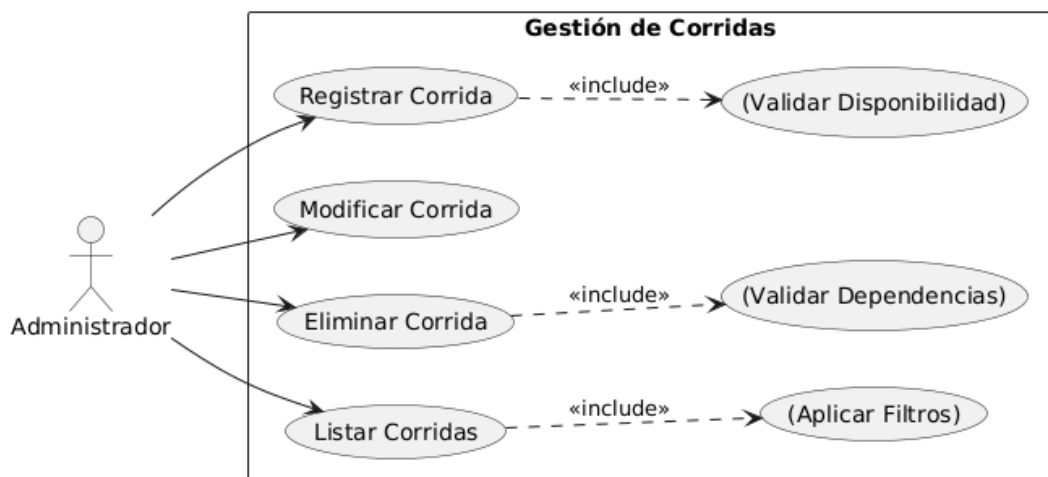


Figura 8: Caso de Uso. Corridas

## Caso de Uso Corridas

<b>Actor(es)</b>	Administrador
<b>Objetivo</b>	Iniciar, actualizar, listar y cerrar corridas de producción.
<b>Descripción</b>	Controla el ciclo global de cultivo asignando un módulo y fechas clave, reservando recursos durante toda la corrida.
<b>Flujo Principal</b>	<ul style="list-style-type: none"> <li>• Selecciona “Nueva Corrida”.</li> <li>• Ingresa nombre, módulo, fechas estimadas.</li> <li>• El sistema verifica que el módulo esté disponible y lo marca “En uso”.</li> <li>• Durante la corrida, puede cambiar estado o fechas.</li> <li>• Al finalizar, cierra y libera módulo.</li> </ul>
<b>Subflujos</b>	<ul style="list-style-type: none"> <li>• Validar disponibilidad (incluye).</li> <li>• Reabrir corrida (extiende) en caso de ajustes operativos.</li> </ul>
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>• Módulo ocupado ⇒ creación rechazada.</li> <li>• Intento de eliminar corrida con siembras activas.</li> </ul>

Tabla 6: Caso de Uso. Corridas

## Caso de Uso: Siembras

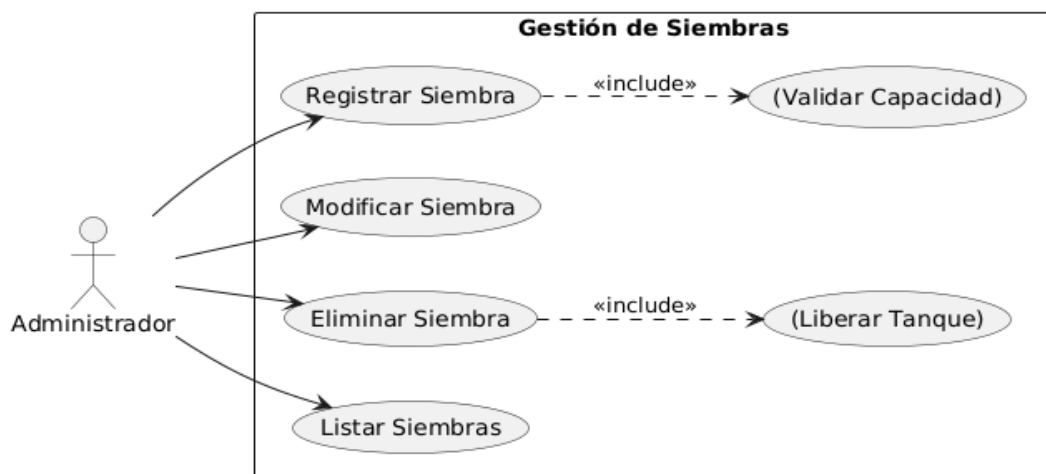


Figura 9: Caso de Uso. Gestión de Siembras

### Caso de Uso Siembras

<b>Actor(es)</b>	Administrador
<b>Objetivo</b>	Registrar, modificar y eliminar siembras dentro de una corrida.
<b>Descripción</b>	Registra la introducción de larvas en tanques, validando la capacidad y fechando el evento.
<b>Flujo Principal</b>	<ul style="list-style-type: none"><li>• Elige corrida y tanque.</li><li>• Ingresa cantidad, fecha y estadio.</li><li>• El sistema valida capacidad y guarda.</li><li>• Puede editar datos o finalizar siembra.</li></ul>
<b>Subflujos</b>	<ul style="list-style-type: none"><li>• Validar capacidad (incluye).</li><li>• Liberar tanque al eliminar siembra.</li></ul>
<b>Excepciones</b>	<ul style="list-style-type: none"><li>• Capacidad excedida.</li><li>• Eliminación rechazada si hay cosecha registrada.</li></ul>

Tabla 7: Caso de Uso. Gestión de Siembras

### Caso de Uso: Cosechas

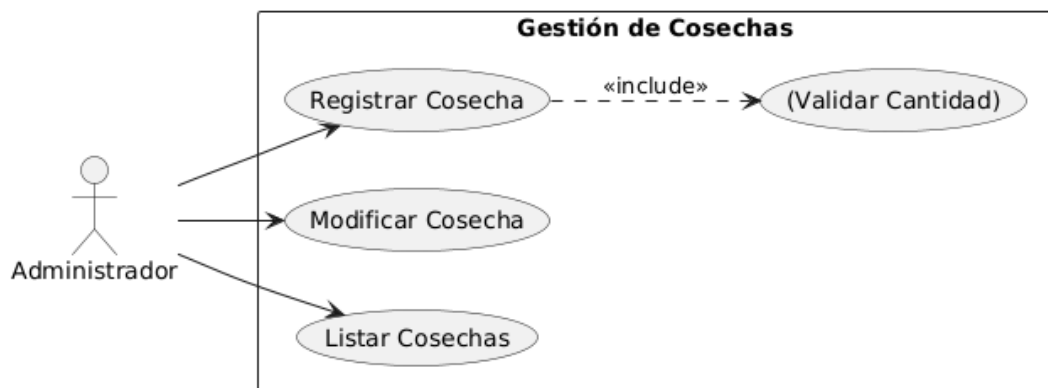


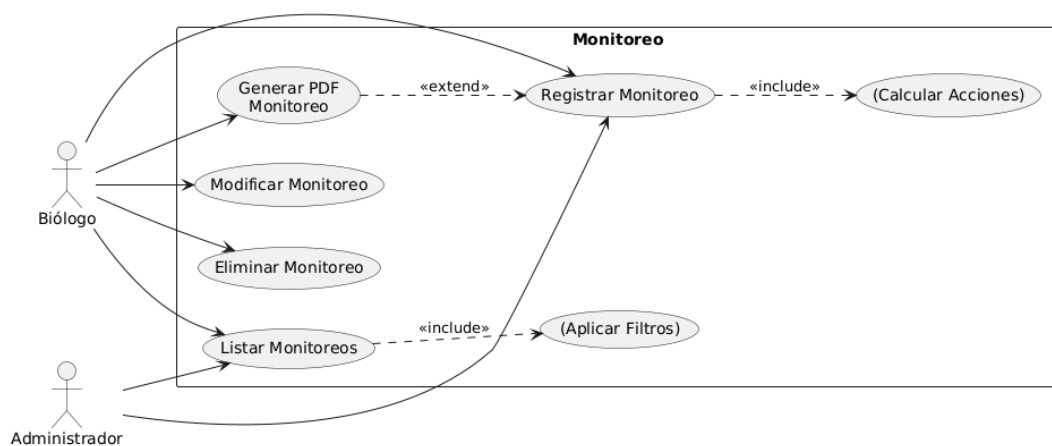
Figura 10: Caso de Uso. Gestión de Cosechas

**Caso de Uso Cosechas**

<b>Actor(es)</b>	Administrador
<b>Objetivo</b>	Registrar y analizar las cosechas resultantes de cada siembra.
<b>Descripción</b>	Controla la salida de producto final, actualiza estados y genera historial productivo.
<b>Flujo Principal</b>	<ul style="list-style-type: none"> <li>• Selecciona siembra finalizada.</li> <li>• Ingresa cantidad cosechada y fecha.</li> <li>• Sistema valida que no supere la cantidad sembrada.</li> <li>• Actualiza indicadores de rendimiento.</li> </ul>
<b>Subflujos</b>	<ul style="list-style-type: none"> <li>• Validar cantidad (incluye).</li> <li>• Reimpresión de acta (extiende).</li> </ul>
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>• Cantidad &gt; sembrada ⇒ error de entrada.</li> <li>• Modificación bloqueada después de cierre contable.</li> </ul>

*Tabla 8: Caso de Uso. Cosechas*

**Caso de Uso: Monitoreo**



*Figura 11: Caso de Uso. Monitoreo*

## Caso de Uso **Monitoreo**

<b>Actor(es)</b>	Biólogo, Administrador
<b>Objetivo</b>	Registrar parámetros diarios, generar observaciones y acciones.
<b>Descripción</b>	Permite un seguimiento técnico de temperatura, salinidad, pH, OD, etc.; calcula mortalidad y recomienda ajustes.
<b>Flujo Principal</b>	<ul style="list-style-type: none"> <li>• Selecciona corrida/tanque.</li> <li>• Ingresar valores medidos.</li> <li>• Sistema calcula subestadio y mortalidad.</li> <li>• Genera observaciones y acciones sugeridas.</li> <li>• Guarda registro.</li> </ul>
<b>Subflujos</b>	<ul style="list-style-type: none"> <li>• Calcular acciones (incluye).</li> <li>• Generar PDF (extiende) para reporte diario.</li> </ul>
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>• Parámetros fuera de rango sin acción definida ⇒ alerta crítica.</li> <li>• Registro duplicado para la misma fecha.</li> </ul>

Tabla 9: Caso de Uso. Monitoreo

## Caso de Uso: Planes Alimenticios

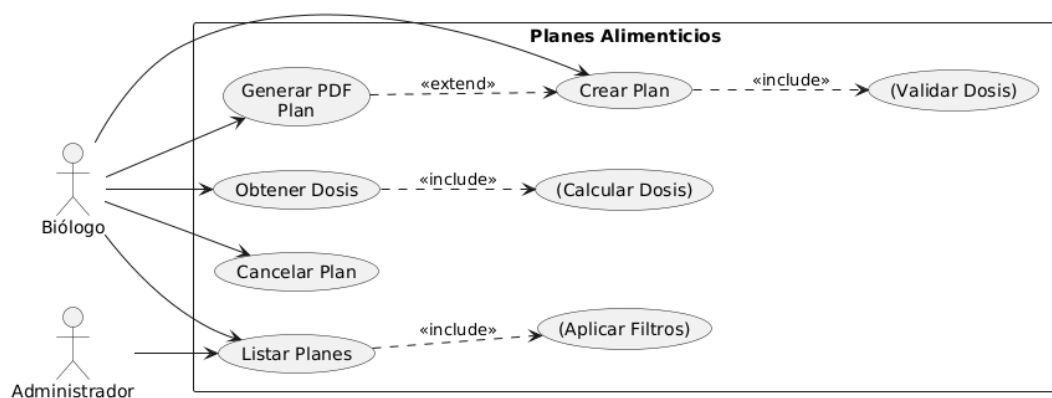


Figura 12: Caso de Uso. Planes Alimenticios

## Caso de Uso Planes Alimenticios

<b>Actor(es)</b>	Biólogo, Administrador
<b>Objetivo</b>	Calcular dosis y programar dieta diaria por subestadio.
<b>Descripción</b>	Basado en datos de monitoreo, calcula la cantidad de alimento, suplementos y probióticos; genera plan detallado para el operario.
<b>Flujo Principal</b>	<ul style="list-style-type: none"> <li>• Selecciona corrida y fecha.</li> <li>• Sistema lista tanques y subestadios actuales.</li> <li>• Calcula dosis recomendada por producto.</li> <li>• El biólogo ajusta si es necesario y guarda plan.</li> <li>• El plan queda disponible para impresión.</li> </ul>
<b>Subflujos</b>	<ul style="list-style-type: none"> <li>• Validar dosis (incluye).</li> <li>• Exportar PDF (extiende) con logo y tabla.</li> </ul>
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>• Falta de configuraciones de dosis base.</li> <li>• Plan superpuesto (ya existe plan para esa fecha).</li> </ul>

Tabla 10: Caso de Uso. Planes Alimenticios

## Caso de Uso: Reportes & Análisis

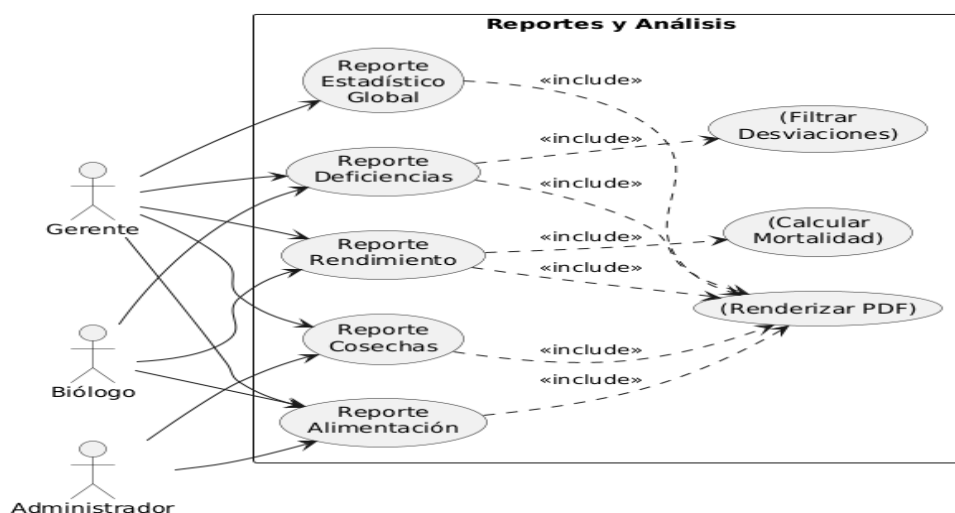


Figura 13: Caso de Uso. Reportes & Análisis

<b>Caso de Uso Reportes y Análisis</b>	
<b>Actor(es)</b>	Gerente, Biólogo, Administrador
<b>Objetivo</b>	Generar informes técnicos y gerenciales sobre producción, alimentación y deficiencias.
<b>Descripción</b>	Consolida los datos de corridas, planes y monitoreos para producir PDFs con gráficos y tablas que apoyan la toma de decisiones.
<b>Flujo Principal</b>	<ul style="list-style-type: none"> <li>• Actor selecciona tipo de reporte (alimentación, deficiencias, rendimiento, cosechas, KPI globales).</li> <li>• Define filtros (fecha, corrida, tanque).</li> <li>• Sistema consulta datos, genera gráficos y compone PDF.</li> <li>• Descarga o envía por correo.</li> </ul>
<b>Subflujos</b>	<ul style="list-style-type: none"> <li>• Filtrar desviaciones (incluye) para deficiencias.</li> <li>• Calcular mortalidad (incluye) para rendimiento.</li> <li>• Renderizar PDF (incluye).</li> </ul>
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>• No hay datos para el rango seleccionado ⇒ aviso “sin resultados”.</li> <li>• Error de generación (falta fuente o gráfico) ⇒ log y mensaje de fallo.</li> </ul>

*Tabla 11: Caso de Uso. Reportes & Análisis*

### 3.2.3 Modelado de Datos

Se llevó a cabo la creación de datos utilizando MySQL y PhpMyAdmin, con un total de 17 tablas interconectadas para el seguimiento del sitio web. Se empleó el modelo de entidad-relación en el diseño y modelado de la base de datos para mostrar visualmente las entidades del aplicativo web y sus conexiones. Este modelo facilita la identificación clara y precisa de los diferentes componentes y sus dependencias. También posibilita una mayor comprensión de la estructura y lógica de la base de datos, lo que simplifica la planificación, implementación y mantenimiento del software.

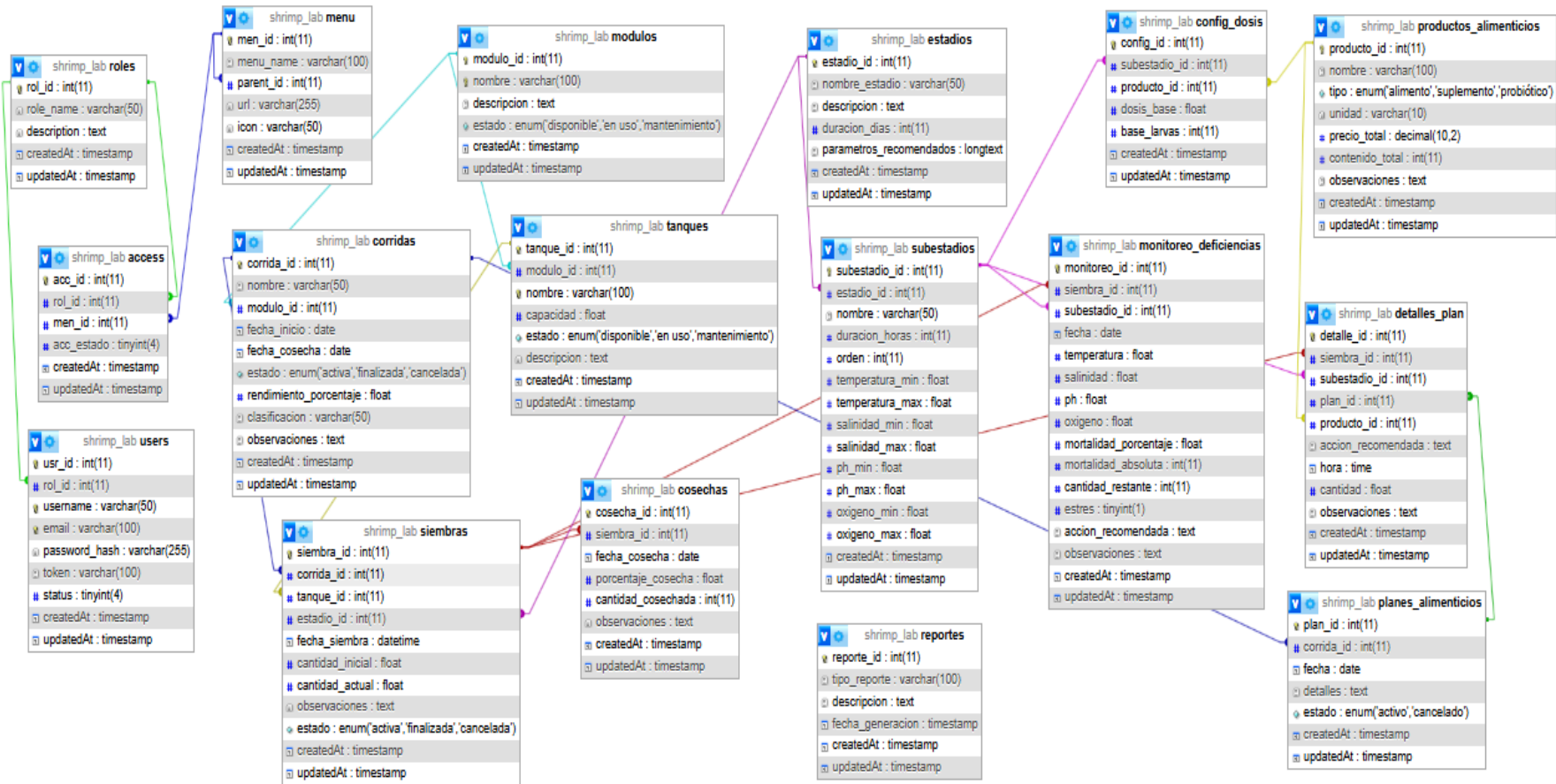
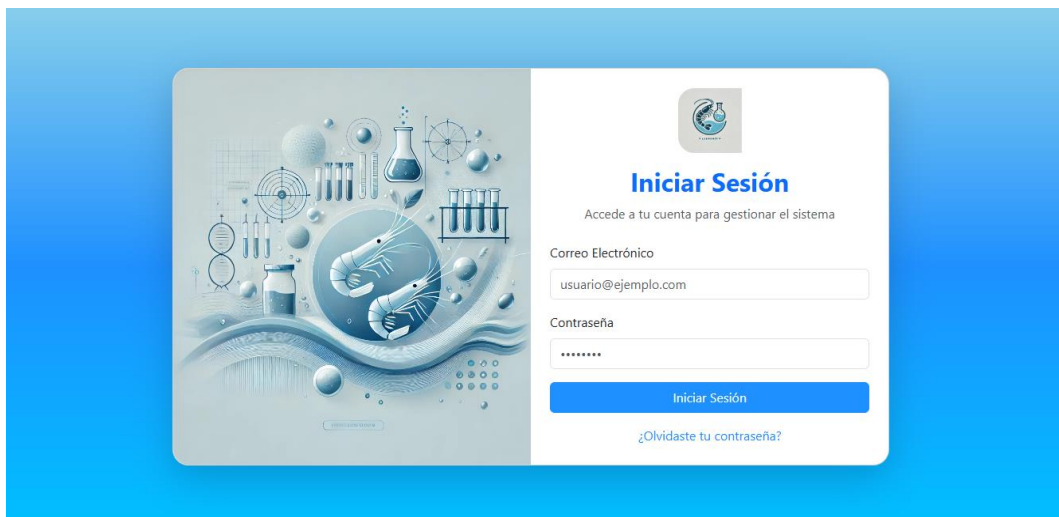


Figura 14: Modelado de Base de Datos

### 3.3 Diseño de Interfaces

#### Interfaz Inicio de Sesión

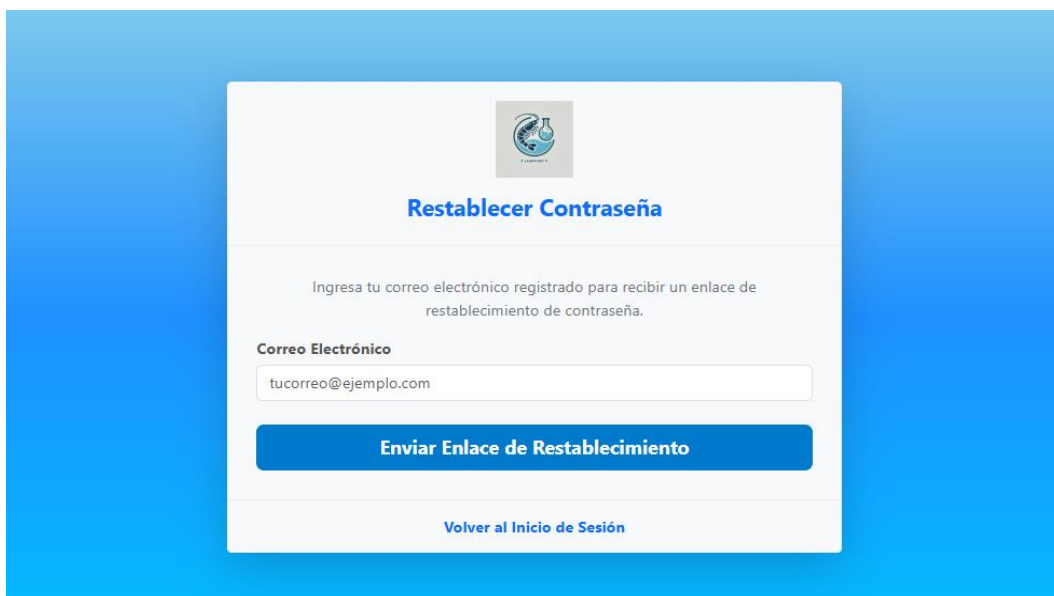
En esta pantalla inicio de sesión guía al usuario a autenticar su identidad: primero ingresa su correo electrónico y contraseña, luego al pulsar “Iniciar Sesión” el formulario realiza validaciones básicas (campos obligatorios y formato de correo), envía las credenciales al servidor para comprobarlas y, si son correctas, redirige automáticamente al usuario al panel principal; en caso de error, muestra un mensaje explicando si los datos no coinciden o si la cuenta está deshabilitada. Además, el enlace “¿Olvidaste tu contraseña?” inicia el flujo de recuperación, enviando un correo de restablecimiento para que el usuario recupere el acceso de forma segura.



*Figura 15: Interfaz – Inicio de Sesión*

#### Interfaz Restablecimiento de Contraseña

En esta pantalla de restablecimiento de contraseña, el usuario introduce su correo electrónico registrado y al pulsar “Enviar Enlace de Restablecimiento” el sistema valida que el campo no esté vacío y que tenga formato de email, luego envía la solicitud al servidor para generar y remitir un enlace único de recuperación a esa dirección; tras el envío, muestra un mensaje de confirmación o un error si el correo no existe o hay un fallo. Además, incluye un enlace “Volver al Inicio de Sesión” que retorna al usuario al formulario de acceso para intentar autenticarse de nuevo.



*Figura 16: Interfaz - Restablecimiento de Contraseña*

### **Interfaz Nueva Contraseña**

En esta pantalla el usuario ingresa y confirma su nueva contraseña tras abrir el enlace de recuperación; al enviar, el sistema valida que los campos coincidan y envía el token al servidor para actualizar la clave, mostrando un mensaje de éxito o el error correspondiente si algo falla.



*Figura 17: Interfaz - Nueva Contraseña*

## Interfaz Dashboard – Biólogo

Este dashboard muestra de un vistazo las métricas clave de producción larvaria (corridas activas, tanques en uso, larvas vivas y mortalidad), junto con gráficos de evolución de mortalidad, distribución por subestadio y tablas de datos de parámetros de calidad de agua y cantidades de alimentación diaria; todo actualizado en tiempo real.

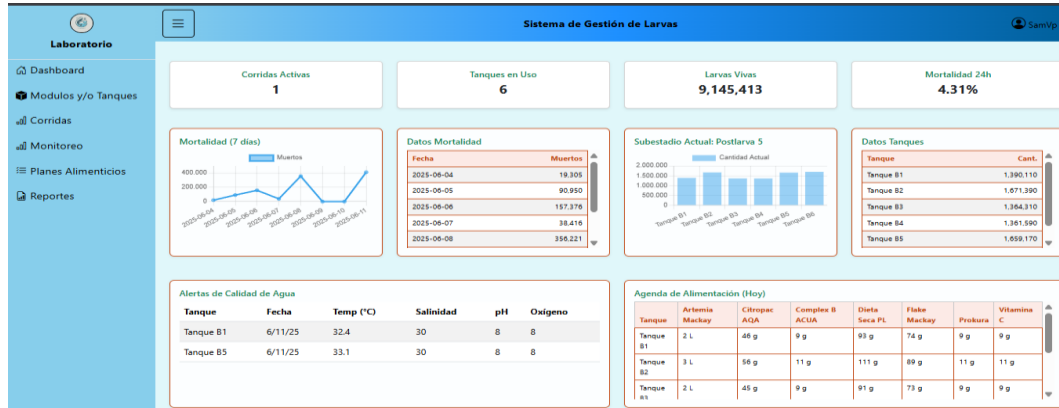


Figura 18: Interfaz – Dashboard Biólogo

## Interfaz Dashboard – Gerente

En este dashboard el usuario visualiza de un vistazo cuatro KPIs esenciales, junto a dos gráficos: uno de “Historial de Cosechas” para analizar la evolución temporal de los volúmenes cosechados y otro de “Rendimiento por Módulo” para comparar la eficiencia entre módulos.

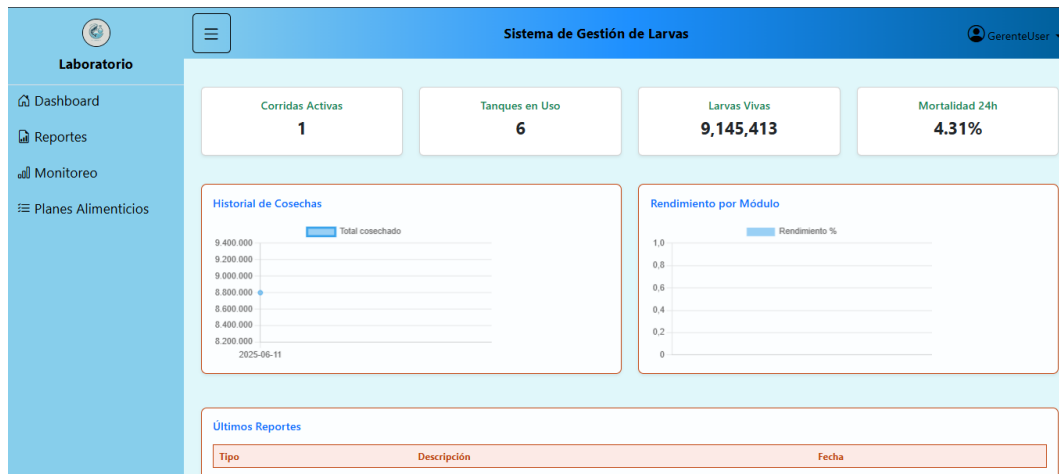


Figura 19: Interfaz - Dashboard Gerente

## Interfaz Gestión de Usuarios

Esta pantalla permite administrar las cuentas de usuario: incluye filtros para buscar por nombre, correo o rol, un botón para registrar nuevos usuarios y una tabla que muestra usuario, correo, rol y estado (activado/desactivado). Desde la columna de acciones se puede ver detalles, editar los datos o eliminar el usuario con un solo clic.

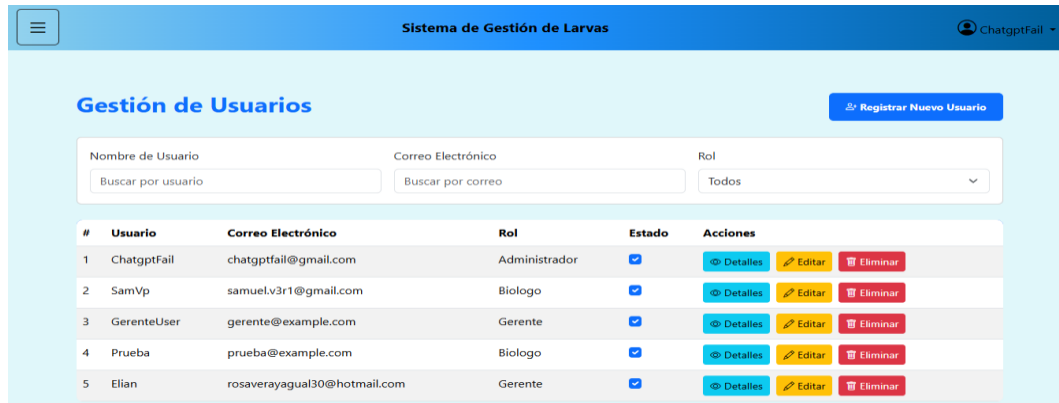


Figura 20: Interfaz - Gestión de Usuarios

## Interfaz Registro de Usuario

Este modal reúne nombre, teléfono, credenciales (correo, contraseña) y rol; al pulsar “Guardar Usuario” valida los campos y envía la información al servidor para crear la cuenta, actualizando la lista o mostrando errores si algo falla.



Figura 21: Interfaz - Registro de Usuario

## Interfaz de Módulos y Tanques

Esta pantalla permite seleccionar un módulo para ver y gestionar sus tanques (nombre, capacidad y estado) con acciones de editar o eliminar, y ofrece botones para crear nuevos módulos o tanques y eliminar el módulo activo.



#	Nombre del Tanque	Capacidad	Estado	Acciones
1	Tanque B1	1500000	EN USO	Editar Eliminar
2	Tanque B2	1850000	EN USO	Editar Eliminar
3	Tanque B3	1500000	EN USO	Editar Eliminar
4	Tanque B4	1500000	EN USO	Editar Eliminar
5	Tanque B5	2000000	EN USO	Editar Eliminar
6	Tanque B6	2000000	EN USO	Editar Eliminar

Figura 22: Interfaz - Módulos y Tanque

## Interfaz de Corridas

La pantalla de Gestión de Corridas permite filtrar al instante por nombre, módulo y estado mediante los campos en la cabecera. La tabla resultante muestra para cada corrida su nombre, módulo, fecha de inicio, fecha de cosecha y un distintivo visual de estado (Activa o Finalizada). Un botón “Nueva Corrida” abre el formulario de creación y, en cada fila, los botones Editar y Eliminar permiten modificar o borrar corridas con un solo clic.



#	Nombre	Módulo	Fecha Inicio	Fecha Cosecha	Estado	Acciones
1	Mayo/C	Modulo A	2025-05-01	2025-06-11	FINALIZADA	Editar Eliminar
2	Junio/C	Modulo B	2025-06-01	0000-00-00	ACTIVA	Editar Eliminar

Figura 23: Interfaz – Corridas

## Interfaz de Siembras

Esta pantalla permite elegir una corrida y visualizar todas las siembras asociadas en una tabla (tanque, fecha de siembra, cantidad inicial, estadio y observaciones), con botones para editar o eliminar cada registro y un botón “Registrar Nueva Siembra” para agregar nuevas entradas.



The screenshot shows the 'Siembras' interface. At the top, there is a title 'Siembras' and a subtitle 'Gestiona las siembras asociadas a una corrida'. A blue button 'Registrar Nueva Siembra' is located in the top right. Below the title, there is a dropdown menu 'Seleccionar Corrida:' with the selected option 'Junio/C (Jun 1, 2025)'. The main content is a table with the following data:

#	Tanque	Fecha de Siembra	Cantidad Inicial	Estadio Inicial	Observaciones	Acciones
1	Tanque B1	31/05/2025	1500000	Nauplio	Sin observaciones	<a href="#">Editar</a> <a href="#">Eliminar</a>
2	Tanque B2	31/05/2025	1850000	Nauplio	Siembra realizada con normalidad, capacidad máxima sembrada	<a href="#">Editar</a> <a href="#">Eliminar</a>
3	Tanque B3	31/05/2025	1500000	Nauplio	Siembra realizada con normalidad, capacidad máxima sembrada	<a href="#">Editar</a> <a href="#">Eliminar</a>
4	Tanque B4	31/05/2025	1500000	Nauplio	Siembra realizada con normalidad, capacidad máxima sembrada	<a href="#">Editar</a> <a href="#">Eliminar</a>
5	Tanque B5	31/05/2025	1850000	Nauplio	Siembra realizada con normalidad.	<a href="#">Editar</a> <a href="#">Eliminar</a>
6	Tanque B6	31/05/2025	2000000	Nauplio	Siembra realizada con normalidad, caoacidad máxima sembrada	<a href="#">Editar</a> <a href="#">Eliminar</a>

Figura 24: Interfaz – Siembras

## Interfaz de Cosechas

Esta pantalla permite seleccionar una corrida y listar sus cosechas en una tabla (tanque, fecha de cosecha, cantidad, porcentaje de rendimiento y observaciones), con botones para editar o eliminar cada registro y un botón “Registrar Nueva Cosecha” para añadir nuevas entradas.



The screenshot shows the 'Cosechas' interface. At the top, there is a title 'Cosechas' and a subtitle 'Gestiona las cosechas asociadas a una corrida'. A blue button 'Registrar Nueva Cosecha' is located in the top right. Below the title, there is a dropdown menu 'Seleccionar Corrida:' with the selected option 'Junio/C (Jun 1, 2025)'. The main content is a table with the following data:

#	Tanque	Fecha de Cosecha	Cantidad Cosechada	Porcentaje Rendimiento	Observaciones	Acciones
---	--------	------------------	--------------------	------------------------	---------------	----------

Figura 25: Interfaz – Cosechas

## Interfaz de Registro Monitoreo

Esta pantalla permite seleccionar una corrida y capturar o actualizar en lote los parámetros críticos de cada tanque (subestadio, temperatura, salinidad, pH, oxígeno), registrar mortalidad y estrés, calcular pérdidas y larvas restantes, y añadir observaciones y recomendaciones antes de guardar todos los datos de monitoreo.

**Registro de Monitoreo**  
Monitorea los parámetros críticos de cada tanque

Seleccionar Corrida:  
Junio/C (Jun 1, 2025)

#	Tanque	SubEstadio	Temperatura (°C)	Salinidad (ppt)	pH	Oxígeno (mg/L)	Mortalidad (%)	Estrés	Perdida	Restantes	Observaciones	Recomendaciones
1	Tanque B1	Postlarva 4	32.4	30	8	8		<input type="checkbox"/>	57921 larvas	1390109 larvas	Temperatura baja	Ajustar calefor
2	Tanque B2	Postlarva 4	33	30	8	8		<input type="checkbox"/>	51692 larvas	1671388 larvas	Sin observaciones fuera de rango	Sin acciones necesarias
3	Tanque B3	Postlarva 4	33	30	8	8		<input checked="" type="checkbox"/>	71806 larvas	1364314 larvas	Estrés larval detectado	revise constantemente la calidad del agua
4	Tanque B4	Postlarva 4	32.5	30	8	8		<input checked="" type="checkbox"/>	71663 larvas	1361587 larvas	Estrés larval detectado	revise constantemente la calidad del agua
5	Tanque B5	Postlarva 4	33.1	30	8	8		<input type="checkbox"/>	69132 larvas	1659168 larvas	Temperatura alta	Revisar sistema de enfriamiento o sombreado
6	Tanque B6	Postlarva 4	33	30	8	8		<input checked="" type="checkbox"/>	89413 larvas	1698847 larvas	Estrés larval detectado	revise constantemente la calidad del agua

Figura 26: Interfaz - Registro Monitoreo

## Interfaz Reporte Monitoreo

Esta pantalla de Reporte de Monitoreo permite seleccionar la corrida y la fecha, pulsar “Generar PDF” y ver en el visor integrado el informe diario con todos los datos críticos (tabla de tanques, subestadio, temperatura, salinidad, pH, oxígeno, mortalidad, estrés, observaciones y recomendaciones), con opciones para descargar o imprimir directamente el documento.

**Reporte de Monitoreo**

Corrida: Junio/C (6/1/25)      Fecha: 11/06/2025      [Generar PDF](#)

**Reporte de Monitoreo Diario**  
Corrida 2 — Fecha 2025-06-11  
Aquaculture Monitoring System      11 de Junio de 2025

#	Fecha	Tanque	Subestadio	Temperatura (°C)	Salinidad	pH	Oxígeno (mg/L)	Mortalidad (%)	Estrés	Observaciones	Recomendaciones
1	2025-06-11	Tanque B1	Postlarva 4	32.4	30	8	8	4	No	Temperatura baja	Ajustar caleforones
2	2025-06-11	Tanque B2	Postlarva 4	33	30	8	8	3	No	Sin observaciones fuera de rango	Sin acciones necesarias
3	2025-06-11	Tanque B3	Postlarva 4	33	30	8	8	5	Si	Estrés larval detectado	revise constantemente la calidad del agua
4	2025-06-11	Tanque B4	Postlarva 4	32.5	30	8	8	5	Si	Estrés larval detectado	revise constantemente la calidad del agua
5	2025-06-11	Tanque B5	Postlarva 4	33.1	30	8	8	4	No	Temperatura alta	Revisar sistema de enfriamiento o sombreado
6	2025-06-11	Tanque B6	Postlarva 4	33	30	8	8	4	Si	Estrés larval detectado	revise constantemente la calidad del agua

Figura 27: Interfaz - Reporte Monitoreo

## Interfaz Generar Plan Alimenticio

Página al elegir la corrida muestra todos los tanques con su subestadio y estado “Pendiente”, ofreciendo por cada fila un botón Recomendar / Revisar para definir el plan individual y, al final, un botón Guardar Plan Completo que valida y persiste todas las recomendaciones de una vez.



Figura 28: Interfaz - Generar Plan Alimenticio

## Interfaz Modal Registro Plan Alimenticio

Modal al hacer clic en Recomendar / Revisar se abre un formulario donde se asignan productos, dosis, horarios, frecuencias y observaciones para ese tanque; al pulsar Guardar este tanque, el modal cierra, actualiza el estado en la tabla y añade sus datos al plan general.

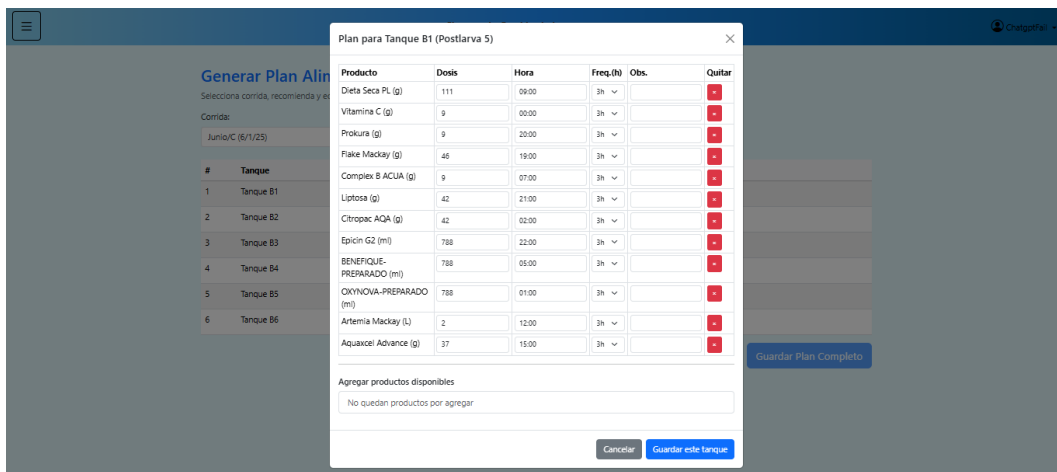


Figura 29: Interfaz - Modal Registro Plan Alimenticio

## Interfaz Reporte Plan Operativo

Al guardar el plan completo, el sistema genera automáticamente un PDF de dos páginas: **Página 1:** Tabla de alimentación por tanque con columna para cada producto (Artemia, Dieta seca, Flake, etc.), dosis y hora programada. **Página 2:** Listado de “Observaciones y Recomendaciones” agrupado por tanque (temperatura baja, estrés detectado, acciones a tomar).

N° Tanque	Artemia Mackay	Dieta Seca PL	Flake Mackay	Cloppes ADA	Complex B ACUA	Postura	Vitaminas C
Hora	08:00 PM, 08:00 AM	08:00 AM, 12:00 PM, 12:00 PM, 08:00 PM, 08:00 AM, 08:00 AM	09:00 AM, 09:00 PM	03:00 AM	09:00 PM	12:00 PM, 12:00 AM	01:00 AM, 01:00 PM
Tanque B1	2L	93 g	74 g	46 g	9 g	9 g	9 g
Tanque B2	3L	111 g	89 g	56 g	11 g	11 g	11 g
Tanque B3	2L	91 g	73 g	45 g	9 g	9 g	9 g
Tanque B4	2L	91 g	73 g	45 g	9 g	9 g	9 g
Tanque B5	3L	111 g	89 g	56 g	11 g	11 g	11 g
Tanque B6	3L	113 g	91 g	57 g	11 g	11 g	11 g

Figura 30: Interfaz - Reporte Plan Operativo

## Interfaz Historial de Planes

Esta pantalla de Historial de Planes permite filtrar por corrida, estado y rango de fechas para listar cada plan con su subestadio, fecha y estado (Activo/Inactivo), y ofrece botones para descargar el PDF o anular el plan con un solo clic.

#	Corrida	Subestadio	Fecha	Estado	Acciones
1	Junio/C	Postlarva 4	6/11/25	ACTIVO	[PDF] [X]
2	Junio/C	Postlarva 3	6/10/25	ACTIVO	[PDF]
3	Junio/C	Postlarva 2	6/9/25	ACTIVO	[PDF]
4	Junio/C	Postlarva 1	6/8/25	ACTIVO	[PDF]
5	Junio/C	Mysis 3	6/7/25	ACTIVO	[PDF]
6	Junio/C	Mysis 2	6/6/25	ACTIVO	[PDF]

Figura 31: Interfaz - Historial de Planes

## Interfaz Reporte Técnico de Alimentación

Esta pantalla permite al biólogo seleccionar la corrida, el tanque y el subestadio, pulsar “Generar PDF” y ver el **Reporte Técnico de Alimentación**, que incluye una tabla detallada con fecha, hora, producto administrado, cantidad aplicada, dosis recomendada, desviación porcentual y observaciones, así como opciones para descargar o imprimir el informe directamente.

Fecha	Hora	Subestadio	Tanque	Producto	Cant. aplicada	Dosis recomendada	Desviación (%)	Observación
2025-06-11	01:00:00	Postlarva 4	Tanque B1	Vitamina C	9	9.27	-2.89	01:00 AM, 01:00 PM
2025-06-11	03:00:00	Postlarva 4	Tanque B1	Citropac AQA	46	46.34	-0.73	03:00 AM
2025-06-11	09:00:00	Postlarva 4	Tanque B1	Dieta Seca PL	93	92.67	0.35	09:00 AM, 12:00 PM, 03:00 PM, 06:00 PM, 09:00 PM, 12:00 AM, 03:00 AM, 06:00 AM
2025-06-11	09:00:00	Postlarva 4	Tanque B1	Flake Mackay	74	74.14	-0.19	09:00 AM, 09:00 PM
2025-06-11	12:00:00	Postlarva 4	Tanque B1	Protokura	9	9.27	-2.89	12:00 PM, 12:00 AM
2025-06-11	18:00:00	Postlarva 4	Tanque B1	Artemia Mackay	2	2.32	-13.68	06:00 PM, 06:00 AM
2025-06-11	21:00:00	Postlarva 4	Tanque B1	Complex B ACUA	9	9.27	-2.89	09:00 PM

Figura 32: Interfaz - Reporte Técnico de Alimentación

## Interfaz Reporte de Deficiencias

Esta pantalla de Reporte de Deficiencias permite filtrar por corrida, tanque, subestadio y rango de fechas y, al pulsar “Generar PDF”, muestra un informe con mortalidad absoluta y porcentual, desviaciones de parámetros, acciones recomendadas y observaciones, listo para descargar o imprimir.

Fecha	Tanque	Subestadio	Mort. abs.	% Mort.	Desviaciones Rango	Acción recomendada	Observaciones
2025-06-11	Tanque B1	Postlarva 4	57921	4.00	Temp:32.4°C [32.5-33]	Ajustar calefones	Temperatura baja

Figura 33: Interfaz - Reporte de Deficiencias

## Interfaz Reporte Rendimiento Biólogo

Esta pantalla permite elegir la corrida y, al pulsar “Generar PDF”, obtener un informe que muestra un gráfico de mortalidad (%) a lo largo del tiempo junto a una tabla detallada por tanque (restantes, muertos, % mort., observaciones), listo para descargar o imprimir.



Figura 34: Interfaz - Reporte Rendimiento Biólogo

## Interfaz Reporte Rendimiento Gerente

Esta pantalla permite elegir la corrida y, al pulsar “Generar PDF”, obtener un informe de dos páginas: **Página 1:** gráfico de barras con el rendimiento (%) por tanque y su tabla correspondiente. **Página 2:** diagrama de pastel que compara sobrevivientes vs muertos y un resumen de métricas clave (total inicial, total actual, rendimiento y clasificación).



Figura 35: Interfaz Reporte Rendimiento Gerente

## Interfaz Reporte Costo de Alimentación – Cantidad

Esta pantalla permite filtrar por corrida, tanque, subestadio y fecha, y al pulsar **“Generar PDF”** crea un **Reporte de Alimentación** de dos páginas: **Página 1:** gráfico de barras y tabla con la **cantidad aplicada** de cada producto. **Página 2:** gráfico de pastel y tabla con el **costo total** de cada insumo. El informe queda disponible en el visor y se puede descargar o imprimir al instante.

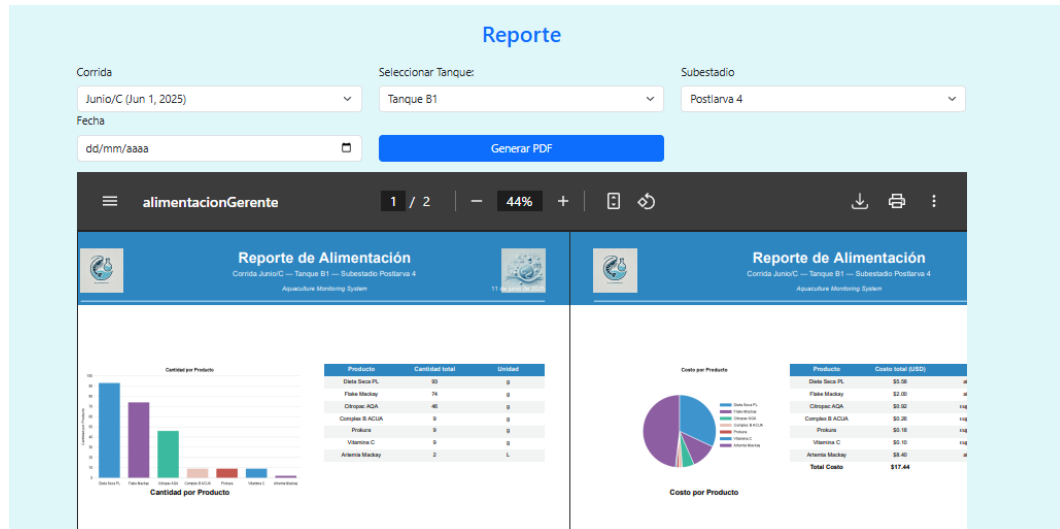


Figura 36: Interfaz - Reporte Costo Alimentación Cantidad

### 3.4 Pruebas

#### DATOS DEL CASO DE PRUEBA

Campo	Detalle
<b>CASO DE PRUEBA N.º</b>	001
<b>CASO DE USO</b>	Autenticación & Acceso
<b>OBJETIVO DE LA PRUEBA</b>	Verificar que el usuario (Administrador, Biólogo o Gerente) pueda iniciar sesión, recuperar contraseña, renovar token y cerrar sesión de forma segura.
<b>CONDICIONES</b>	<ul style="list-style-type: none"> <li>El actor debe tener una cuenta activa con credenciales válidas.</li> <li>Para recuperación: el correo debe estar registrado y accesible.</li> <li>Para renovación de token: la sesión debe haber expirado.</li> </ul>

## PASOS DE LA PRUEBA

- Abrir la pantalla de inicio de la aplicación.
- Ingresar correo y contraseña válidos y pulsar “**Iniciar Sesión**”.
- Verificar que se validen las credenciales y se genere un token, mostrando el menú según el rol.
- Pulsar “**¿Olvidaste tu contraseña?**”, introducir el correo registrado y enviar solicitud.
- Verificar recepción del email con enlace de recuperación, definir nueva clave e iniciar sesión con ella.
- Simular expiración de la sesión y comprobar que, al interactuar, el sistema renueve el token antes de expirar.
- Pulsar “**Cerrar Sesión**” y comprobar que el token se invalida y se redirige a la pantalla de login.

## RESULTADOS DE LA PRUEBA

RESULTADOS OBTENIDOS	Exitoso	Fallido
El usuario inicia sesión correctamente, recupera contraseña, renueva token antes de expirar y cierra sesión sin errores.	✓	

Tabla 12: Prueba - Inicio Sesión

## DATOS DEL CASO DE PRUEBA

Campo	Detalle
CASO DE PRUEBA N.º	002
CASO DE USO	Gestión de Usuarios
OBJETIVO DE LA PRUEBA	Verificar que el Administrador pueda crear, editar, asignar rol, activar/inactivar y eliminar usuarios, garantizando unicidad de correo y trazabilidad de cambios.
CONDICIONES	<ul style="list-style-type: none"><li>• El actor debe estar autenticado como Administrador y con sesión activa.</li><li>• Para edición o eliminación, debe existir al menos un usuario de prueba.</li></ul>

- Para creación, el correo no debe existir previamente.
- Para eliminación física, el usuario no debe tener dependencias críticas.

### PASOS DE LA PRUEBA

- Navegar a la sección “**Usuarios**”.
- Aplicar filtros opcionales (nombre, rol, estado) y comprobar que la lista se actualiza correctamente.
- Pulsar “**Nuevo**”, completar el formulario con datos válidos (nombre, correo único, rol) y guardar.
- Verificar que el sistema envía correo de confirmación y que el nuevo usuario aparece en la lista con el rol asignado.
- Seleccionar un usuario existente, editar sus datos (p. ej. cambiar nombre o rol) y guardar.
- Verificar que el registro se actualiza en la lista y que se registra un cambio en el historial de auditoría.
- Intentar crear un usuario usando un correo ya existente; comprobar que el sistema bloquea la creación y muestra un mensaje de error por duplicidad.
- Activar o inactivar un usuario (soft-delete) y comprobar que su estado cambia y que ya no aparece en la lista si está inactivo.
- Intentar eliminar físicamente un usuario sin dependencias críticas y confirmar que se elimina correctamente.
- Intentar eliminar físicamente un usuario con dependencias críticas; comprobar que la operación es bloqueada y se muestra un aviso apropiado.

### RESULTADOS DE LA PRUEBA

RESULTADOS OBTENIDOS	Exitoso	Fallido
El usuario inicia sesión correctamente, recupera contraseña, renueva token antes de expirar y cierra sesión sin errores.	✓	

*Tabla 13: Prueba - Gestión de Usuarios*

## DATOS DEL CASO DE PRUEBA

Campo	Detalle
CASO DE PRUEBA N.º	003
CASO DE USO	Módulos y Tanques
OBJETIVO DE LA PRUEBA	Verificar que el Administrador pueda crear, editar, listar y eliminar módulos de cultivo y sus tanques asociados, respetando las reglas de negocio.
CONDICIONES	<ul style="list-style-type: none"> <li>• Para edición/eliminación, debe existir al menos un módulo o tanque de prueba.</li> <li>• No debe haber tanques en uso al eliminar un módulo.</li> <li>• Al añadir tanques, la capacidad debe ser un número entero <math>&gt; 0</math>.</li> </ul>

## PASOS DE LA PRUEBA

- Ir a la sección “Módulos”.
- Pulsar “Nuevo Módulo”, completar nombre y descripción válidos y guardar.
- Verificar que el módulo aparece en la lista.
- Seleccionar el módulo recién creado, pulsar “Añadir Tanque”, ingresar capacidad válida ( $> 0$ ) y estado “Disponibile”, y guardar.
- Comprobar que el tanque aparece en el listado de tanques del módulo.
- Editar la capacidad y estado de un tanque y guardar; verificar cambios.
- Intentar eliminar el módulo mientras aún tiene tanques asociados; comprobar bloqueo de la operación y mensaje de error por dependencias.

## RESULTADOS DE LA PRUEBA

RESULTADOS OBTENIDOS	Exitoso	Fallido
El Administrador puede gestionar módulos y tanques según lo esperado: creación, edición, listado y eliminación funcionan, y se muestran errores adecuados para duplicados, dependencias y capacidad inválida.	✓	

Tabla 14: Prueba - Módulos y Tanques

## DATOS DEL CASO DE PRUEBA

Campo	Detalle
CASO DE PRUEBA N.º	004
CASO DE USO	Corridas
OBJETIVO DE LA PRUEBA	Verificar que el Administrador pueda crear, listar, actualizar, reabrir y cerrar corridas de producción, garantizando que el módulo asignado quede “En uso” durante la corrida y se libere al cerrarla.
CONDICIONES	<ul style="list-style-type: none"> <li>• Debe existir al menos un módulo en estado “Disponible”.</li> <li>• Para actualizar o cerrar, debe haber una corrida previamente creada.</li> <li>• Para reabrir, la corrida debe estar en estado “Cerrada”.</li> <li>• Para eliminar físicamente, no debe haber siembras activas asociadas a la corrida.</li> </ul>

## PASOS DE LA PRUEBA

- Navegar a la sección “Corridas”.
- Pulsar “Nueva Corrida”, completar nombre, seleccionar un módulo disponible y definir fechas de inicio y fin estimado Guardar la corrida
- El módulo queda marcado como “En uso”.
- La corrida aparece en la lista con sus datos y estado “Activa”.
- Seleccionar la corrida creada, editar sus fechas o estado, guardar y comprobar que los cambios se aplican correctamente.

## RESULTADOS DE LA PRUEBA

RESULTADOS OBTENIDOS	Exitoso	Fallido
Crear, listar, actualizar, reabrir y cerrar corridas correctamente; los estados del módulo cambian según la lógica; los intentos de usar módulos ocupados o eliminar corridas con siembras activas son bloqueados con mensajes adecuados.	✓	

Tabla 15: Prueba - Corridas

## DATOS DEL CASO DE PRUEBA

Campo	Detalle
CASO DE PRUEBA N.º	005
CASO DE USO	Siembras
OBJETIVO DE LA PRUEBA	Verificar que el Administrador pueda registrar, modificar y eliminar siembras, validando capacidad y bloqueo según reglas.
CONDICIONES	<ul style="list-style-type: none"> <li>• Sesión activa como Administrador.</li> <li>• Existe al menos una corrida y un tanque con disponibilidad.</li> <li>• No hay cosecha previa para la siembra a eliminar.</li> </ul>

## PASOS DE LA PRUEBA

- Seleccionar una corrida activa y un tanque disponible.
- Registrar una siembra ingresando cantidad, fecha y estadio; guardar y comprobar que aparece en la lista.
- Intentar registrar una siembra con cantidad que exceda la capacidad; verificar mensaje de “Capacidad excedida”.
- Editar la siembra existente (p. ej. cambiar cantidad dentro del límite) y comprobar que se actualiza.
- Eliminar la siembra y comprobar que el tanque vuelve a estar libre.
- Simular que existe una cosecha asociada y tratar de eliminar la siembra; verificar bloqueo con mensaje apropiado.

## RESULTADOS DE LA PRUEBA

RESULTADOS OBTENIDOS	Exitoso	Fallido
Registro, edición y eliminación de siembras funcionan correctamente; validaciones de capacidad y bloqueo por cosecha se muestran cómo se espera.	✓	

Tabla 16: Prueba - Siembras

## DATOS DEL CASO DE PRUEBA

<b>Campo</b>	<b>Detalle</b>
<b>CASO DE PRUEBA N.º</b>	006
<b>CASO DE USO</b>	Cosechas
<b>OBJETIVO DE LA PRUEBA</b>	Verificar que el Administrador pueda registrar cosechas, validar cantidades, respetar bloqueos por cierre de cosecha.
<b>CONDICIONES</b>	<ul style="list-style-type: none"><li>• Sesión activa como Administrador.</li><li>• Existe al menos una siembra marcada como “Finalizada”.</li><li>• No se ha ejecutado el cierre de la cosecha.</li></ul>

## PASOS DE LA PRUEBA

- Seleccionar una siembra finalizada desde la lista.
- Registrar una cosecha ingresando cantidad ( $\leq$  cantidad sembrada) y fecha; guardar y comprobar que se actualiza el historial de rendimiento.
- Intentar registrar una cosecha con cantidad superior a la sembrada; verificar mensaje de “Cantidad excede la sembrada”.
- tratar de modificar o eliminar la cosecha; verificar bloqueo con mensaje de error apropiado.

## RESULTADOS DE LA PRUEBA

<b>RESULTADOS OBTENIDOS</b>	<b>Exitoso</b>	<b>Fallido</b>
Registro, validación de cantidades, verificación del estado y bloqueo tras cierre de cosechas funcionan según lo esperado	✓	

Tabla 17: Prueba - Cosechas

## DATOS DEL CASO DE PRUEBA

Campo	Detalle
CASO DE PRUEBA N.º	007
CASO DE USO	Monitoreo
OBJETIVO DE LA PRUEBA	Verificar que el Biólogo o Administrador pueda registrar parámetros diarios (temperatura, pH, salinidad, OD), calcular mortalidad/subestadio, generar observaciones y acciones, y detectar errores de rango o duplicados.
CONDICIONES	<ul style="list-style-type: none"> <li>• Sesión activa como Biólogo o Administrador.</li> <li>• Existe al menos una corrida y un tanque disponible.</li> <li>• No hay registro previo en la misma fecha para esa combinación.</li> </ul>

## PASOS DE LA PRUEBA

- Seleccionar una corrida activa y su tanque correspondiente.
- Ingresar valores válidos de los parámetros (temperatura, pH, salinidad, OD) y guardar.
- Calcular subestadio y mortalidad correctamente.
- Mostrar observaciones y acciones sugeridas.
- Intentar guardar de nuevo para la misma fecha; comprobar mensaje de “Registro duplicado para la misma fecha”.
- Ingresar un parámetro fuera de rango sin definición de acción y verificar alerta crítica.
- (Opcional) Generar PDF del reporte diario y confirmar que se descarga con los datos ingresados.

## RESULTADOS DE LA PRUEBA

RESULTADOS OBTENIDOS	Exitoso	Fallido
El sistema registra parámetros, calcula mortalidad/subestadio y sugiere acciones; bloquea duplicados y alerta en rangos críticos; genera PDF correctamente.	✓	

Tabla 18: Prueba - Monitoreo

## DATOS DEL CASO DE PRUEBA

<b>Campo</b>	<b>Detalle</b>
<b>CASO DE PRUEBA N.º</b>	008
<b>CASO DE USO</b>	Planes Alimenticios
<b>OBJETIVO DE LA PRUEBA</b>	Verificar que el Biólogo o Administrador pueda generar y ajustar planes alimenticios, validar dosis base, gestionar planes superpuestos y exportar el PDF asociado.
<b>CONDICIONES</b>	<ul style="list-style-type: none"> <li>• Sesión activa como Biólogo o Administrador.</li> <li>• Existen datos de monitoreo recientes y configuraciones de dosis base.</li> <li>• No existe un plan previo para la combinación corrida–fecha seleccionada.</li> </ul>

## PASOS DE LA PRUEBA

- Seleccionar una corrida activa y una fecha sin plan asociado; comprobar que el sistema lista tanques, subestadios y calcula dosis recomendadas.
- Ajustar manualmente al menos una dosis, guardar el plan y verificar que queda disponible para impresión.
- Volver a intentar crear un plan para la misma corrida y fecha; comprobar mensaje de “Plan superpuesto”.
- Generar el PDF del plan y confirmar que incluye logo, tabla de productos y dosis.

## RESULTADOS DE LA PRUEBA

<b>RESULTADOS OBTENIDOS</b>	<b>Exitoso</b>	<b>Fallido</b>
El sistema calcula dosis correctamente, permite ajustes, bloquea planes duplicados y exporta un PDF completo con logo y tabla de dosis.	✓	

*Tabla 19: Prueba - Planes Alimenticios*

## DATOS DEL CASO DE PRUEBA

Campo	Detalle
CASO DE PRUEBA N.º	009
CASO DE USO	Reportes y Análisis
OBJETIVO DE LA PRUEBA	Verificar que Gerente, Biólogo o Administrador puedan generar PDFs de reportes (alimentación, deficiencias, rendimiento, cosechas) con filtros y manejo de errores.
CONDICIONES	<ul style="list-style-type: none"> <li>• Sesión activa con rol correspondiente.</li> <li>• Existen datos de corridas, monitoreos y planes en el rango seleccionado.</li> <li>• El sistema dispone de las fuentes y librerías PDF.</li> </ul>

## PASOS DE LA PRUEBA

- Seleccionar “Reportes y Análisis”, escoger tipo de reporte y aplicar filtros (fecha, corrida, tanque).
- Pulsar “Generar” y descargar el PDF; comprobar que incluye gráficos y tablas según el tipo.
- Ejecutar la misma operación para un rango sin datos; verificar mensaje “Sin resultados”.
- Simular falta de recurso (por ejemplo, eliminar una plantilla de gráfico) y comprobar que se registra el error y se muestra mensaje de fallo.

## RESULTADOS DE LA PRUEBA

RESULTADOS OBTENIDOS	Exitoso	Fallido
El sistema genera correctamente los PDFs con datos y gráficos, maneja el caso “sin resultados” y reporta fallos de generación de forma adecuada.	✓	

Tabla 20: Prueba - Reportes y Análisis

## CONCLUSIONES

- Se definió con éxito la arquitectura y el modelo de datos relacional, entre corridas de producción, parámetros ambientales y planes alimenticios. Además, el diseño permite futuras ampliaciones sin comprometer el rendimiento.
- El módulo de monitoreo registra parámetros ambientales diarios y genera acciones recomendadas ante desviaciones críticas, lo que permite intervenciones oportunas. Este mecanismo ha demostrado una mejora notable en la detección temprana de variaciones.
- El módulo de planificación de dietas calcula dosis y frecuencias de alimentación por subestadio larval, asegurando que cada fase reciba la nutrición adecuada. Se validó su precisión frente a casos controlados.
- El módulo de reportes genera documentos PDF detallados sobre corridas, alimentación, deficiencias y rendimiento, mejorando la visibilidad histórica y la toma de decisiones. Los reportes se han valorado positivamente por su claridad y utilidad.

## RECOMENDACIONES

- Validar el esquema con datos reales, optimizar índices y normalización, documentar el modelo de datos y desarrollar scripts de migración y pruebas de carga para asegurar escalabilidad. Se sugiere además revisar periódicamente la estructura para incorporar nuevos requisitos.
- Incorporar la personalización de umbrales, análisis de tendencias históricas y notificaciones automáticas (correo/SMS) para amplificar la capacidad de respuesta. Sería conveniente también evaluar la integración de alertas móviles vía app.
- Integrar retroalimentación en tiempo real para ajustes dinámicos de raciones, habilitar validación experta de los planes y ampliar la base de datos de suplementos y probióticos. Se recomienda además capacitar al personal en el uso de este módulo para maximizar su eficacia.
- Desarrollar dashboards interactivos, programar envíos automáticos de reportes y habilitar exportación a Excel/CSV para facilitar análisis complementarios y distribución. Además, se sugiere incorporar gráficos comparativos de tendencias a largo plazo.

## REFERENCIAS

- [1] G. NEWSWIRE, 30 Diciembre 2024. [En línea]. Available: <https://www.globenewswire.com/news-release/2024/12/30/3002556/0/en/Global-Shrimp-Feed-Industry-Poised-for-4-2-CAGR-Growth-Reaching-USD-6-400-0-Million-by-2035-Future-Market-Insights-Inc.html>.
- [2] E. R. Monsalve, «Farmed shrimp aquaculture in coastal wetlands of Latin America,» [En línea]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0025326X22006385>.
- [3] Y. Piedrahita, «Historical evolution, genetic improvement, mangrove reforestation, sanitary barriers and other developments,» [En línea]. Available: <https://www.globalseafood.org/advocate/shrimp-farming-industry-in-ecuador-part-1/>.
- [4] SKRETTING. [En línea]. Available: <https://www.skretting.com/en-in/blogs/hatchery-feed-management-shrimp-larvae/>.
- [5] I. Sort-Rite, 23 Octubre 2024. [En línea]. Available: <https://sort-rite.com/blog/shrimp-farming-problems-solutions/>.
- [6] HATCHERY, 14 Junio 2023. [En línea]. Available: <https://hatcheryfm.com/news/editors-picks/shrimp-ii-feed-and-management/>.
- [7] H. Ariadi, M. Fadjar, M. Mahmudi y Supriatna., «ProQuest,» Diciembre 2019. [En línea]. Available: <https://www.proquest.com/openview/b8d3a7645900d953f12efed91637972f/1?cbl=2046424&pq-origsite=gscholar>.

- [8] S. F. Mollo y D. A. Pizarro, «SciELO,» Agosto 2018. [En línea]. Available: [https://www.scielo.cl/scielo.php?pid=S0718-33052018000500055&script=sci\\_arttext](https://www.scielo.cl/scielo.php?pid=S0718-33052018000500055&script=sci_arttext).
- [9] V. R. C. Jazmani, «UPSE,» 13 Octubre 2022. [En línea]. Available: <https://repositorio.upse.edu.ec/handle/46000/8682>.
- [10] T. -J. C. Y. -J. S. J. -K. Y. J. W. a. X. -Y. L. T. -S. Lin, «Application and Development of Shrimp Farming Intelligent Monitoring System on Edge Computing,» 2021.
- [11] FAO, 2020. [En línea]. Available: <https://openknowledge.fao.org/items/2e7f55e5-ea53-4f88-948a-6f8c1d42b8ec>.
- [12] R. C.-H. J. L.-E. R. B.-G. L. M.-C. R.A. Bórquez-Lopez, «Improving feeding strategies for shrimp farming using fuzzy logic, based on water quality parameters,» Elsevier, 2018.
- [13] FAO, «Penaeus vannamei,» [En línea]. Available: [https://www.fao.org/fishery/docs/CDrom/aquaculture/I1129m/file/es/es\\_whatlegshrimp.htm](https://www.fao.org/fishery/docs/CDrom/aquaculture/I1129m/file/es/es_whatlegshrimp.htm).
- [14] G. D. y M. T. [En línea]. Available: [https://repository.library.noaa.gov/view/noaa/12407/noaa\\_12407\\_DS1.pdf?utm\\_source=chatgpt.com](https://repository.library.noaa.gov/view/noaa/12407/noaa_12407_DS1.pdf?utm_source=chatgpt.com).
- [15] La Colina, «Importancia de la etapa larvica en el cultivo del camarón,» [En línea]. Available: <https://lacolina.com.ec/importancia-de-la-etapa-larvica-en-el-cultivo-del-camaron/>.

- [16] Avance en Nutrición Acuícola, «Nutrición de Larvas de Camarón,» [En línea]. Available: <https://nutricionacuicola.uanl.mx/index.php/acu/article/view/288>.
- [17] R. S. S. T. B. a. F. M. Yathish Ramena, «<https://gsla.us/>,» [En línea]. Available: [https://gsla.us/wp-content/uploads/2023/10/Enhancing-Growth-Performance\\_SP\\_web.pdf](https://gsla.us/wp-content/uploads/2023/10/Enhancing-Growth-Performance_SP_web.pdf).
- [18] G. M. G. SERRANO, «repositorio.upse,» [En línea]. Available: <https://repositorio.upse.edu.ec/bitstream/46000/8075/1/UPSE-TBM-2022-0008.pdf>.
- [19] P. PIÑA-VALDEZ, «researchgate,» [En línea]. Available: [https://www.researchgate.net/publication/277955523\\_EFECTO\\_COMBINADO\\_DE\\_TEMPERATURA\\_Y\\_SALINIDAD\\_EN\\_EL\\_CONSUMO\\_DE\\_OXIGENO\\_EN\\_POSTLARVAS\\_DE\\_CAMARON\\_BLANCO\\_Litopenaeu\\_vanamei](https://www.researchgate.net/publication/277955523_EFECTO_COMBINADO_DE_TEMPERATURA_Y_SALINIDAD_EN_EL_CONSUMO_DE_OXIGENO_EN_POSTLARVAS_DE_CAMARON_BLANCO_Litopenaeu_vanamei).
- [20] TechTarget Contributor, «web application (web app),» TechTarget, [En línea]. Available: <https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app>.
- [21] «Angular,» [En línea]. Available: <https://angular.dev/>.
- [22] Y. D. González y Y. Fernández, «Patrón Modelo-Vista-Controlador.,» Telemática, 2012. [En línea]. Available: <https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15>.
- [23] E. Taylor, «Angular Material,» The Knowledge Academy, 28 Agosto 2023. [En línea]. Available: <https://www.theknowledgeacademy.com/blog/what-is-angular-material/>.

- [24] «What is database,» Oracle, 24 Noviembre 2020. [En línea]. Available: <https://www.oracle.com/database/what-is-database/>.
- [25] «What is a programming language,» Github, [En línea]. Available: <https://github.com/resources/articles/software-development/what-is-a-programming-language>.
- [26] «Visual Studio Code,» Visual Studio Code, [En línea]. Available: <https://code.visualstudio.com/docs>.
- [27] «XAMPP,» XAMPP, [En línea]. Available: <https://www.apachefriends.org/es/index.html>.
- [28] «PhpMyAdmin,» PhpMyAdmin, [En línea]. Available: <https://www.phpmyadmin.net/>.
- [29] R. Sheldon, «Apache,» The Knowledge Academy, [En línea]. Available: <https://www.techtarget.com/whatis/definition/Apache>.
- [30] J. Erickson, «MySQL,» Oracle, 29 Agosto 2024. [En línea]. Available: <https://www.oracle.com/mysql/what-is-mysql/>.
- [31] «Node JS,» Node JS, [En línea]. Available: <https://nodejs.org/en/about>.
- [32] «What is a rest-api,» OPC-Router, [En línea]. Available: <https://www.opc-router.com/what-is-a-rest-api/>.
- [33] «EmailJS,» EmailJS, [En línea]. Available: <https://www.emailjs.com/docs/introduction/how-does-emailjs-work/>.
- [34] A. Zola, «Bootstrap,» Techtarget, [En línea]. Available: <https://www.techtarget.com/whatis/definition/bootstrap>.

- [35] «What is Postman?,» Postman, [En línea]. Available: <https://www.postman.com/product/what-is-postman/>.
- [36] Instituto del Agua, «Historia de la Acuicultura,» [En línea]. Available: <https://institutodelagua.es/acuicultura/historia-de-la-acuicultura-en-costaricaacuicultura/>.
- [37] Global Seafood Alliance, «La industria de cultivo de camarón en Ecuador, parte 1,» [En línea]. Available: <https://www.globalseafood.org/advocate/la-industria-de-cultivo-de-camaron-en-ecuador-parte-1/>.
- [38] L. Villegas, H. Humberto , A. Núñez y N. Van, «Bioseguridad en el laboratorio: medidas importantes,» *redalyc*, vol. vol. 33, Junio 2006.
- [39] V. H. Garcia, «Introduction to Angular Framework,» 2023.
- [40] A. Freeman, «Pro Angular: Build Powerful and Dynamic Web Apps,» 2022.
- [41] «OBS Businnes School,» [En línea]. Available: <https://www.obsbusiness.school/blog/caracteristicas-y-fases-del-modelo-incremental>.
- [42] CNA-Ecuador, «Camara Nacional de Acuicultura,» 2024. [En línea]. Available: <https://www.cna-ecuador.com/estadisticas/>.
- [43] industrialmindset. [En línea]. Available: <https://industrialmindset.com/ai-en-analisis-de-comportamiento-humano/>.

## ANEXO

### Anexo 1. Observación del contexto



### UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

*"La observación se realizó en un centro de producción de larvas de camarón que opera con sistemas de monitoreo manual y planificación alimenticia basada en experiencia tradicional. A través de conversaciones con profesionales del área y la revisión de registros internos, se identificaron desafíos en los datos ambientales y la alimentación administrada. Aunque los valores de temperatura, salinidad y oxígeno disuelto son registrados periódicamente, no existe una integración directa con la planificación de dietas, lo que puede ocasionar dificultades en la adaptación de los planes nutricionales a las necesidades específicas de las larvas. Esta observación permitió estructurar las principales causas y posibles efectos de la problemática descrita en este proyecto."*