



UPSE

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

TÍTULO DEL TRABAJO DE TITULACIÓN

**IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL Y REGISTRO
EN TIEMPO REAL DE PROCESOS DE PRODUCCIÓN DE CAMARÓN
MEDIANTE EL USO DE WEBSOCKETS Y HERRAMIENTAS OPEN
SOURCE EN PROMAROSA**

AUTOR

Lucas Floreano Anthony Steven

PROYECTO DE UNIDAD DE INTEGRACIÓN CURRICULAR

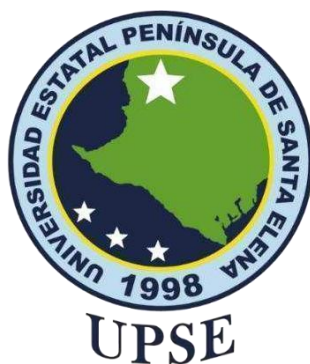
Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

TUTOR

Ing. Orozco Iguasnia Jaime Benjamín Mgt.

Santa Elena, Ecuador

Año 2025



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

TRIBUNAL DE SUSTENTACIÓN

**Ing. Jose Sancher 2., Mse.
DIRECTOR DE LA CARRERA**

**Ing. Jalme Orazo: 1 Mgt
TUTOR**

**Ing Tlan Caronel 5, Mga
DOCENTE ESPECIALISTA**

**Ind. Marioric Caronal 5, Mgn
DOCENTE GUIA 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 **LUCAS FLOREANO ANTHONY STEVEN**, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 10 días del mes de diciembre del año 2025

TUTOR



ING. JAIME OROZCO IGUASNIA. MGT



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

DECLARACIÓN DE RESPONSABILIDAD

Yo, LUCAS FLOREANO ANTHONY STEVEN

DECLARO QUE:

El trabajo de Titulación, **“Implementación de un sistema de control y registro en tiempo real de procesos de producción de camarón mediante el uso de WebSockets y herramientas Open Source en PROMAROSA”** 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 10 días del mes de diciembre del año 2025

EL AUTOR

LUCAS FLOREANO ANTHONY STEVEN



**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 **Implementación de un sistema de control y registro en tiempo real de procesos de producción de camarón mediante el uso de WebSockets y herramientas Open Source en PROMAROSA**, presentado por el estudiante, Lucas Floreano Anthony Steven fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 8%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.

**LUCAS FLOREANO ANTHONY
STEVEN**



Nombre del documento: LUCAS FLOREANO ANTHONY STEVEN.pdf
ID del documento: 45e8307881ef7c5104038859d483635fe60f296
Tamaño del documento original: 1021,76 kB

Depositante: JAIME BENJAMÍN OROZCO IGUASNIA
Fecha de depósito: 10/12/2025
Tipo de carga: interface
fecha de fin de análisis: 10/12/2025

Número de palabras: 15.911
Número de caracteres: 114.287

Ubicación de las similitudes en el documento:

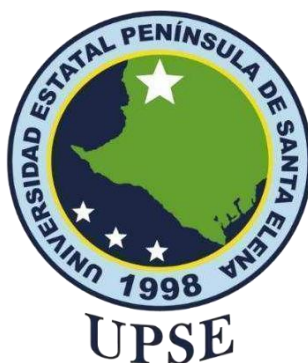


TUTOR



**JAIME BENJAMÍN
OROZCO IGUASNIA**
Validado digitalmente con FimafEC

ING. JAIME OROZCO IGUASNIA. MGT



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

AUTORIZACIÓN

Yo, LUCAS FLOREANO ANTHONY STEVEN

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 10 días del mes de diciembre del año 2025

EL AUTOR

LUCAS FLOREANO ANTHONY STEVEN

AGRADECIMIENTO

Gracias a mi familia, que creyó en mí y me apoyó de forma incondicional, gracias a su esfuerzo. Ustedes son la raíz, el refugio y la certeza de que podía seguir, incluso en los días grises.

Gracias a mis mejores amigos: Helen Reyes, Jeancarlos Neira, Ivian Reyes, Valeria Duma, Juan Angulo, Jefferson Pozo, Gabriel Díaz, Joseph Nowak y Steven Panimboza; quienes con risas tejieron puentes sobre el cansancio, que escucharon y me recibieron sin juzgar, que celebraron hasta los pequeños logros y estuvieron ahí cuando lo necesitaba.

A mis docentes, que, con paciencia, comprensión y el alma, impartieron conocimientos y deseos de aprender. Gracias ing. Marjorie Coronel.

Este trabajo lleva mi nombre, pero está escrito con la huella de muchos. Creí que estaría solo, pero ustedes me acompañaron, este logro no es mío: es nuestro...

...A quienes fueron mi hogar en el camino.

Anthony Steven, Lucas Floreano

DEDICATORIA

A quienes alguna vez soñaron con graduarse, pero la vida, con sus giros impredecibles, los llevó por otros caminos.

A quienes tuvieron que dejar las aulas por cuidar a alguien, para sostener un hogar.

A esa madre que dejó sus libros para criar a sus hijos, al joven que cambió sus cuadernos por un empleo, a esos amigos que no pudieron, a quienes aún esperan su momento.

Y, especialmente, a quienes partieron de este mundo sin poder concluir lo que un día comenzaron. Amigos, conocidos, compañeros de carpeta, de pasillos, de ideas, que se fueron con un sueño en pausa.

Hoy les dedico estas páginas como un homenaje. Porque, aunque no llegaron a una meta visible, su presencia sigue caminando con nosotros, en las memorias, en las historias compartidas.

Este logro es de todos, quienes nos inspiraron a continuar.

... a mi amigo Kleitver Balón.

Anthony Steven, Lucas Floreano

ÍNDICE GENERAL

TITULO DEL TRABAJO DE TITULACIÓN	I
TRIBUNAL DE SUSTENTACIÓN	II
CERTIFICACIÓN	III
DECLARACIÓN DE RESPONSABILIDAD	IV
DECLARO QUE:	IV
CERTIFICACIÓN DE ANTIPLAGIO	V
AUTORIZACIÓN	VI
AGRADECIMIENTO	VII
DEDICATORIA	VIII
ÍNDICE GENERAL	IX
ÍNDICE DE TABLAS	XIV
ÍNDICE DE FIGURAS	XVI
RESUMEN	XVIII
ABSTRACT	XIX
CAPÍTULO I	
1.1. ANTECEDENTES	4
1.2. DESCRIPCIÓN DEL PROYECTO	7
1.3. OBJETIVOS DEL PROYECTO	8
1.3.1. Objetivo General	8
1.3.2. Objetivos Específicos	8
1.4. JUSTIFICACIÓN DEL PROYECTO	9
1.5. ALCANCE DEL PROYECTO	10
1.6. METODOLOGÍA DEL PROYECTO	12

1.6.1. Metodología de Investigación	12
1.6.2. Beneficiarios del Proyecto	12
1.6.3. Variables	13
1.6.4. Técnicas de Recolección de Información	13
1.6.5. Análisis de Recolección de Datos	13
1.7. METODOLOGÍA DE DESARROLLO	14
1.7.1. Fases de Desarrollo Incremental	14
1.7.1.1. Inicio y análisis de requerimientos	14
1.7.1.2. Diseño del modelo de datos y arquitectura	15
1.7.1.3. Desarrollo incremental de módulos	15
1.7.1.4. Implementación	15
1.7.1.5. Pruebas	16
1.7.1.6. Evaluación y Revisión	16
CAPÍTULO II	
2.1. MARCO CONTEXTUAL	17
2.1.1. PROMAROSA	17
2.1.1.1. Ubicación	17
2.1.2. Justificación Tecnológica	18
2.1.3. Impacto Esperado	18
2.1.4. Base Legal	18
2.1.4.1. Constitución de la República del Ecuador (2008)	18
2.1.4.3. Reglamento General de Protección de Datos (GDPR, UE)	19
2.1.4.4. Código Orgánico Integral Penal (Sección Sexta)	19
2.2. MARCO CONCEPTUAL	20
2.2.1. Industria Camaronera Ecuatoriana: Líder Mundial en Producción	20

2.2.2. Fases del Procesamiento de Camarón	20
2.2.2.1. Recepción de Materia Prima	20
2.2.2.2. Control de Calidad	21
2.2.2.3. Clasificación y Procesamiento	21
2.2.2.4. Masterizado	21
2.2.3. Trazabilidad en la Industria Acuícola	22
2.2.4. Aplicaciones móviles vs. Aplicaciones de software	22
2.2.4.1. Aplicaciones móviles	22
2.2.4.2. Aplicaciones web	22
2.2.5. Beneficios de Herramientas Open Source en el Desarrollo de Software	23
2.2.6. Herramientas	23
2.2.6.1. Entorno de desarrollo	23
2.2.6.1.1. Visual Studio Code	23
2.2.6.2. Entorno de ejecución	23
2.2.6.2.1. Node.js	23
2.2.6.3. Lenguaje de programación	24
2.2.6.3.1. JavaScript	24
2.2.6.4. Gestor de base de datos	24
2.2.6.4.1. MySQL Workbench	24
2.2.6.4.2. phpMyAdmin	24
2.2.6.5. Servidor	24
2.2.6.5.1. Clever Cloud	24
2.2.6.5.2. GitHub	25
2.2.6.5.3. Render	25
2.2.6.6. Entorno local de desarrollo	25

2.2.6.6.1. XAMPP	25
2.2.6.7. Framework	25
2.2.6.7.1. Angular	25
2.2.6.7.2. Ionic	26
2.2.6.8. API	26
2.2.6.8.1. APIRESTful	26
2.2.6.8.2. WebSockets	26
2.2.6.9. Herramientas de prueba	26
2.2.6.9.1. Postman	26
2.3. MARCO TEÓRICO	27
2.3.1. Importancia de la Trazabilidad en la Cadena Productiva	27
2.3.2. Eficiencia Técnica con Arquitectura Cliente-Servidor	27
2.3.3. Integración Tecnológica para la Optimización Productiva	28
2.4. REQUERIMIENTOS	28
2.4.1. Requerimientos Funcionales	28
2.4.2. Requerimientos Técnicos	30
2.5. COMPONENTE DE LA PROPUESTA TECNOLÓGICA	32
2.5.1. Fases del proyecto	32
2.5.1.1. Fase 1: Inicio y análisis de requerimientos	32
2.5.1.2. Fase 2: Diseño del modelo de datos y arquitectura	33
2.5.1.2.1. Identificación de herramientas iniciales	33
2.5.1.2.2. Preparación del Entorno de Desarrollo	34
2.5.1.2.3. Diseño de la Arquitectura	36
2.5.1.3. Fase 3: Desarrollo incremental de módulos	36
2.5.1.4. Fase 4: Implementación	48

2.5.1.4.1. Configuración del entorno para el despliegue	49
2.5.1.4.2. Integración de los módulos del sistema	49
2.5.1.4.3. Implementación en dispositivos móviles	49
2.5.1.4.4. Validación de los procesos operativos	50
2.5.1.4.5. Procesamiento de Datos Reales	52
2.5.1.5. Fase 5: Pruebas	53
2.5.1.5.1. Pruebas unitarias	53
2.5.1.5.2. Pruebas de integración	53
2.5.1.5.3. Pruebas de rendimiento	53
2.5.1.5.4. Evaluación con Datos Reales	54
2.5.1.5.5. Evaluación Final	54
2.5.1.6. Fase 6: Evaluación y Revisión	56
2.6. DIAGRAMAS DE CASOS DE USO	57
2.7. RESULTADOS	60
CONCLUSIONES	65
RECOMENDACIONES	67
REFERENCIAS	68

ÍNDICE DE TABLAS

Tabla 1: Número de beneficiarios	12
Tabla 2: Problemas detectados por área (Hallazgos de entrevistas y observación)	14
Tabla 3: Requerimientos Funcionales - Autenticación y Seguridad	28
Tabla 4: Requerimientos Funcionales - Módulos Móviles	29
Tabla 5: Requerimientos Funcionales - Módulos Web	29
Tabla 6: Requerimientos Funcionales - Integración y Sincronización	29
Tabla 7: Requerimientos Técnicos - Configuración Mínima para Clientes	30
Tabla 8: Requerimientos Técnicos - Configuración Recomendada para Clientes	30
Tabla 9: Requerimientos Técnicos - Servidor y Base de Datos	31
Tabla 10: Requerimientos Técnicos - Servidor y Base de Datos	31
Tabla 11: Comparación de Herramientas en Fase 2	32
Tabla 12: Componentes de la Arquitectura	33
Tabla 13: Detalles Incremento 1	36
Tabla 14: Detalles Incremento 2	37
Tabla 15: Detalles Incremento 3	39
Tabla 16: Detalles Incremento 4	42
Tabla 17: Detalles Incremento 5	45
Tabla 18: Detalles Incremento 6	46
Tabla 19: Detalles Incremento 7	48
Tabla 20: Comparación Tiempos Módulo 2 (Recepción)	50
Tabla 21: Comparación Tiempos Módulo 3 (Calidad)	51
Tabla 22: Comparación Tiempos Módulo 4 (Clasificación)	51
Tabla 23: Comparación Tiempos Módulo 5 (Masterizado)	52
Tabla 24: Tiempos usando Postman	54
Tabla 25: Tiempos Pruebas Reales	54
Tabla 26: Precisión de cálculos automatizados	55
Tabla 27: Tabla de Descripción del Actor “Usuario Recepción”	57
Tabla 28: Tabla de Descripción del Actor “Usuario Calidad”	58
Tabla 29: Tabla de Descripción del Actor “Usuario Clasificación”	58

Tabla 30: Tabla de Descripción del Actor “Usuario Masterizado”	59
Tabla 31: Tabla de Descripción del Actor “Usuario Gerente”	60
Tabla 32: Consistencia de la Trazabilidad (Módulo M7)	60
Tabla 33: Consistencia de la Trazabilidad (Módulo M7)	61
Tabla 34: Tiempos de Registro por Módulo (Pruebas en Dispositivo Móvil Ionic)	62
Tabla 35: Latencias y Rendimiento del Sistema en Tiempo Real (WebSockets)	63
Tabla 36: Impacto en Tiempos Operativos Antes vs. Después del Sistema	63

ÍNDICE DE FIGURAS

Figura 1: Áreas de procesos de producción de camarón [5]	5
Figura 2: Proceso de Metodología	16
Figura 3: Vista del mapa de matriz PROMAROSA	17
Figura 4: Conexión WebSockets [61]	33
Figura 5: Ejemplo de conexión WebSocket [74]	34
Figura 6: Creación de archivos necesarios para la API	35
Figura 7: Creación de archivos necesarios para la App Móvil	35
Figura 8: Código base para módulo de autenticación	37
Figura 9: Opciones del usuario Recepción (Peso Promedio como Próxima mejora)	38
Figura 10: Listado de vehículos / Detalles del vehículo seleccionado / Editar vehículo / Nuevo vehículo	38
Figura 11: Registro de un nuevo lote (Lote C-200-2025)	39
Figura 12: Opciones del usuario Calidad	40
Figura 13: Listado de Defectos Registrados	40
Figura 14: Listado de Controles de Calidad Registrados	41
Figura 15: Control de Calidad detallado del lote C-006-2025	41
Figura 16: Registro de un nuevo Control de Calidad	42
Figura 17: Opciones del usuario Clasificación	43
Figura 18: Información de Lotes registrados	43
Figura 19: Registro de Ingreso de Túnel (Ingreso de Coche)	44
Figura 20: Generación de una nueva Liquidación (Entero)	44
Figura 21: Opciones del usuario Masterizado	45
Figura 22: Listado de Masterizado	45
Figura 23: Registro de Masterizado	46
Figura 24: Opciones del usuario Gerencia	46
Figura 25: Listado de Ordenes de Producción	47
Figura 26: Historial de un Lote	47
Figura 27: Pantalla de Reporte de Rendimiento por Proceso en la Aplicación	48
Figura 28: Código de Cálculo Rendimiento	52
Figura 29: Diagrama de uso de usuario Recepción	57

Figura 30: Diagrama de uso de usuario Calidad	57
Figura 31: Diagrama de uso de usuario Clasificación	58
Figura 32: Diagrama de uso de usuario Masterizado	59
Figura 33: Diagrama de uso de usuario Gerencia	59

RESUMEN

El presente trabajo aborda la implementación de un sistema de control y registro en tiempo real para los procesos de producción de camarón en la empacadora PROMAROSA, utilizando WebSockets y herramientas Open Source. El objetivo principal es reducir el tiempo de generación de reportes, mejorando la disponibilidad inmediata de información, la trazabilidad de lotes y la eficiencia operativa, entendida como la capacidad para acceder rápidamente a datos actualizados, facilitar la creación de informes detallados y agilizar la toma de decisiones informadas, lo que minimiza retrasos en los flujos de trabajo. El método empleado incluye el análisis de procesos actuales, modelado de una base de datos relacional (PROMAROSA_BD), diseño de una API RESTful, desarrollo de una aplicación móvil para recolección de datos en campo y una aplicación web para consulta y reportes. Como resultados esperados, se logra una reducción del 80% en tiempos de reportes (de 30-60 minutos a 5-10 minutos), mayor precisión en registros y mejor coordinación entre áreas. En conclusión, esta solución tecnológica fortalece la competitividad de PROMAROSA al optimizar sus operaciones con herramientas accesibles y escalables.

Palabras claves: Open Source, procesos de producción, WebSockets.

ABSTRACT

The main objective is to reduce report generation time, improving the immediate availability of information, batch traceability, and operational efficiency, understood as the ability to quickly access updated data, facilitate the creation of detailed reports, and expedite informed decision-making, thus minimizing workflow delays. The methodology employed includes the analysis of current processes, the modeling of a relational database (PROMAROSA_BD), the design of a RESTful API, the development of a mobile application for field data collection, and a web application for queries and reports. Expected results include a reduction of at least 80% in report generation time (from 30-60 minutes to 5-10 minutes), greater accuracy in records, and improved coordination between departments. In conclusion, this technological solution will strengthen PROMAROSA's competitiveness by optimizing its operations with accessible and scalable tools.

Keywords: Open Source, production processes, WebSockets.

INTRODUCCIÓN

En la realidad actual de la industria camaronera en Ecuador, las empacadoras como PROMAROSA se enfrentan a varios desafíos que surgen por seguir utilizando métodos tradicionales para registrar y controlar sus procesos diarios. Estos métodos, basados en anotaciones manuales en cuadernos y hojas de cálculo, generan demoras en la actualización de datos, errores de transcripción y limitaciones en la coordinación entre áreas clave como recepción, calidad, clasificación y masterizado. Como consecuencia, se afectan la trazabilidad de los lotes, la generación oportuna de reportes y la capacidad para tomar decisiones informadas, lo que impacta la eficiencia operativa —entendida como la rapidez en el acceso a información veraz y actualizada, la facilidad para producir informes detallados y la agilidad en los flujos de trabajo para minimizar interrupciones—.

El presente proyecto nace como una respuesta a estas dificultades, propone un sistema integral que permita registrar los procesos de producción en tiempo real dentro de la empresa. Para lograrlo, se utiliza tecnología de protocolo WebSocket y herramientas de código abierto mediante una aplicación móvil que facilita la recolección de datos en campo y una aplicación web destinada a la consulta. Se busca organizar y modernizar cada etapa de los procesos de producción y con ello, que la información se vuelva inmediata y accesible desde la llegada de la Materia Prima —en recepción— hasta masterizado. Este enfoque se ajusta a las tendencias tecnológicas actuales, que promueven la comunicación bidireccional y la integración de datos, aportando a una operación más ágil, eficiente y competitiva dentro del sector.

La estructura del presente trabajo está distribuida en capítulos que abordan la fundamentación teórica y contextual, la propuesta detallada, y los aspectos metodológicos, conclusiones y recomendaciones para su aplicación práctica. De esta manera, no solo se resuelve una necesidad de PROMAROSA, sino que se ofrece un modelo replicable para otras empacadoras en Ecuador, fomentando la

adopción de tecnologías accesibles para elevar los estándares de productividad y sostenibilidad en la industria.

El uso constante de métodos manuales como cuadernos y formularios físicos —u hojas de cálculo en el mejor de los casos— para realizar el registro de los procesos de producción provoca atrasos en la actualización de datos, errores en la transcripción y una limitada comunicación entre las áreas de la empresa: recepción, calidad, clasificación y masterizado. Todo esto dificulta de manera indirecta los procesos y afecta directamente a la trazabilidad de los lotes y la rapidez con la que pueden generarse informes de suma importancia. Como resultado, afecta la eficiencia general de las operaciones, especialmente en lo que respecta al acceso oportuno a datos precisos, la elaboración de reportes detallados y la agilidad requerida para prevenir errores o interrupciones en el flujo de trabajo.

Bajo este contexto, y con el objetivo de solucionar estas dificultades, el presente proyecto propone el desarrollo de un sistema integral capaz de registrar los procesos principales de producción en tiempo real. La solución se apoya en el uso del protocolo WebSocket y tecnologías Open Source, combinando una aplicación móvil para la captura rápida de información en producción y una aplicación web destinada a la consulta y análisis de los procesos. Esta resolución busca modernizar y estructurar de manera más eficiente las labores productivas diarias, garantizando la disponibilidad inmediata de información y fortaleciendo la trazabilidad desde la recepción de la Materia Prima hasta masterizado.

CAPÍTULO I

1. FUNDAMENTACIÓN

1.1. ANTECEDENTES

El constante progreso tecnológico que día a día se da en los distintos ámbitos sociales, empresariales, académicos, entre otros, y la creciente complejidad de las empresas e instituciones ha incrementado significativamente la necesidad de tener información confiable, veraz y oportuna en todo momento [1]. Para poder llevar a cabo esto, se requiere de herramientas de Software que ayuden en la minimización de pérdida de tiempo y maximización de resultados [2]. Por esta razón, en la actualidad, tanto las empresas industriales como comerciales, han adoptado los cambios y evoluciones tecnológicas de este tiempo y que se convierten en una herramienta de mucha utilidad para el desarrollo. Esto contribuye a un mejor desempeño del personal y eficacia en los procesos, lo que da como resultado un mejor servicio [3].

La tecnología permite a las empresas incrementar su rentabilidad al optimizar sus procesos y fomentar un entorno innovador que les facilita adaptarse a un futuro incierto [4]. Tanto los pequeños negocios comerciales, como las grandes sociedades industriales y de servicios, han realizado un esfuerzo clave para sistematizar determinadas tareas teniendo como instrumento fundamental las TIC's [5]. Empujados por la necesidad de aumentar la competitividad y eficacia de sus empresas, los responsables han introducido los medios informáticos, primero en funciones rutinarias, como contabilidad y gestión de almacenes para incorporarlos más adelante, incluso, en tareas muy especializadas [6].

Es así como Productos del Mar Santa Rosa (PROMAROSA) dedicada al procesamiento y comercialización de camarones y pelágicos frescos y congelados, empackadora matriz ubicada en la provincia de Santa Elena, parroquia Chanduy, se ha consolidado como una empackadora estratégica en la cadena de suministro acuícola nacional [7]. Su área de producción principal está enfocada en el tratamiento y empaque de camarón fresco y congelado, el cual es distribuido a clientes nacionales e internacionales.

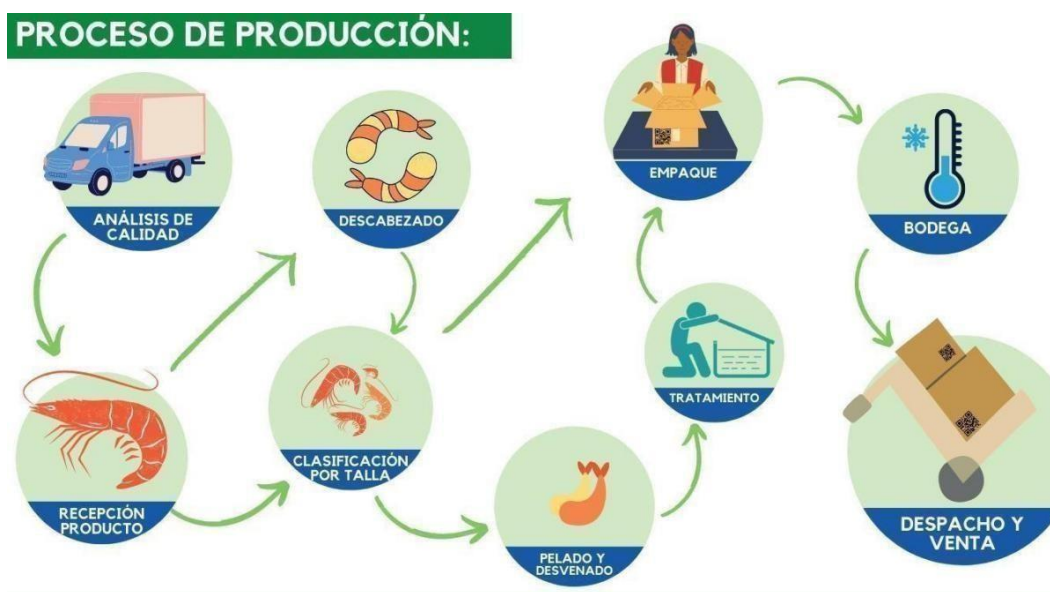


Figura 1: Áreas de procesos de producción de camarón [5]

PROMAROSA lleva un control de los procesos basados en órdenes de compra cuando los clientes lo requieren. Estos datos son registrados en cuadernos, y en algunos casos, en hojas de cálculos que contienen la información necesaria para llevar a cabo el proceso de producción. Mediante una observación aplicada (Ver **Anexo 1**) y una entrevista a los supervisores de la empresa (Ver **Anexo 2**), se identificó que PROMAROSA enfrenta diversos problemas debido a su metodología de trabajo donde se identifican tres causas principales.

La primera causa —antes mencionada— es el registro manual propenso a errores, debido a que la información sobre cada lote recibido se gestiona en cuadernos y a veces hojas de cálculo de Excel, lo que conlleva errores de transcripción y demoras en la actualización de datos. La segunda causa es la falta de comunicación entre las distintas áreas, donde la información no fluye de manera eficiente entre los supervisores de recepción y clasificación —por ejemplo—, generando confusión y descoordinación en el proceso. Por último, la tercera causa es el nulo acceso a datos en tiempo real, lo que impide a los supervisores realizar un seguimiento adecuado de lotes en proceso.

Basándonos en estas situaciones, se identificaron tres problemas principales. El primer problema presentado lo tenemos en las demoras en el registro que afectan

directamente la eficiencia de la producción, esto porque la carga de trabajo y el uso de métodos manuales hacen que la captura de datos sea lenta y propensa a retrasos y errores. La acumulación de pedidos sin procesar impacta en la entrega oportuna de los lotes, ya que la falta de coordinación entre áreas —recepción y clasificación como principales— genera retrasos que podrían evitarse con una mejor organización. Por último, la ausencia de información actualizada en tiempo real impide a los supervisores estar al tanto de todo y tomar medidas sobre procesos actuales, lo que reduce su capacidad para gestionar la producción de manera ágil y dinámica.

Estudiantes de la Facultad de Ingeniería en Mecánica y Ciencias de la Producción (FIMCP) de la Escuela Superior Politécnica del Litoral (ESPOL) realizaron un diagnóstico de la Empresa XYZ (adaptando este nombre por protección y reputación de la Empacadora) donde se observó que una de las causas raíz en el Área de Clasificado es la demora en la entrega de reportes, generando esto un retraso entre procesos [8]. Manifestaron que la Implementación de un Sistema de Control en dicha Empacadora de Camarón aumentaría la Eficiencia en una Línea de Producción, siendo su principal objetivo gestionar la administración y el control de la Producción de la empresa, con la implementación de controles y estrategias que mejoren la eficiencia.

Bajo la misma problemática, se exploró el uso de tecnologías web con lenguaje PHP, librerías de JavaScript y base de datos en MySQL en Negocios Industriales Real S.A. (NIRSA) [1]. Wilmer Crespín, ingeniero de la Universidad Estatal Península de Santa Elena (UPSE), implementó una propuesta tecnológica denominada “Desarrollo de una aplicación bajo entorno web que permita llevar el inventario y planificación de la producción en el área empacadora de Posorja”. En el momento en el que estamos, en el que la tecnología avanza a pasos agigantados, todas las empresas han dedicado parte de sus recursos a seguir los pasos necesarios para no quedarse “anticuados” y tecnologías como esta son necesarias para alcanzar ese objetivo [9].

Pero, según un estudio realizado por el Departamento de Ciencias de la Computación en la Universidad de Lagos, Nigeria, nos menciona algo sobre la

tecnología que usó Crespín, indican que el modelo tradicional de comunicación web funciona mediante solicitudes HTTP iniciadas por el cliente, estableciendo una conexión TCP para cada solicitud [10]. Una vez que el servidor responde, la conexión se cierra. En el caso del WebSocket, cuando se establece la conexión, la API WebSocket proporciona una interfaz de lectura y escritura para leer y escribir datos a través de la conexión establecida de un modo dúplex asíncrono [11]. Este modelo no permite que los servidores envíen datos en tiempo real a los clientes, por lo que se han utilizado tecnologías como Ajax, consulta prolongada y Comet.

La comunicación en tiempo real dio un gran paso adelante con la incorporación del protocolo WebSocket y las tecnologías introducidas con HTML5. WebSocket permite que el navegador y el servidor se comuniquen de forma bidireccional sin necesidad de abrir múltiples conexiones HTTP, lo que reduce tanto la latencia como el tráfico en la red [10]. Gracias a estas ventajas, esta tecnología se ha convertido en una herramienta fundamental para mejorar la interacción entre los usuarios y los servidores.

Todo esto hace que sea indispensable contar con una solución tecnológica que facilite el control adecuado del producto dentro de las órdenes de pedido de la empacadora PROMAROSA. Un sistema de este tipo permitiría gestionar adecuadamente los procesos de producción, fortalecer el seguimiento de este —lo que comúnmente se conoce como trazabilidad—y garantizar un acceso oportuno a información actualizada.

1.2. DESCRIPCIÓN DEL PROYECTO

Ante la necesidad de PROMAROSA de contar con un sistema que mejore el control y registro de sus procesos productivos, este proyecto tiene como objetivo desarrollar e implementar una aplicación móvil y una aplicación web para la digitalización y sistematización de las actividades operativas en la empacadora. Por todo lo expuesto anteriormente y basándonos en la idea del software InforFish [12], este proyecto es una respuesta a los actuales desafíos de la empresa.

Para desarrollar esta solución tecnológica, se propone el uso del protocolo WebSocket de comunicación bidireccional para actualización dinámica y continua

de datos a través del modelamiento de la información, creación de la base de datos PROMAROSA_BD y la construcción de módulos basados en cada uno de los roles. Los módulos propuestos en el sistema están interconectados y se diseñaron para cubrir cada etapa del proceso de producción debido a su forma secuencial —cada módulo es dependiente de su anterior—. Además, la aplicación será utilizada por seis tipos de usuarios, cada uno con roles y funciones específicas: recepción, calidad, liquidación, masterizado, gerencia y administración.

1.3. OBJETIVOS DEL PROYECTO

1.3.1. Objetivo General

Implementar un sistema de control y registro en tiempo real para los procesos de producción de camarón en PROMAROSA, utilizando tecnologías WebSocket y herramientas Open Source, que permita reducir el tiempo de generación de reportes, mejorando la disponibilidad de información, la trazabilidad de los lotes y la eficiencia operativa.

1.3.2. Objetivos Específicos

- ✓ Analizar los procesos actuales de generación de reportes en PROMAROSA, identificando los factores que provocan demoras y proponiendo mejoras orientadas a la sistematización.
- ✓ Modelar la base de datos relacional PROMAROSA_BD para permitir el almacenamiento estructurado y seguro de los datos de producción.
- ✓ Diseñar una API RESTful que permita el acceso eficiente y en tiempo real a la información requerida para generar reportes, optimizando así los tiempos de respuesta.
- ✓ Desarrollar una aplicación móvil orientada a la recolección y sincronización inmediata de datos del proceso productivo que integre funcionalmente los procesos de recepción de Materia Prima, evaluación de calidad, clasificación y procesamiento y masterizado, disminuyendo el desfase temporal entre la operación y el reporte.

- ✓ Desarrollar una aplicación web con un módulo específico de generación de reportes y visualización de estadísticas que permita reducir el tiempo dedicado a esta tarea por parte del personal administrativo y de supervisión.

1.4. JUSTIFICACIÓN DEL PROYECTO

La tecnología se ha transformado en un recurso fundamental para diversas actividades en múltiples sectores, modificando tanto los métodos de gestión como las formas de operar, lo que llevó a un impacto en el ámbito empresarial [13]. La tecnología deja de ser un lujo y el uso de esta se volvió fundamental en todo ámbito. Alrededor del mundo, las empresas la aprovechan el avance tecnológico al máximo para resolver problemas y eliminar barreras a través de sistemas y soluciones innovadoras que son fácilmente adaptables a lo que necesita cada una [3].

En la actualidad, PROMAROSA enfrenta varios problemas porque aún depende de registros manuales en cuadernos, formularios físicos y, en el mejor de los casos, hojas de cálculo —diseñadas por los supervisores, en un esfuerzo de simplificar procesos—. Aunque estos métodos son los comunes, y sea difícil dejarlos de lado, aumentan la posibilidad de que se cometan errores que llevan a la pérdida de información y compliquen la generación de reportes de procesos como liquidaciones de camarón. Como resultado de esto, tenemos que la información no está disponible de manera inmediata, teniendo como consecuencia demoras significativas mientras los supervisores tratan de cuadrar cuentas provocando también que la trazabilidad de los lotes se ve limitada.

El presente proyecto se presenta como una solución directa a las problemáticas planteadas y atiende de forma fundamental el registro de procesos de producción. El uso de WebSockets establece una conexión continua entre el servidor y los dispositivos, lo que brinda una actualización de la información en tiempo real. Esto contribuye también a una mejora notable en la comunicación y coordinación entre las distintas áreas de la empresa.

Esta propuesta usa herramientas Open Source como Node.js, Angular, Ionic y MySQL, lo permite desarrollar un sistema robusto, escalable y económico, alineándose con los principios de sostenibilidad tecnológica. El despliegue del

proyecto se realizará en un entorno Cloud (Clever Cloud), lo que permite la disponibilidad continua y protección de la información bajo normativas europeas de ciberseguridad y privacidad.

Con este nuevo sistema, PROMAROSA podrá digitalizar sus procesos de producción, mejorar la forma en la que registra la información, prevenir errores comunes sobre el manejo manual y a reducir considerablemente tiempos de generación de reportes. Al tener todos los datos centralizados, se facilitará la generación sistematizada de reportes estadísticos, como los tiempos de procesamiento, liquidación, y productividad por área o lote. Además, estos reportes podrán visualizarse mediante gráficos dinámicos, lo que contribuirá a una mejora operativa continua y sostenible a la empresa.

El tema propuesto está alineado a los objetivos del Plan de Desarrollo para el nuevo Ecuador, específicamente los siguientes ejes:

Eje1.- Desarrollo Económico.

Objetivo 5.- Fomentar de manera sustentable la producción mejorando los niveles de productividad [14].

Política 5.3.- Impulsar la productividad, desarrollo y la diversificación en la actividad pesquera y acuícola, promoviendo la adopción de tecnologías modernas y ambientales sostenibles [14].

1.5. ALCANCE DEL PROYECTO

El presente proyecto abarca el diseño, desarrollo e implementación de un sistema integral para el registro en tiempo real de los procesos de producción de camarón en PROMAROSA. Para lograrlo, se emplearán tecnologías basadas en el protocolo WebSocket y herramientas Open Source. El sistema tendrá una estructura de arquitectura híbrida, compuesta por una aplicación móvil para la captura de datos, una aplicación web para consulta y análisis, y generación de órdenes de producción, con una API encargada de gestionar la comunicación, almacenamiento y trazabilidad de toda la información.

El sistema permitirá reemplazar los procesos manuales que utiliza actualmente la empresa, permitiendo mejorar el registro de los procesos de producción y su trazabilidad — historial—, la comunicación entre áreas y generación de reportes —incluida la liquidación—. Las etapas que se digitarán son: recepción, control de calidad, clasificación y procesamiento, y masterizado, las principales del proceso de producción. Cada etapa contará con su propio módulo, pero todos se conectarán entre sí para que la información se actualice al instante gracias al uso de WebSockets.

El sistema inicia con un **Módulo de autenticación** que regula el acceso a los supervisores mediante roles, asignando permisos acordes al tipo de usuario. Debido al contexto de la empresa, la autenticación se realiza de forma nativa en la aplicación con una base de datos local. A partir de este punto, el proyecto se volverá secuencial y cada módulo dependerá de los datos registrados en el módulo anterior.

El **Módulo de Recepción de Materia Prima** permite registrar la información inicial de cada lote que llega a planta, estableciendo la base para el seguimiento y uso de su información en los módulos posteriores. Paralelamente, el **Módulo de Control de Calidad** gestiona la evaluación del producto, esto basados en el análisis de Materia Prima y sus defectos, asociándolo a su respectivo lote.

El **Módulo de Clasificación y Procesamiento** es el más pesado y complejo de todos. Éste continúa el flujo de producción, aquí es donde se utiliza la información de los módulos anteriores para tomar un lote con control de calidad y defectos para procesarlo y empacarlo en lo que llamamos *Ingreso de Túnel*. Este componente integra cálculos automáticos necesarios para este proceso, tales como rendimiento, sobrantes y total empacado, reemplazando una tarea que tradicionalmente demandaba un tiempo considerable y era propenso a errores.

Por otro lado, el **Módulo de Masterizado** presenta una menor complejidad, al menos para este trabajo. Este módulo es una continuación del de clasificación y se consolida como el producto final en el proceso de producción. Finalmente se incorpora un **Módulo de Reportes y Estadísticas**, encargado de generar reportes básicos de los procesos de producción, proporcionando también el historial de un lote —*Trazabilidad*—.

Todos estos módulos funcionan sobre una única base de datos relacional — *PROMAROSA_BD*— diseñada para mantener la integridad y eficiencia en cada consulta generada, y se comunica directamente con la **API RESTful** que centraliza todos los procesos de la empresa. La API RESTful se apoya en la comunicación bidireccional gracias al uso de WebSockets, garantizando la actualización instantánea de los datos desde cualquier área.

1.6. METODOLOGÍA DEL PROYECTO

1.6.1. Metodología de Investigación

El diseño de esta investigación es No Experimental y de tipo Transversal, ya que se observará y analizará la problemática en un momento determinado, sin manipular las variables directamente [15].

En cuanto a su alcance, esta investigación es de carácter Descriptivo, dado que busca detallar los procesos actuales de producción en PROMAROSA, identificar sus debilidades y proponer una solución basada en herramientas tecnológicas que permitan mejorar el control y registro de los procesos productivos [16].

1.6.2. Beneficiarios del Proyecto

Los beneficiarios son los trabajadores de PROMAROSA involucrados directamente en los procesos de Recepción, Calidad, Clasificación, Masterizado y Supervisión en general. Si el número de personas fuera mayor, se hubiera necesitado hacer un muestreo no probabilístico intencional [17], pero en este caso la población está conformada por 16 supervisores distribuidos de la siguiente forma:

BENEFICIARIOS	
ÁREA	NÚMERO DE PERSONAS
Recepción	2
Clasificación	3
Calidad	4
Masterizado	3
Gerencia	4
TOTAL	16

Tabla 1: Número de beneficiarios

1.6.3. Variables

La implementación de un sistema de control y registro en tiempo real en PROMAROSA, mediante el uso de WebSockets y herramientas Open Source, permitirá optimizar la eficiencia operativa en los procesos productivos de la empacadora. Este sistema digital estará presente en cada una de las etapas clave del proceso (recepción, clasificación, calidad y masterizado), reemplazando el registro manual con formularios digitales interconectados. Gracias a la comunicación en tiempo real, los datos serán transmitidos y sincronizados automáticamente, reduciendo los tiempos muertos y mejorando la coordinación entre áreas.

1.6.4. Técnicas de Recolección de Información

Esta investigación está dentro del enfoque cuantitativo, ya que su objetivo principal es implementar un sistema tecnológico orientado a mejorar los procesos de control y registro en tiempo real en PROMAROSA [18]. Aunque se recolectarán datos cualitativos a través de entrevistas y observaciones, estos se utilizarán únicamente como contexto para entender mejor la problemática y no influirán directamente en el diseño del sistema. En cuanto al método de investigación, se empleará el método deductivo, dado que parte de principios generales (como las deficiencias observadas en sistemas manuales y las ventajas comprobadas del uso de tecnologías como WebSockets) para aplicarlos en un caso específico [19]. Este método permite establecer una lógica clara y ordenada en la construcción de la solución tecnológica, siguiendo un razonamiento basado en evidencias previas y objetivos definidos.

1.6.5. Análisis de Recolección de Datos

La recolección de datos se realizará por medio de entrevistas a los supervisores de PROMAROSA con el objetivo de identificar problemas y necesidades relacionadas con los procesos actuales de producción. Esto se complementará con la observación directa en las áreas de recepción, clasificación y producción —que son las áreas más importantes—, lo cual permitirá registrar el desarrollo real de las actividades y detectar cualquier ineficiencia. También se realizará un análisis de los registros existentes en cuadernos y hojas de cálculo, para entender mejor la estructura y flujo de la información.

Después del análisis, se realizarán las pruebas necesarias de cada usuario y rol, donde el personal podrá hacer uso del sistema en condiciones reales de trabajo para evaluar su funcionamiento y facilidad de uso. Finalmente, se implementará la captura automática de datos en tiempo real mediante el uso de formularios digitales conectados a través de WebSockets. Esto permite generar registros precisos y actualizados, totalmente alineados con la operación diaria de la planta.

Área	Problema Detectado	Impacto	Frecuencia estimada
Recepción	Registro manual lento y propenso a errores	Demora en actualización de lotes	Alta
Calidad	Parámetros incompletos o ingresados fuera de tiempo	Fallas en trazabilidad	Media
Clasificación	Confusión entre tallas, presentaciones y coches	Errores en producción	Alta
Masterizado	Consolidación tardía de datos	Retrasos en empaques finales	Media
Gerencia	Reportes tardíos (30-60 min)	Decisiones imprecisas	Alta

Tabla 2: Problemas detectados por área (Hallazgos de entrevistas y observación)

1.7. METODOLOGÍA DE DESARROLLO

El desarrollo del proyecto se enmarca en una metodología tradicional incremental, que consiste en construir el sistema por etapas funcionales (módulos), entregando versiones operativas de manera progresiva [20]. Cada módulo se desarrolla y valida antes de pasar al siguiente, permitiendo incorporar mejoras en cada iteración y cada incremento incluye análisis, diseño, desarrollo, pruebas y validación de uno o varios módulos específicos, permitiendo que partes del sistema estén operativas mientras se continúa desarrollando el resto. Adicional a los módulos, la metodología usada permite implementar el sistema por fases.

1.7.1. Fases de Desarrollo Incremental

1.7.1.1. Inicio y análisis de requerimientos

Esta fase se centra en entender a fondo cómo funciona la producción de camarón en PROMAROSA y cuáles son las necesidades reales del proceso. Se recopila información mediante entrevista con los supervisores, observación directa

en campo y análisis de los registros existentes. Esto permite identificar los pasos de cada área, los puntos donde el registro manual de datos es más complicado y la falta de trazabilidad en tiempo real. Toda esa información sirve para definir qué debe hacer el sistema y establecer objetivos claros, como reducir errores, agilizar los registros y mejorar la coordinación entre áreas.

1.7.1.2. Diseño del modelo de datos y arquitectura

En esta etapa se organiza cómo se guardará y administrará la información del sistema diseñando la estructura de la base de datos y planificando cómo se conectarán entre sí las diferentes partes del sistema. Aquí se incluye la actualización instantánea de los datos y el diseño busca que cada módulo —recepción, control de calidad, clasificación, masterizado, y gerencia— funcionen de manera independiente, pero se integre perfectamente con el resto, asegurando que toda la información esté disponible y sea confiable.

1.7.1.3. Desarrollo incremental de módulos, organizados según su funcionalidad y dependencia

El sistema se desarrolla en partes o módulos, cada uno dedicado a una función específica. Cada módulo se desarrolla y somete a varias pruebas permitiendo realizar ajustes necesarios para su correcto funcionamiento. Primero se establecen las reglas de acceso según el rol de cada supervisor. Luego se construyen los módulos de:

- ✓ Autenticación y Control de Acceso.
- ✓ Recepción de Materia Prima.
- ✓ Control de Calidad.
- ✓ Clasificación y Procesamiento.
- ✓ Masterizado.
- ✓ Trazabilidad y Control de Lotes.
- ✓ Reportes básicos y Estadísticas.

1.7.1.4. Implementación

Una vez que los módulos estén listos, el sistema se pone en funcionamiento dentro de la empresa. Esto incluye la instalación de la aplicación, revisión previa de la base

de datos y pruebas de funcionamiento de la API RESTful de su servidor en línea. Durante esta fase se observa el comportamiento del sistema en situaciones reales de trabajo, asegurando que la información se sincronice correctamente y que todas las funciones de éste operen sin problemas.

1.7.1.5. Pruebas

El sistema atraviesa diferentes tipos de pruebas para asegurarse de que todo funcione correctamente. Se revisa que los registros de lotes, control de calidad y defectos, ingresos de túnel, liquidaciones y reportes se realicen sin errores y en tiempo real. Además, se evalúa la rapidez de las aplicaciones y la claridad de la información, garantizando que el sistema cumpla con los objetivos de eficiencia y precisión establecidos al inicio.

1.7.1.6. Evaluación y Revisión

En esta etapa, se analiza el funcionamiento general del sistema y se comparan los resultados con los objetivos definidos al inicio. Se detectan oportunidades de mejoras en la organización de la información, tiempos de respuesta y la experiencia de los usuarios, tomando en cuenta los comentarios y recomendaciones. Aunque concluya el ciclo de desarrollo, se deja abierta la puerta a una posible ampliación o ajustes futuros, asegurando que el sistema pueda crecer y adaptarse a nuevas necesidades operativas de PROMAROSA.

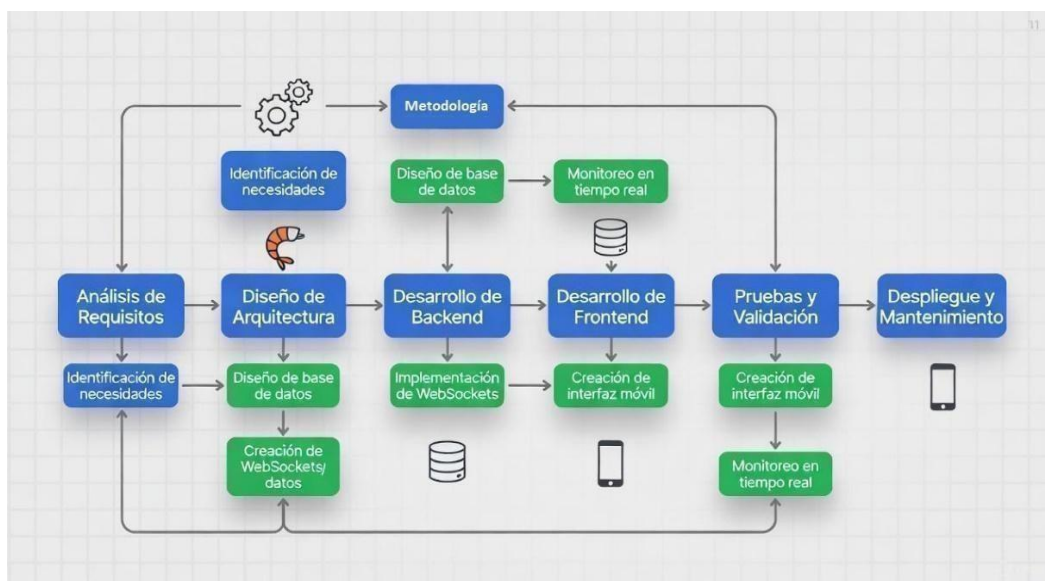


Figura 2: Proceso de Metodología

CAPÍTULO II

2. PROPUESTA

2.1. MARCO CONTEXTUAL

2.1.1. PROMAROSA

Productos del Mar Santa Rosa (PROMAROSA), ubicada en Chanduy, Santa Elena, Ecuador, es una empacadora de camarón que procesa y comercializa productos frescos y congelados para mercados nacionales e internacionales. Con una capacidad de procesamiento de hasta 50,000 libras diarias, PROMAROSA opera con 16 supervisores y 50 trabajadores en áreas de recepción, calidad, clasificación y masterizado. Los procesos actuales dependen de registros manuales en papel, lo que genera demoras (reportes en 30-60 minutos), errores humanos (aproximadamente 5% de lotes con datos inconsistentes) y falta de trazabilidad en tiempo real. Este sistema busca digitalizar los procesos, reduciendo el tiempo de reportes a 5-10 minutos, mejorando la coordinación y asegurando la trazabilidad de lotes.

2.1.1.1. Ubicación

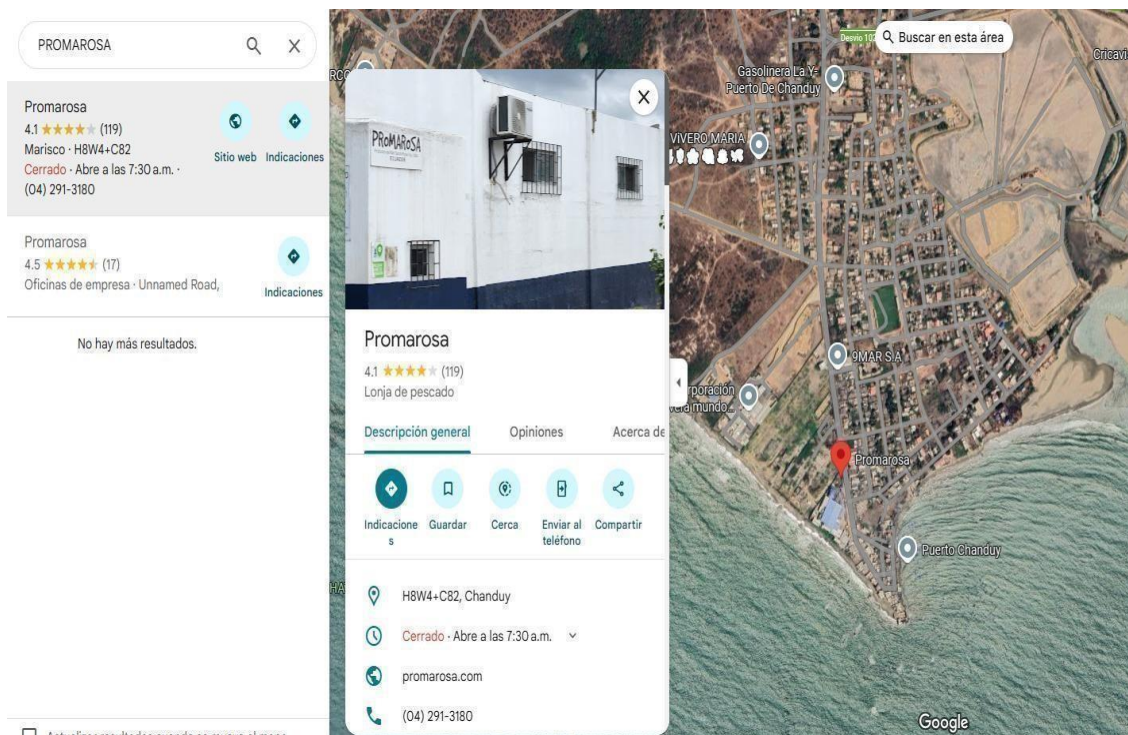


Figura 3: Vista del mapa de matriz PROMAROSA

2.1.2. Justificación Tecnológica

La adopción de herramientas Open Source (Node.js, Angular, Ionic, MySQL) reduce costos de desarrollo y licenciamiento, mientras que el despliegue en Clever Cloud garantiza escalabilidad y cumplimiento con normativas de ciberseguridad (GDPR). WebSockets permiten sincronización en tiempo real, esencial para coordinar las áreas de producción con alta carga de datos (10-20 lotes diarios). Esta tecnología supera limitaciones de protocolos tradicionales como HTTP, reduciendo latencia y habilitando actualizaciones bidireccionales, lo que optimiza la eficiencia operativa al proporcionar acceso rápido a datos actualizados y facilitar la generación de informes detallados.

2.1.3. Impacto Esperado

- ✓ Disminución en los tiempos necesarios para generar liquidaciones, ya que la información se actualiza de manera inmediata y genera reportes de forma automática.
- ✓ Mayor claridad en la trazabilidad o seguimiento de cada lote, permitiendo revisar su recorrido paso a paso en cada etapa de su proceso de producción.
- ✓ Disminución de errores ocasionados por el registro manual, esto gracias al uso de formularios digitales que rempazan la transcripción manual de datos.

2.1.4. Base Legal

2.1.4.1. Constitución de la República del Ecuador (2008)

Artículo 66.- Garantiza el derecho a la protección de los datos personales, lo que incluye el poder acceder a ellos, decidir sobre su uso y exigir su adecuada protección [21]. Cualquier recolección, almacenamiento, tratamiento, distribución o difusión de estos datos requiere necesariamente el consentimiento expreso del titular o una autorización establecida por la ley [22].

2.1.4.2. Ley Orgánica de Protección de Datos Personales (Ecuador, 2021)

Artículo 37.- El responsable del tratamiento de datos personales está a cumplir con el principio de seguridad, implementando medidas técnicas, organizativas y de cualquier otra naturaleza que sean adecuadas al volumen y categorías de los datos, al estado actual de la tecnología, así como la probabilidad y gravedad de los riesgos

identificados. Además, deberá establecer un proceso permanente y continuo de verificación, evaluación y valoración de la eficiencia, eficacia y efectividad de todas las medidas implementadas, con el propósito de garantizar y mejorar constantemente la seguridad en el tratamiento de los datos personales [23].

2.1.4.3. Reglamento General de Protección de Datos (GDPR, UE)

Para cumplir con el Reglamento General de Protección de Datos (GDPR) de la Unión Europea —aplicable debido al despliegue del sistema en Clever Cloud—, se garantiza la confidencialidad de la información mediante el uso de accesos estrictamente controlados y limitados al personal autorizado [24]. Con el objetivo de prevenir infracciones al Reglamento y proteger los datos personales, se está obligado a evaluar de forma previa y continua los riesgos e implementar medidas de mitigación adecuadas asegurando en todo momento un nivel de seguridad apropiado al riesgo, que incluya la confidencialidad, considerando el estado actual de la tecnología, los costos razonables de aplicación y la naturaleza específica de los datos que se deben proteger [25].

2.1.4.4. Código Orgánico Integral Penal (Sección Sexta)

Artículo 178.- Quien, sin consentimiento del titular ni autorización legal, acceda, intercepte, examine, retenga, grabe, reproduzca, difunda o publique datos personales, mensajes, voz, audio, vídeo, correspondencia o comunicaciones privadas de otra persona por cualquier medio, será castigado con prisión de uno a tres años [26]. Esta sanción no aplica si la persona divulga grabaciones en las que participa personalmente ni cuando se trata de información pública según la ley.

Artículo 179.- La persona que, por razón de su profesión, oficio, empleo o arte, tenga conocimiento de un secreto que pueda causar daño a otro y lo revele sin autorización, será sancionada con prisión privativa de seis meses a un año [26].

Artículo 180.- La persona que divulgue información de acceso restringida será castigado con prisión de uno a tres años. Se considera información de circulación restringida aquella protegida por cláusula de reserva legal, la producida por la Fiscalía durante una investigación previa, o la que afecte los derechos de niñas, niños y adolescentes según el Código Orgánico de la Niñez y Adolescencia [26].

2.2. MARCO CONCEPTUAL

2.2.1. Industria Camaronera Ecuatoriana: Líder Mundial en Producción

La industria camaronera ecuatoriana es un pilar fundamental de la economía nacional, destacándose como uno de los principales exportadores mundiales de camarón hacia mercados como Estados Unidos, China y la Unión Europea [27]. Este sector genera más de 260.000 empleos directos e indirectos, beneficiando a numerosas familias, y debe su éxito a condiciones naturales favorables, innovación tecnológica y estrictos estándares de sostenibilidad y calidad [28].

El mercado de procesamiento de camarones está influenciado por varios factores importantes, como el creciente consumo mundial de mariscos, la creciente demanda de mariscos listos para consumir y convenientes y el aumento del comercio mundial [29]. A pesar de desafíos como enfermedades y regulaciones internacionales, la constante inversión en investigación y capacitación mantiene a Ecuador como líder competitivo, consolidando la trazabilidad como elemento clave para su posicionamiento estratégico en el mercado mundial [30].

2.2.2. Fases del Procesamiento de Camarón

El camarón congelado llega a planta desde las granjas camaroneras y es transportado en barco o en camiones de carga, y es descargado en bines o gavetas dentro de la planta con el fin de obtener productos terminados congelados [31]. El procesamiento de camarón sigue estrictos controles de seguridad sanitaria. Todo el personal que participa activa y directamente de los procesos de producción debe cumplir estrictamente las normas de bioseguridad para asegurar la calidad del producto [32].

2.2.2.1. Recepción de Materia Prima

La Materia Prima o camarón fresco —pelágico en ciertas plantas— se recibe en la planta procesadora de camarón. Atraviesa el primer filtro que consiste en verificar la documentación del seguimiento del producto, y éste después de ser receptado se somete a un riguroso control de calidad con parámetros que varían según las exigencias de los clientes —como tamaño, color, sabor, corte—, antes de su admisión y procesamiento [32].

2.2.2.2. Control de Calidad

El término “calidad” aplicado a un producto puede no tener el mismo significado para todo el mundo [33]. Por lo general, llamamos “Control de Calidad” a la revisión constante de las propiedades organolépticas, microbiológicas y fisicoquímicas del camarón durante su recepción, procesamiento y empaque [34]. También podríamos definirlo como el análisis y evaluación de las muestras congeladas como descongeladas, en estado crudo, peladas y desvenadas, cocidas, siguiendo un procedimiento estándar de las condiciones del producto: glaseado, presencia de camarones rojos —o naranjas— rotos o con defectos como patas desprendidas, materias extrañas, además de su clasificación por tamaños [33].

2.2.2.3. Clasificación y Procesamiento

Una vez finalizada la fase de revisión y limpieza de los camarones el producto —camarón entero o cola— se coloca con la ayuda de un carro monta carguero en las tolvas de las máquinas clasificadoras, con abundante hielo [32]. Todo el camarón, sea entero o sea cola, pasa por las máquinas clasificadoras para determinar las tallas correspondientes según especificaciones indicadas [35]. Todo producto que no cumpla con las exigencias del cliente —el mercado en general— para ser empacado como camarón entero, tendrá que reingresar a la máquina clasificadora y posterior línea de producción para su procesamiento como camarón-cola [32].

2.2.2.4. Masterizado

Llamado también proceso de Congelación y Almacenamiento —o Congelación y Empaquetado—, aquí los camarones empacados permanecen en túneles, cuartos o cámaras de congelación un tiempo aproximado de 18 horas, a una temperatura de entre -18°C y -20°C , para obtener al final una congelación adecuada y efectiva [32]. Existen empaques como: IWP, IVP, bulk, fundas al retail y sacos de 20 kg [7]. El producto terminado se empaca en cajas máster y es paletizado siguiendo parámetros establecidos por la empacadora —siguiendo órdenes de clientes— y luego se traslada a la bodega de congelación, donde se mantiene a una temperatura de -18°C , hasta que deba ser despachado y embarcado para su comercialización [32].

2.2.3. Trazabilidad en la Industria Acuícola

La Comisión del Codex Alimentarius define la trazabilidad como la capacidad de rastrear el recorrido de un alimento a lo largo de etapas específicas de su producción, procesamiento y distribución, lo que permite conocer la identidad, el historial y el origen de un producto o de sus ingredientes, así como su destino final una vez distribuido [36] [37]. En consecuencia, los sistemas de trazabilidad funcionan como herramientas esenciales para la gestión eficaz de la información alimentaria.

2.2.4. Aplicaciones móviles vs. Aplicaciones de software

Las aplicaciones móviles están creadas especialmente para para funcionar en dispositivos portátiles, aprovechando características como su movilidad, capacidad de conexión y los sensores integrados en ellos [38]. Es bastante común pensar que cuando hablamos de aplicaciones web y de aplicaciones móviles nos estamos refiriendo al mismo concepto, pero no es el caso [39]. Una aplicación de software es, por lo general, un programa más complejo, diseñado principalmente para ser utilizado en ordenadores [38].

2.2.4.1. Aplicaciones móviles

Es una aplicación de software creada para funcionar en dispositivos móviles, como teléfonos inteligentes o tabletas y su instalación es un programa que se descarga de una web o de una tienda de aplicaciones y se instala en el dispositivo móvil del usuario [39]. Estas aplicaciones o programas utilizan los recursos del dispositivo para ofrecer distintas funcionalidades, además algunas requieren una conexión a internet para funcionar correctamente, mientras que otras pueden funcionar sin conexión [38].

2.2.4.2. Aplicaciones web

Es una herramienta que está alojada en un servidor y que los usuarios pueden acceder desde la red mediante un navegador, además se podría decir que es una página web que se ha adaptado a los formatos de programación de smartphones o tablets [39]. Las aplicaciones de software ofrecen funciones más robustas y

complejas, perfectamente adecuadas para la realización de tareas más intensivas [38].

2.2.5. Beneficios de Herramientas Open Source en el Desarrollo de Software

El software de código abierto —Open Source— ofrece varias ventajas. Al ser desarrollado por una comunidad activa, su código es confiable y de alta calidad [40]. Además, al poder acceder libremente al código fuente, es posible adaptarlo a las necesidades específicas de cada proyecto sin depender de un solo proveedor [41].

El Código Abierto ayuda a reducir costos por su uso gratuito, elimina la dependencia de una sola empresa y se mantiene actualizado constantemente por una comunidad activa de programadores [42]. También ofrece mayor seguridad mediante estándares abiertos y la rápida incorporación de parches, y fomenta una colaboración global que aporta soporte más allá de intereses netamente comerciales [43].

2.2.6. Herramientas

2.2.6.1. Entorno de desarrollo

2.2.6.1.1. Visual Studio Code

Es un editor de código gratuito, de código abierto y que funciona en diferentes sistemas operativos y fue desarrollado por Microsoft, una empresa con amplia experiencia en crear entornos de desarrollo [44]. Gracias a esto, ofrece una herramienta ligera, practica y muy popular entre la comunidad de programadores. Una gran ventaja de la herramienta es una amplia personalización con extensiones para una mejor integración y experiencia [45].

2.2.6.2. Entorno de ejecución

2.2.6.2.1. Node.js

Entorno de ejecución de JavaScript multiplataforma que permite crear servidores, aplicaciones web, herramientas de línea de comando y scripts, además proporciona un conjunto de primitivas de E/S asíncronas en su biblioteca estándar que evitan que el código JavaScript se bloquee y, en general, las bibliotecas en Node.js se escriben utilizando paradigmas que no bloquean, lo que hace que el

comportamiento de bloqueo sea la excepción en lugar de la norma [46]. Con lo popular que es Node.js y Express.js, los desarrolladores pueden seguir el principio “de separación de preocupaciones” durante el desarrollo [47].

2.2.6.3. Lenguaje de programación

2.2.6.3.1. JavaScript

Aunque es ampliamente reconocido como un lenguaje de scripting —secuencias de comandos— para páginas web, JavaScript también se utiliza en numerosos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat. JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo y dinámico, con soporte para programación orientada a objetos, imperativa y declarativa —por ejemplo, programación funcional— [48].

2.2.6.4. Gestor de base de datos

2.2.6.4.1. MySQL Workbench

Es una herramienta visual *todo-en-uno* para arquitectos de bases de datos, desarrolladores y administradores de base de datos. MySQL Workbench ofrece funciones para el modelado de datos, desarrollo SQL y herramientas de administración integrales para configuración de servidores, gestión de usuarios, copias de seguridad y mucho más; MySQL Workbench está disponible en Windows, Linux y Mac OS X [49].

2.2.6.4.2. phpMyAdmin

Es una herramienta de software libre escrita en PHP, destinado a manejar la administración de MySQL en la Web, además admite una amplia gama de operaciones en MySQL y MariaDB; las operaciones de uso frecuente (gestión de bases de datos, tablas, las columnas, relaciones, índices, usuarios, permisos) se pueden realizar a través de interfaz de usuario, mientras que todavía tiene la capacidad de ejecutar directamente cualquier instrucción SQL [50].

2.2.6.5. Servidor

2.2.6.5.1. Clever Cloud

Plataforma en la nube que permite la implementación de aplicaciones y servicios digitales garantizando la confidencialidad y el control de los datos. Clever Cloud

permite a sus clientes beneficiarse de la legislación europea más estricta y protectora para los usuarios de servicios en la nube, incluido el Reglamento General de Protección de Datos (GDPR), la Ley de Nube, la Ley de Servicios Digitales (DSA), la Ley de Mercados Digitales (DMA) o las normas y regulaciones de la UE relacionadas con la ciberseguridad y la ética en digital [51].

2.2.6.5.2. GitHub

GitHub es una plataforma en la nube que utiliza Git, un sistema de control de versiones, para que los desarrolladores puedan colaborar juntos en proyectos compartidos [52]. Con cada Git, cada persona puede trabajar en su propia copia de archivos, llamada rama, donde hacen sus modificaciones sin afectar el trabajo de los demás y cuando los cambios están listos, Git combina de manera ordenadas las modificaciones de cada persona en la versión final del proyecto [53].

2.2.6.5.3. Render

Render es una plataforma de servicios en la nube (PaaS - Platform as a Service) que simplifica el despliegue y alojamiento de aplicaciones, bases de datos, sitios estáticos y otros servicios web [54]. Gracias a su versatilidad, brinda todos los recursos necesarios para desarrollar tus proyectos. Su capacidad de adaptación y eficiencia la convierten en una opción ideal para cualquier tipo de hospedaje web o de aplicaciones [55].

2.2.6.6. Entorno local de desarrollo

2.2.6.6.1. XAMPP

Es un paquete de software que incluye Apache, MariaDB, PHP y Perl, lo que proporciona todos los componentes necesarios para desarrollar aplicaciones web con total garantía [56]. XAMPP es el servidor más utilizado y popular entre los desarrolladores web y programadores, porque permite instalar y configurar el entorno de un servidor local de manera sencilla [57].

2.2.6.7. Framework

2.2.6.7.1. Angular

Siendo una plataforma capaz de adaptarse y escalar desde proyectos individuales hasta aplicaciones para entornos empresariales, Angular es un Framework de

JavaScript de código abierto escrito en TypeScript donde su objetivo principal es desarrollar aplicaciones de una sola página; Google se encarga del mantenimiento y constantes actualizaciones de mejoras para este Framework [58].

2.2.6.7.2. Ionic

Framework diseñado para funcionar rápidamente en todos los últimos dispositivos móviles que permite crear aplicaciones con prácticas integradas. Ionic se centra en la interacción frontend UX y UI de una aplicación — UI controles, interacciones, gestos, animaciones; además es fácil de aprender y se integra con otras bibliotecas o marcos, como Angular, React, o Vue [59].

2.2.6.8. API

2.2.6.8.1. API RESTful

Es un estilo arquitectónico —interfaz— que dos sistemas de computación utilizan para comunicación entre sistemas de manera segura a través de Internet. Los sistemas que utilizan API REST pueden escalar de una forma más eficiente porque REST optimiza la comunicación entre el cliente y el servidor; esta tecnología sin estado elimina la carga del servidor porque este no debe retener la información de solicitudes anteriores del cliente [60].

2.2.6.8.2. WebSockets

Es un protocolo que consta de dos categorías: la primera parte es el protocolo de enlace, que contiene la respuesta del servidor y el mensaje del cliente; la segunda parte es la transferencia de datos también proporciona un socket inherente al navegador, lo que elimina la mayoría de los problemas con el uso de HTTP, creando así una aplicación escalable en tiempo real [10]. Los WebSockets proporcionan un canal de comunicación permanente y bidireccional — full-duplex— entre el cliente y el servidor, lo que permite una transferencia de datos fluida en tiempo real con una sobrecarga mínima [61].

2.2.6.9. Herramientas de prueba

2.2.6.9.1. Postman

Herramienta diseñada para facilitar el desarrollo y colaboración entre programadores, permitiéndoles verificar y probar el comportamiento de servicios

web y aplicaciones por medio de una interfaz gráfica sencilla e intuitiva que posibilita el envío de solicitudes a servidores web y la recepción de sus respuestas [62]. Postman cuenta con los métodos GET, POST, PUT, PATCH y DELETE que permiten tomar acción ante nuestras peticiones [63].

Este proyecto contribuye a la línea de investigación de desarrollo de software, la cual se relaciona con la sub - línea de investigación TSI en las organizaciones y en la sociedad [64].

2.3. MARCO TEÓRICO

2.3.1. Importancia de la Trazabilidad en la Cadena Productiva

La trazabilidad nos permite seguir el rastro de un producto desde el origen o fabricación hasta su destino final, y no solo podemos ver todos los procesos de manufactura involucrados en su producción, sino también todos los procesos administrativos que acompañan al producto durante su ciclo [65]. En el ámbito alimentario, para comprender la Trazabilidad de un producto a lo largo de su cadena de suministro o logística, el concepto como tal se divide Trazabilidad Interna donde se obtiene la traza de lo que va dejado un producto por todos los procesos internos de una empresa, y Trazabilidad Externa donde se externaliza los datos de la traza interna y se añaden indicios adicionales necesarios [66].

2.3.2. Eficiencia Técnica con Arquitectura Cliente-Servidor

La arquitectura Cliente-Servidor es uno de los modelos distribuidos más conocidos, el cual está conformado por dos partes principales: el servidor (proveedor) y el cliente (consumidor) [67]. Una de las ventajas más evidentes de este enfoque es la posibilidad de emplear equipos mucho más económicos comparados con los que exigen una infraestructura centralizada y basada en sistema de gran escala. Además, permite la integración de componentes de hardware y software de diferentes fabricantes, lo que no solo reduce significativamente los costos, sino que también facilita la flexibilidad al momento de implementar y actualizar las soluciones tecnológicas [68].

Una arquitectura Cliente-Servidor incrementa el poder de procesamiento al separar el manejador de la base de datos de la aplicación [69]. Este modelo resulta adecuado para el desarrollo de aplicaciones móviles ya que facilita una interacción y comunicación sencilla entre los componentes: los clientes que envían solicitudes y el servidor que responde a ellas [70].

2.3.3. Integración Tecnológica para la Optimización Productiva

La tecnología se relaciona directamente con la productividad, estableciéndose de esta manera que la variable independiente es la tecnología y la variable dependiente es la productividad [71]. La optimización de procesos administrativos para la empresa se presenta cuando se habla de la automatización o sistematización de tareas de bajo valor agregado, la inferencia suele ser tareas administrativas, ingreso manual de datos y trabajo de verificación tedioso [72]. Como la eficiencia de una empresa depende de sus procesos, su optimización es fundamental para alcanzar la competitividad esperada y convertirse en una referencia en el mercado [73].

2.4. REQUERIMIENTOS

2.4.1. Requerimientos Funcionales

Autenticación y Seguridad	
Código	Descripción
RF-1	Permitir acceso seguro mediante usuario y contraseña, y validación de roles (recepción, calidad, clasificación, masterizado y gerencia).
RF-2	Generar credenciales por administrador y restringir accesos según rol.
RF-3	Registrar logs de accesos en tabla de auditoría para trazabilidad.

Tabla 3: Requerimientos Funcionales - Autenticación y Seguridad

Módulos Móviles (Recepción, Calidad, Clasificación, Masterizado)	
Código	Descripción
RF-4	Registrar datos de recepción: proveedor, lote, estado, libras, con sincronización en tiempo real.

Módulos Móviles (Recepción, Calidad, Clasificación, Masterizado)	
Código	Descripción
RF-4	Registrar datos de recepción: proveedor, lote, estado, libras, con sincronización en tiempo real.
RF-5	Evaluar calidad: uniformidad, olor, defectos.
RF-6	Procesar clasificación: talla, corte, pesos, actualizando estado del lote constantemente.
RF-7	Verificar masterizado: calidad final y empaque, contribuyendo a inventario con alta precisión.

Tabla 4: Requerimientos Funcionales - Módulos Móviles

Módulos Web (Trazabilidad y Reportes)	
Código	Descripción
RF-8	Consultar historial de lotes desde recepción hasta masterizado, mostrando recursos y procesos.
RF-9	Generar reportes automáticos con tiempos de 5-10 minutos con exportación a PDF.
RF-10	Visualizar estadísticas básicas: tablas estáticas y gráficos simples para optimización operativa.

Tabla 5: Requerimientos Funcionales - Módulos Web

Integración y Sincronización	
Código	Descripción
RF-11	Implementar WebSockets para actualizaciones en tiempo real, con reconexión automática.
RF-12	Validar datos en formularios para reducir errores.
RF-13	Calcular métricas de eficiencia: reducción de tiempos de reportes y precisión en trazabilidad.

Tabla 6: Requerimientos Funcionales - Integración y Sincronización

2.4.2. Requerimientos Técnicos

Configuración Mínima para Clientes (Aplicación Móvil y Web)	
Código	Descripción
RT-1	Dispositivos Móviles: Cualquier smartphone o tablet genérico con Android 8.0+ o iOS 12.0+, procesador dual-core (e.g., Snapdragon 400 o equivalente), 2 GB RAM, y pantalla de al menos 4 pulgadas para visualización óptima de formularios.
RT-2	2 Ordenadores de Oficina: Cualquier PC o laptop genérica con Windows 10+, macOS 10.13+, o Linux (Ubuntu 18.04+), procesador Intel Core i3 o equivalente, 4 GB RAM, y navegador moderno (Chrome, Firefox, Edge) para acceso a la app web.
RT-3	Conectividad: Acceso a internet estable de al menos 10 Mbps para sincronización en tiempo real vía WebSockets; compatible con redes Wi-Fi o datos móviles estándar en entornos de oficina.
RT-4	Almacenamiento Local: Mínimo 100 MB libres en el dispositivo para caché de datos y app móvil (Ionic), sin requerimientos intensivos ya que la mayoría de las operaciones son en nube.

Tabla 7: Requerimientos Técnicos - Configuración Mínima para Clientes

Configuración Recomendada para Clientes (Aplicación Móvil y Web)	
Código	Descripción
RT-5	Dispositivos Móviles: Smartphone o tablet con Android 11+ o iOS 15+, procesador quad-core (e.g., Snapdragon 600+), 4 GB RAM, y pantalla HD+ para mejor rendimiento en recolección de datos en campo.
RT-6	Ordenadores de Oficina: PC o laptop con Windows 11+, macOS 12+, o Linux reciente, procesador Intel Core i5 o equivalente, 8 GB RAM, y navegador con soporte para WebSockets (e.g., Chrome 100+) para visualización fluida de reportes.
RT-7	Conectividad: 50 Mbps+ para actualizaciones bidireccionales rápidas; recomendado en oficinas con redes seguras para manejar 10-20 lotes diarios sin interrupciones.
RT-8	Almacenamiento Local: 500 MB+ para optimizar rendimiento offline limitado (en revisión), asegurando compatibilidad con dispositivos genéricos sin hardware especializado.

Tabla 8: Requerimientos Técnicos - Configuración Recomendada para Clientes

Servidor y Base de Datos (Despliegue en Nube - Configuración Predeterminada)	
Código	Descripción
RT-9	Plataforma: Clever Cloud o similar, con Node.js v18+, MySQL 8.0+, y Socket.IO para WebSockets; escalabilidad automática para hasta 50 usuarios concurrentes.
RT-10	Recursos: Instancia básica con 2 vCPU, 4 GB RAM, y 50 GB almacenamiento; maneja API RESTful y sincronización en tiempo real sin hardware local.
RT-11	Seguridad: HTTPS con SSL/TLS gestionado por el proveedor, JWT para autenticación, y cumplimiento GDPR para protección de datos.
RT-12	Rendimiento: Latencia <5 seg en sincronizaciones, con reconexión automática; ideal para entornos en nube donde no se requiere infraestructura física en sitio.

Tabla 9: Requerimientos Técnicos - Servidor y Base de Datos

Servidor y Base de Datos (Despliegue Local Opcional - Configuración Más Robusta)	
Código	Descripción
RT-13	Hardware: Servidor dedicado o PC robusta con procesador Intel Core i7 o equivalente, 16 GB RAM, 1 TB SSD, y UPS para estabilidad en entornos de oficina con posibles cortes de energía.
RT-14	Software: XAMPP o similar para entorno local (Apache/MariaDB), Node.js v18+, MySQL 8.0+; requiere configuración manual de WebSockets y backups.
RT-15	Conectividad: Red LAN de 100 Mbps+ con firewall; recomendado para escenarios sin internet constante, pero aumenta costos de mantenimiento.
RT-16	Rendimiento: Soporte para hasta 50 usuarios, con énfasis en hardware robusto para manejar cargas locales sin depender de nube, reduciendo latencia pero requiriendo IT en sitio.

Tabla 10: Requerimientos Técnicos - Servidor y Base de Datos

2.5. COMPONENTE DE LA PROPUESTA TECNOLÓGICA

Basado en las fases de desarrollo incremental descritas en la metodología, esta sección detalla los componentes tecnológicos utilizados en la implementación del sistema. Se adapta el enfoque de las tesis referenciadas, donde se enfatiza la integración de herramientas Open Source, el preprocesamiento de datos y la evaluación de modelos, aplicados aquí al contexto de control de procesos productivos en la empacadora PROMAROSA.

2.5.1. Fases del proyecto

2.5.1.1. Fase 1: Inicio y análisis de requerimientos

En esta etapa se examina de manera detallada de los procesos de producción que actualmente se llevan a cabo en PROMAROSA. El análisis se realiza mediante la observación directa en las principales áreas del proceso de producción: recepción, control de calidad, clasificación y procesamiento, y masterizado. Terminado el proceso de observación, se complementa con una entrevista a los 16 supervisores y operarios. Estos dos trabajos permiten identificar con claridad problemas frecuentes como retrasos provocados por el registro manual de datos y la limitada coordinación entre áreas, lo que sirve para comprender las necesidades del sistema.

De igual manera, se establece el alcance del proyecto, teniendo como prioridad el desarrollo de módulos relacionados la etapa más crítica de cada proceso de producción. Además, se evalúa cada requisito no funcional, como la demanda del sistema para soportar altas cargas de trabajo y escalar según sea necesario, para garantizar que la solución planteada sea sostenible y adecuada para la empresa.

Herramienta	Ventajas	Desventajas	Aplicación en Fase
Node.js	Asíncrono, escalable para real-time	Alto consumo en conexiones persistentes	Backend para API y WebSockets
MySQL	Relacional, eficiente para queries estructuradas	Menos flexible para datos no estructurados	Almacenamiento de lotes y auditoría
WebSockets	Bidireccional, baja latencia para monitoreo	Requiere manejo de conexiones	Sincronización real-time
Ionic	Híbrido para móviles, integración con Angular	Dependencia de Cordova para nativo	Apps de campo
Postman	Pruebas API intuitivas	Limitado para pruebas automatizadas	Validación inicial

Tabla 11: Comparación de Herramientas en Fase 2

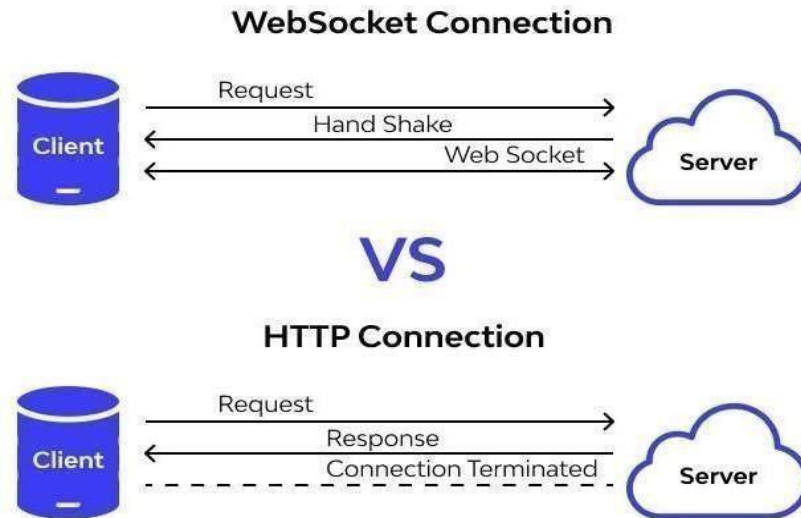


Figura 4: Conexión WebSockets [61]

2.5.1.2. Fase 2: Diseño del modelo de datos y arquitectura

2.5.1.2.1. Identificación de herramientas iniciales

En esta fase se realiza el diseño detallado de la base de datos PROMAROSA_BD, usando MySQL Workbench para crear tablas como lotes —con información necesaria para el posterior registro, que será fundamental para la trazabilidad—, calidad, clasificación y masterizado. Se asegura la integridad de los datos mediante claves foráneas e índices que permiten consultas rápidas para generación de reportes —Liquidación—.

Al mismo tiempo, se desarrolla la API RESTful con varios Endpoints necesarios que permitan registrar nuevos lotes o consultar su trazabilidad con filtros como código o número de lote. Se integra WebSockets usando Socket.IO para notificaciones en tiempo real y sincronizar la información entre varios usuarios reduciendo la latencia y manteniendo la información siempre actualizada.

Capa	Componente	Función	Herramienta Usada
Cliente	AppMóvil/Web	Interfaz para entrada/consulta de lotes	Ionic/Angular
Servidor	API RESTful	Procesamiento y validación de datos	Node.js
Comunicación	WebSockets	Sincronización bidireccional real-time	Socket.IO
Datos	Base de Datos	Almacenamiento relacional con auditoría	MySQL
Seguridad	Autenticación	Tokens JWT para roles	bcript/JWT
Pruebas	Testing API	Validación de endpoints	Postman

Tabla 12: Componentes de la Arquitectura

La **Figura 2.3** muestra el proceso básico de establecimiento de una conexión WebSocket entre un cliente y un servidor. Primero, el cliente envía una solicitud inicial mediante el protocolo HTTP utilizando el mecanismo *handshake*. Una vez aceptada, la conexión se actualiza a WebSocket, permitiendo una comunicación bidireccional y en tiempo real. A partir de ese momento, tanto el cliente como el servidor pueden intercambiar mensajes de forma continua sin necesidad de reabrir la conexión para cada intercambio.

```
1 const http = require('http');
2 const WebSocketServer = require('websocket').server;
3
4 const server = http.createServer();
5 server.listen(9898);
6
7 const wsServer = new WebSocketServer({
8   httpServer: server
9 });
10
11 wsServer.on('request', function(request) {
12   const connection = request.accept(null, request.origin);
13
14   connection.on('message', function(message) {
15     console.log('Received Message:', message.utf8Data);
16     connection.sendUTF('Hi this is WebSocket server!');
17   });
18   connection.on('close', function(reasonCode, description) {
19     console.log('Client has disconnected.');
```

Figura 5: Ejemplo de conexión WebSocket [74]

2.5.1.2.2. Preparación del Entorno de Desarrollo

Para la preparación del entorno se instala Node.js (versión 18 o superior), junto con npm para la gestión de dependencias como express para la API, socket.io para la comunicación en tiempo real y mysql2 para la conexión con la base de datos. En el entorno local, XAMPP se utiliza para realizar pruebas iniciales mediante Apache, mientras que en la nube la plataforma Clever Cloud se encarga del escalado automático del servicio. Además, se habilita GitHub como plataforma para el control de versiones, organizando el trabajo mediante ramas —como dev— destinadas a gestionar las distintas fases del desarrollo incremental. Para el desarrollo de la aplicación, se considera el uso Ionic/Angular para ofrecer una interfaz de usuario responsiva, incorporando también Capacitor para el acceso a funcionalidades del dispositivo.

La **Figura 2.5** ilustra la estructura inicial de archivos requerida para implementar la API. En ella se muestran los componentes fundamentales, como los controladores,

modelos, rutas y configuraciones básicas. Esta organización permite establecer una base ordenada para el desarrollo, facilitando la separación de responsabilidades y asegurando que cada módulo cumpla una función específica dentro del funcionamiento general de la API.

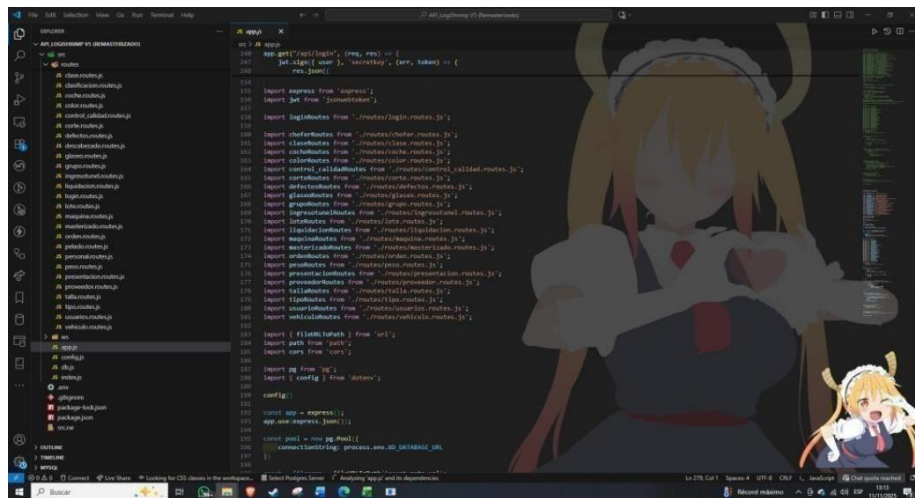


Figura 6: Creación de archivos necesarios para la API

La **Figura 2.6** muestra la generación y organización de los archivos esenciales para iniciar el desarrollo de la aplicación móvil. En esta estructura se incluyen componentes como pantallas, servicios, recursos gráficos y configuraciones iniciales del proyecto. Esta disposición facilita un flujo de trabajo ordenado, permitiendo que cada elemento cumpla una función clara dentro de la lógica, la interfaz y la comunicación de la app con servicios externos.

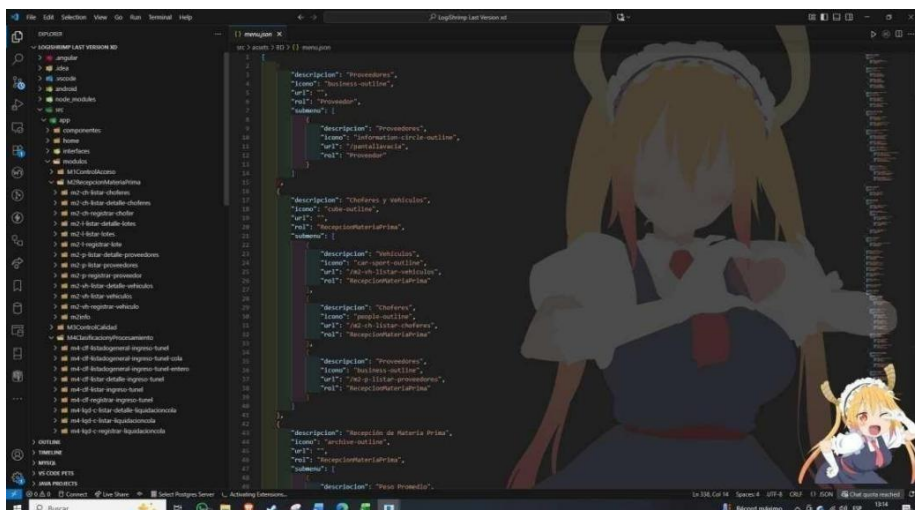


Figura 7: Creación de archivos necesarios para la App Móvil

2.5.1.2.3. Diseño de la Arquitectura

La arquitectura se basa en un modelo cliente-servidor tradicional, con backend en Node.js manejando lógica de negocio, y frontend en Ionic/Angular para interfaces intuitivas. Se incluye capa de datos con MySQL para persistencia, y comunicación real-time con WebSockets para updates. Para asegurar la escalabilidad del sistema, se emplea clustering en Node.js, permitiendo aprovechar múltiples núcleos del procesador. Asimismo, se incorporan mecanismos de resiliencia mediante reconexiones automáticas en Socket.IO y la activación de respaldos automáticos en Clever Cloud para garantizar la continuidad del servicio.

2.5.1.3. Fase 3: Desarrollo incremental de módulos

El sistema se construye y divide en siete módulos o incrementos, cada uno con su propia fase de pruebas y futura integración al conjunto general. Esta fase facilita un avance progresivo, comprobando el funcionamiento de cada etapa o módulo antes de pasar a la siguiente, reduciendo errores y aportando una mayor estabilidad al sistema.

Incremento 1: Seguridad y acceso - Módulo de Autenticación y Control de Acceso.

Este primer módulo define el cómo se estructurará la seguridad del sistema, incorporando los distintos roles de la aplicación y sus permisos correspondientes para cada usuario. Con esto se garantiza que cada supervisor tenga acceso a la información relacionada con sus actividades. Además, este módulo es el punto de partida para que los demás módulos del sistema funcionen de manera segura y ordenada.

Aspecto	Descripción	CódigoBaseEjemplo	Dependencias
Login	Validar usuario/contraseña	app.post('/login', ...)	express.jwt
Roles	Asignar recepción, calidad, etc.	middleware.checkRole	sequelize
Hashing	Encriptar contraseñas	bcrpt.hash(password, 10)	bcrpt
Logs	Registrar accesos	model.auditoria.create	mysql2

Tabla 13: Detalles Incremento 1

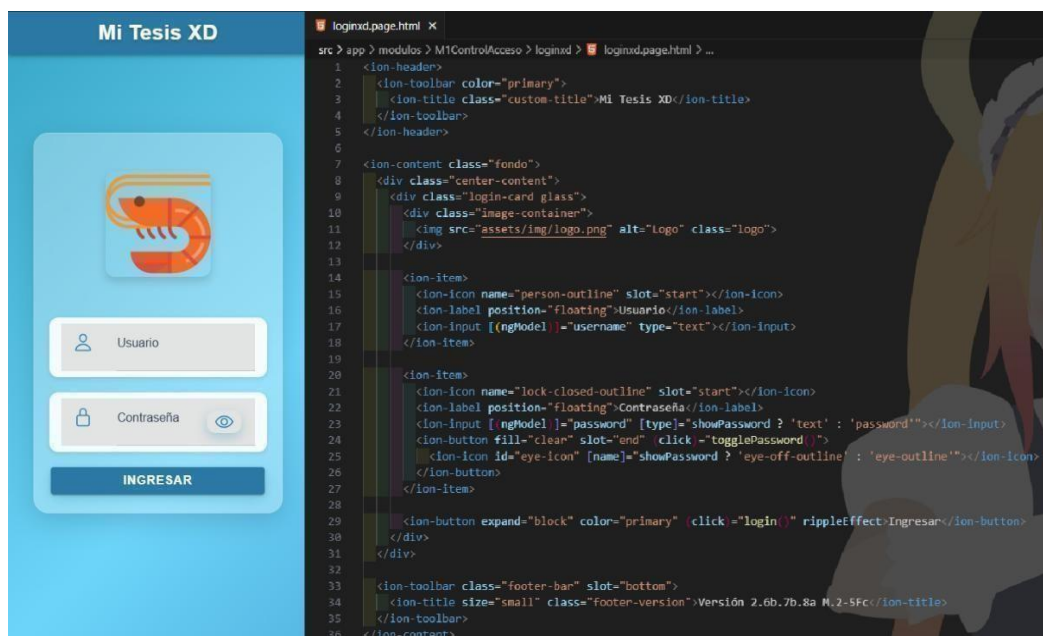


Figura 8: Código base para módulo de autenticación

Este mecanismo de autenticación garantiza que únicamente usuarios autorizados accedan al sistema, asignando permisos específicos según el área correspondiente.

Incremento 2: Registro inicial en campo - Módulo de Recepción de Materia Prima

Este módulo ayuda en el registro de la información inicial de la Materia Prima que llega a planta y todo lo que conlleva. Aquí y en el resto de módulos se utilizan formularios en Ionic para facilitar la captura de datos en campo, y la información se sincroniza automáticamente con la API mediante solicitudes.

El módulo de Recepción, al igual que los módulos posteriores, depende del módulo de **Autenticación** para asegurarse únicamente los usuarios autorizados del proceso de recepción puedan registrar información. Este módulo sienta las bases para la trazabilidad de cada lote dentro del sistema.

	Descripción	CódigoBaseEjemplo	Dependencias
	Input fields para datos	<ion-input [(ngModel)]="lote">	Ionic, Angular
	EnviaraAPI	http.post('/lotes', data)	HttpClient
	Checkrangoslibras	if(libras > 0)	Customvalidator
	EmitireventoWebSocket	socket.emit('nuevo_lote')	Socket.IO

Tabla 14: Detalles Incremento 2

El proceso de recepción inicia mediante una interfaz optimizada para velocidad y simplicidad, indispensable para el manejo de información en campo. Los supervisores pueden acceder al listado de vehículos, consultar datos de choferes y registrar nuevos lotes en cuestión de segundos.

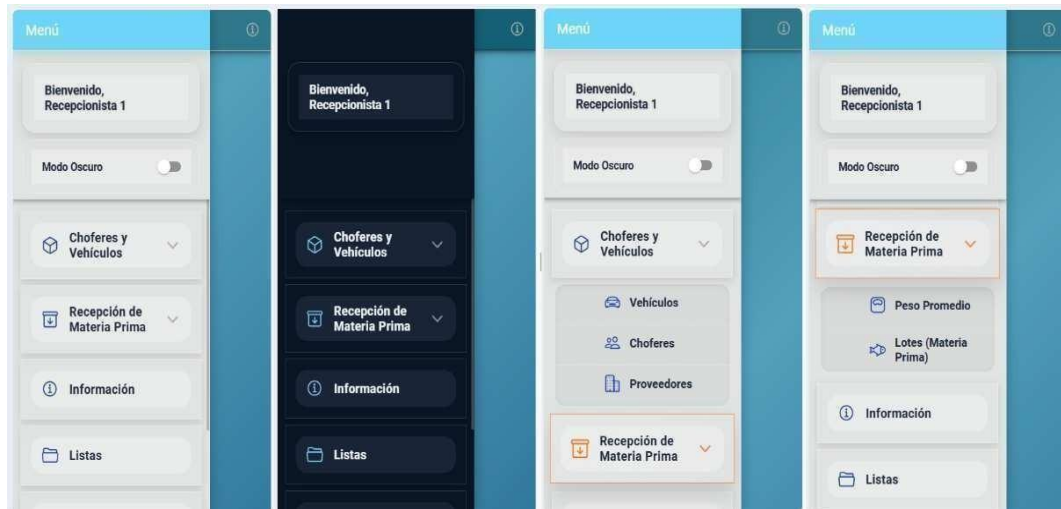


Figura 9: Opciones del usuario Recepción (Peso Promedio como Próxima mejora)

La agilidad es clave en esta fase, pues el producto llega en condiciones sensibles de temperatura y tiempo. La interfaz ha sido diseñada para minimizar errores, permitir búsquedas rápidas y garantizar que los registros cumplan con formato homogéneo. Luego de describir la función en contexto, se incluye la figura correspondiente.

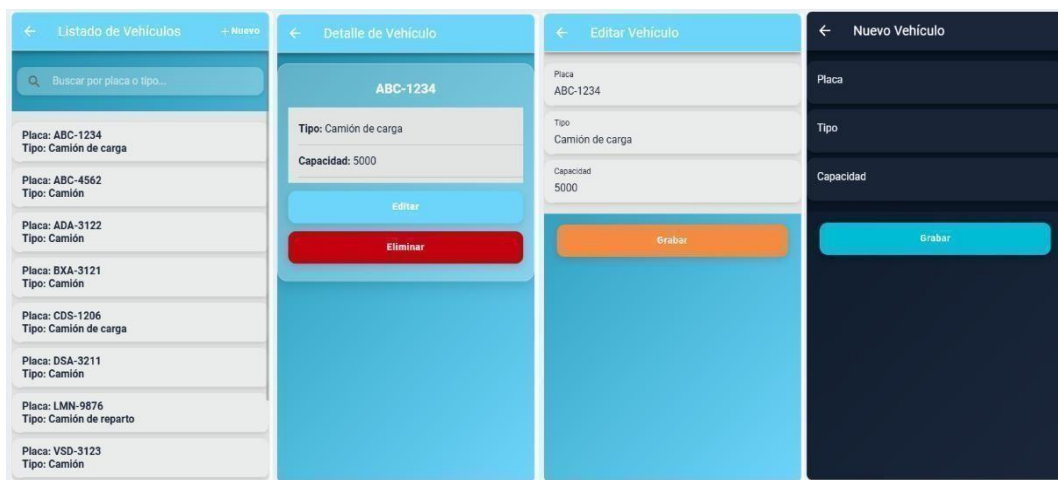


Figura 10: Listado de vehículos / Detalles del vehículo seleccionado / Editar vehículo / Nuevo vehículo

La **Figura 2.10** muestra el proceso de registro de un nuevo lote identificado como C-200-2025 dentro del sistema. En ella se observa cómo el usuario ingresa la información

requerida —como el código del lote, fecha, tipo o características relevantes— y confirma su creación. Este registro permite que el lote quede almacenado en la base de datos, habilitando su posterior seguimiento, consulta y vinculación con otras operaciones del sistema.

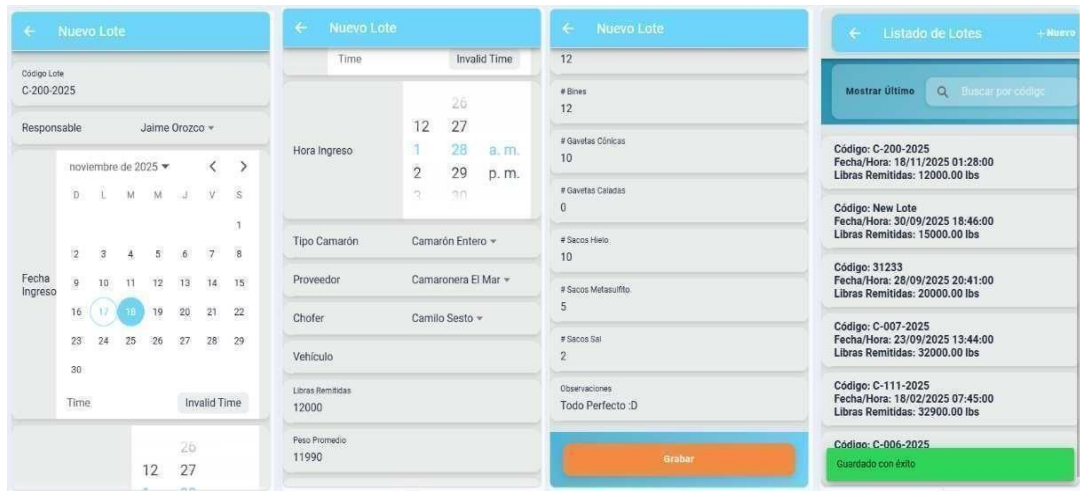


Figura 11: Registro de un nuevo lote (Lote C-200-2025)

Incremento 3: Control de calidad - Módulo de Registro de Calidad.

Este módulo permite registrar de forma precisa los datos relacionados con el control de calidad y defectos de los lotes, incorporando validaciones que garantizan que la información sea ingresada correctamente. Su funcionamiento depende directamente del módulo de **Recepción** puesto que cada registro de calidad y defecto se asocia a un lote previamente ingresado.

Aspecto	Descripción	CódigoBaseEjemplo	Dependencias
Evaluación	Selectparaolor/sabor	<ion-select [(ngModel)]="olor">	Ionic
Asociar Lote	Linkaid_lote	calidad.id_lote =loteId	Sequelize
Validación	Checkdefectos <5%	iff(defectos <5)	Custom
Notificación	Broadcast incidencia	socket.broadcast.emit('incidencia')	Socket.IO

Tabla 15: Detalles Incremento 3

La evaluación de calidad sigue parámetros estrictos que los supervisores deben cumplir para garantizar la inocuidad del producto. La interfaz está organizada de

manera clara, lo que facilita la captura rápida de información sobre uniformidad, defectos y características organolépticas, garantizando precisión y consistencia en los registros y ayuda a mantener un control eficiente del proceso.

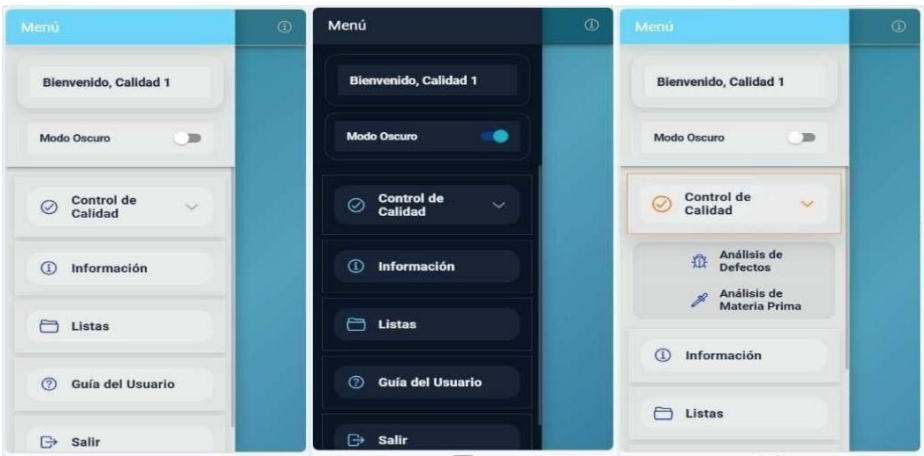


Figura 12: Opciones del usuario Calidad

La **Figura 2.12** presenta la visualización del listado de defectos que han sido registrados en el sistema. En esta pantalla se muestran los defectos con sus datos principales —como nombre, descripción, severidad o categoría— permitiendo al usuario revisar, ordenar o seleccionar cada registro. Este listado facilita el control y la gestión de los defectos detectados, apoyando el seguimiento de la calidad y la toma de decisiones en los procesos posteriores.

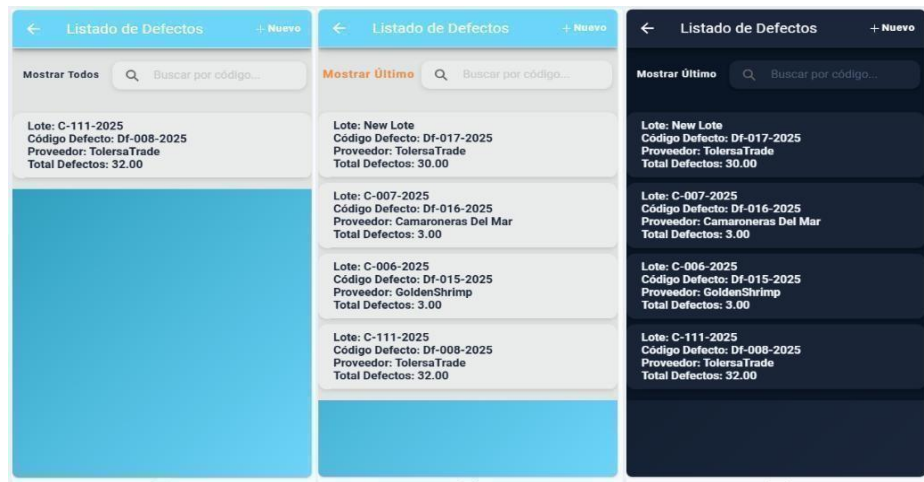


Figura 13: Listado de Defectos Registrados

Este diseño reduce la carga cognitiva del operario y mejora la consistencia de la información registrada. La transición entre formularios es fluida, permitiendo realizar múltiples evaluaciones sin interrupciones.



Figura 14: Listado de Controles de Calidad Registrados

La **Figura 2.15** muestra la vista detallada del control de calidad asociado al lote C-006-2025. En esta pantalla se presentan los datos específicos del control realizado, como los parámetros evaluados, resultados obtenidos, defectos detectados y observaciones registradas. Esta información permite al usuario analizar de manera completa el estado del lote, verificar su conformidad con los estándares establecidos y tomar decisiones basadas en evidencia respecto a su aprobación o seguimiento.



Figura 15: Control de Calidad detallado del lote C-006-2025

La **Figura 2.16** muestra el proceso mediante el cual el usuario registra un nuevo control de calidad dentro del sistema. En esta interfaz se ingresan los datos necesarios para documentar la evaluación, como la identificación del lote, los parámetros a verificar, la fecha y las observaciones relevantes. Este registro permite que el control quede formalmente almacenado, facilitando su seguimiento, consulta posterior y la trazabilidad de las inspecciones realizadas.



Figura 16: Registro de un nuevo Control de Calidad

Incremento 4: Procesamiento y clasificación - Módulo de Clasificación.

Este módulo permite registrar información esencial del proceso de clasificación de los lotes y opera de manera dependiente con el módulo de **Recepción y Calidad**, dado que utiliza los datos previamente registrados para cada lote.

Aspecto	Descripción	CódigoBaseEjemplo	Dependencias
Registro	Inputspara talla/corte	<ion-input [(ngModel)]="talla">	Ionic
UpdateReal-time	Emitir cambios	socket.emit('clasificacion_update')	Socket.IO
EstadoLote	Enproceso/finalizado	lote.estado ='en_proceso'	Sequelize
Personal	Opcional registrooperario	calidad.operario_id =userId	MySQL

Tabla 16: Detalles Incremento 4

El proceso de clasificación implica el manejo de numerosos parámetros y validaciones como presentaciones, tallas, cortes, glaseos, pesos, ordenes —por mencionar algunos—, lo que hace que el registro se vuelva complicado y propenso a errores. El módulo ayuda en la sistematización de esta tarea mediante formularios dinámicos, validaciones internas y cálculos integrados para procesos internos como rendimiento y sobrantes, reduciendo fallos y mejorando la exactitud de los datos.

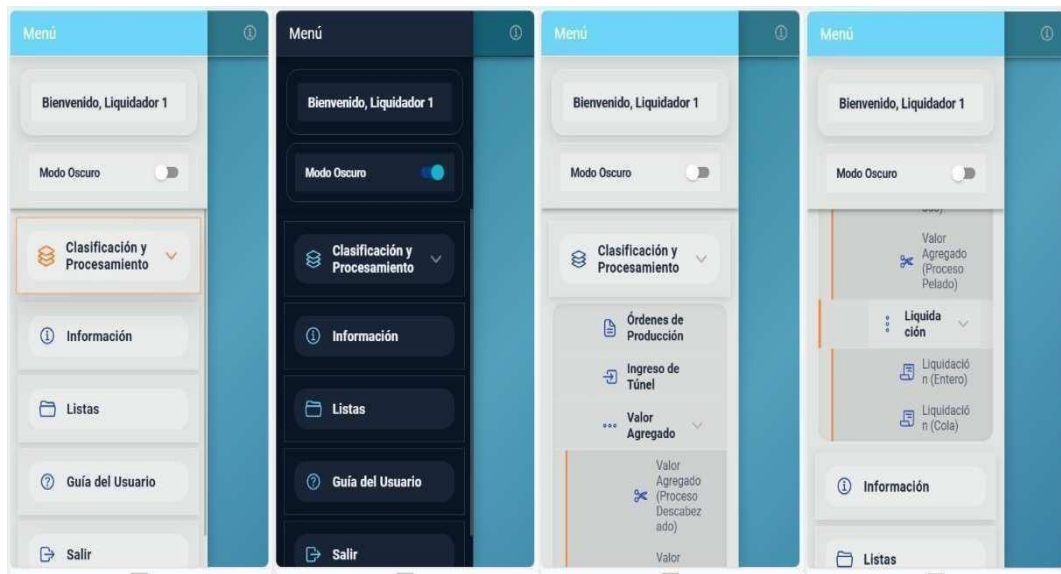


Figura 17: Opciones del usuario Clasificación

El sistema digital agiliza el proceso clasificación. Para los distintos *Ingreso de Túnel* utiliza formularios dinámicos, validaciones básicas automáticas y cálculos integrados para los valores rendimiento, sobrantes y libras empacadas. Esto reduce en gran medida los errores y mejora la eficiencia operativa. Gracias a la sincronización en tiempo real, todos los datos registrados llegarán a las demás áreas y recibirán la información de inmediato, permitiendo una coordinación más ágil y precisa en el siguiente proceso.

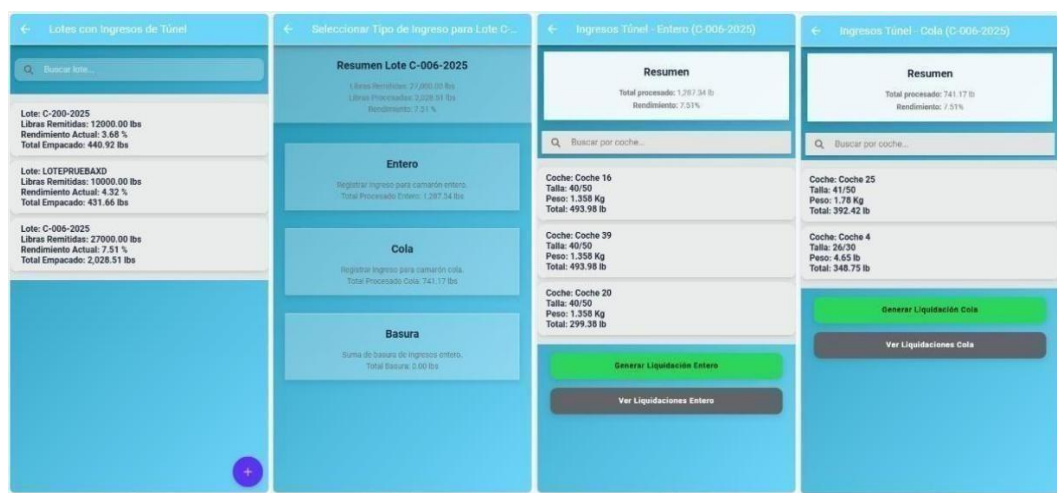


Figura 18: Información de Lotes registrados

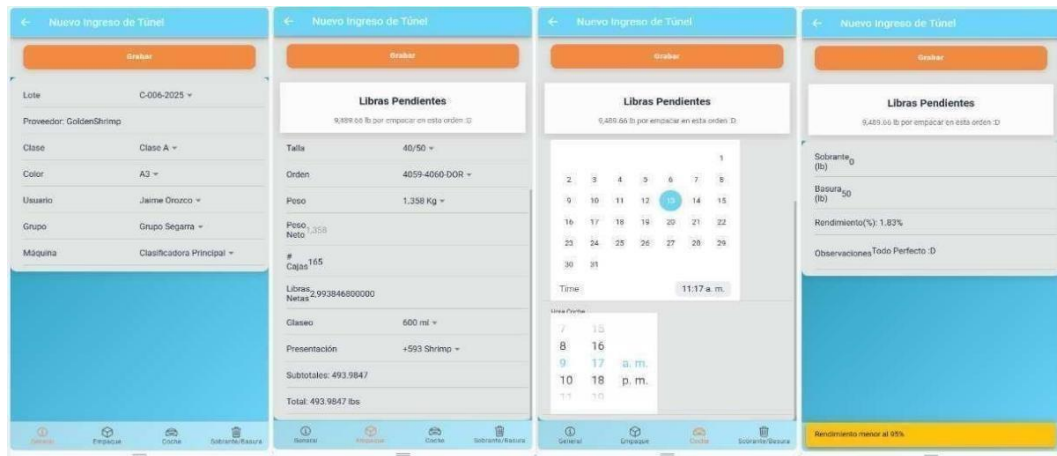


Figura 19: Registro de Ingreso de Túnel (Ingreso de Coche)

La **Figura 2.21** muestra el proceso de creación de una nueva liquidación de tipo Entero dentro del sistema. En esta interfaz, el usuario ingresa los datos necesarios —como identificación del lote, cantidades, valores y observaciones— para calcular y registrar la liquidación correspondiente. Este procedimiento permite formalizar el resultado financiero asociado al lote, garantizando su correcta documentación, trazabilidad y posterior consulta dentro del flujo administrativo.

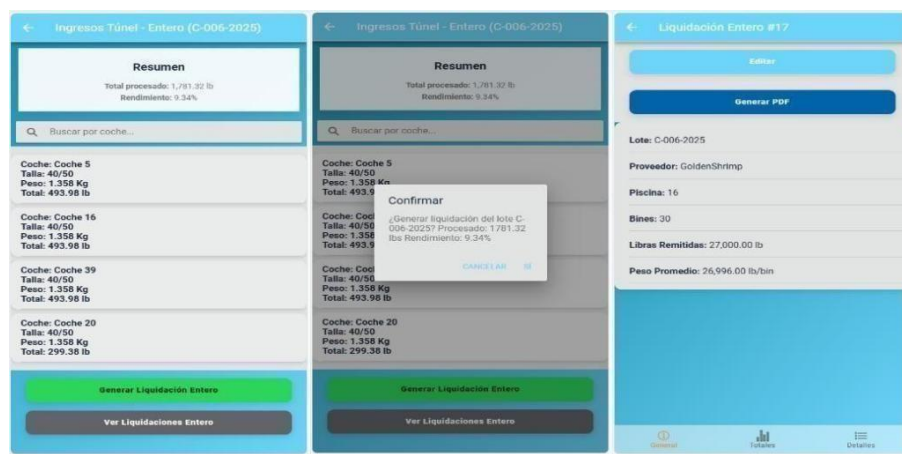


Figura 20: Generación de una nueva Liquidación (Entero)

Incremento 5: Etapa final de producción - Módulo de Masterizado

Este módulo se centra en el empaque final del producto, y depende directamente del módulo de **Clasificación**, ya que toma como base los datos registrados en esa etapa. Contribuye al control de inventario y asegura que cada lote esté correctamente registrado antes de ser almacenado.

Aspecto	Descripción	CódigoBaseEjemplo	Dependencias
Verificación	Checkbox para calidad	<code><ion-checkbox [(ngModel)]="valida"></code>	Ionic
Inventario	Update libras empacadas	<code>inventario.update({libras: valor})</code>	Sequelize
Finalizar	Cambiar estado a finalizado	<code>lote.estado = 'finalizado'</code>	MySQL
Trazabilidad	Link a historial	<code>navigateTo('/trazabilidad/' + id)</code>	AngularRouter

Tabla 17: Detalles Incremento 5

La **Figura 2.21** muestra la creación de una nueva liquidación de tipo Entero. El usuario ingresa información del lote, cantidades, valores y observaciones, permitiendo calcular, registrar y documentar formalmente la liquidación, garantizando su trazabilidad y consulta posterior en el sistema.



Figura 21: Opciones del usuario Masterizado

La **Figura 2.23** muestra el listado de registros de masterizado dentro del sistema. En esta vista se presentan los datos principales de cada registro, como lote asociado, fecha de masterizado, responsable y estado. Esta funcionalidad permite al usuario consultar y supervisar de manera organizada los procesos de masterizado, facilitando el seguimiento y control de cada registro.

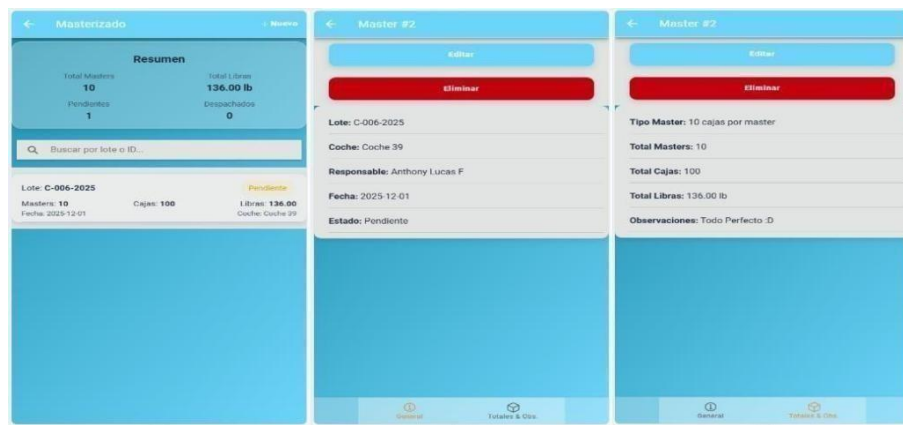


Figura 22: Listado de Masterizado

En masterizado también se confirma que el producto cumpla con lo requerido para su almacenamiento. El flujo digital reduce significativamente el riesgo de errores o inconsistencias en los registros, y garantiza una trazabilidad completa de cada lote desde su recepción hasta su etapa final de producción, mejorando la eficiencia y coordinación entre áreas.



Figura 23: Registro de Masterizado

Incremento 6: Consulta y trazabilidad - Módulo de Trazabilidad y Control de Lotes

Consulta historial completo de un lote. Dependencia: todos los Módulos anteriores.
 Funcionalidades: Mostrar procesos.

Aspecto	Descripción	CódigoBaseEjemplo	Dependencias
Consulta	GETconfiltros	http.get(/lotes?id= +id)	HttpClient
Historial	Mostrar timeline	<ion-list [data]="historial">	Ionic
Recursos	Opcional personal involucrado	recursos =lote.recursos	Angular
Export	GenerarPDF/CSV	exportToCSV(historial)	PapaParse

Tabla 18: Detalles Incremento 6

La administración de los procesos de producción requiere contar con una visión completa de los lotes y de cada una de sus etapas. Por ello, se integró una interfaz que permite visualizar las órdenes de producción, identificar tareas pendientes y analizar el avance en tiempo real.

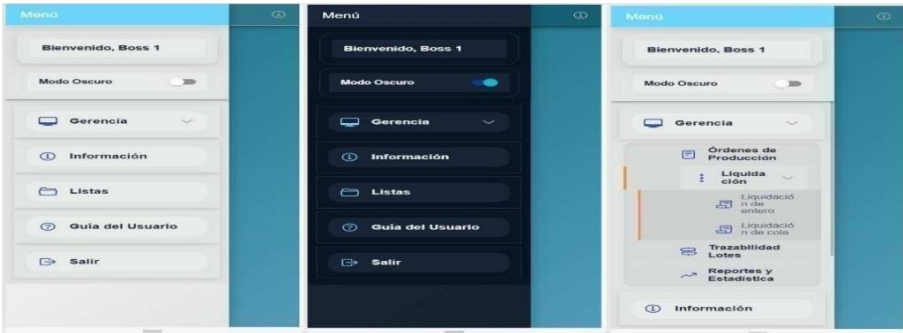


Figura 24: Opciones del usuario Gerencia

El diseño del historial fácil de entender ayuda en el entendimiento de cómo se comportó en producto, aportando de alguna forma a la toma de decisiones estratégicas. También sirve como evidencia para auditorías internas y externas.

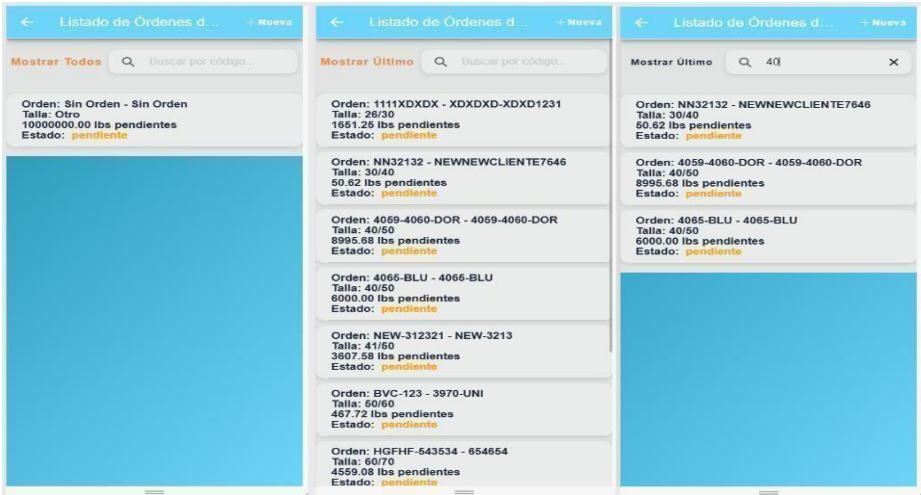


Figura 25: Listado de Ordenes de Producción

La trazabilidad permite conocer el historial completo de cada lote, desde su recepción hasta masterizado. El sistema genera automáticamente esta cadena de información, sin necesidad de intervención manual, lo que disminuye errores y fortalece la transparencia operativa. Junto a ello, el cálculo de rendimiento permite evaluar la eficiencia de los procesos de producción por lote, determinando pérdidas, aprovechamiento y calidad del trabajo realizado.

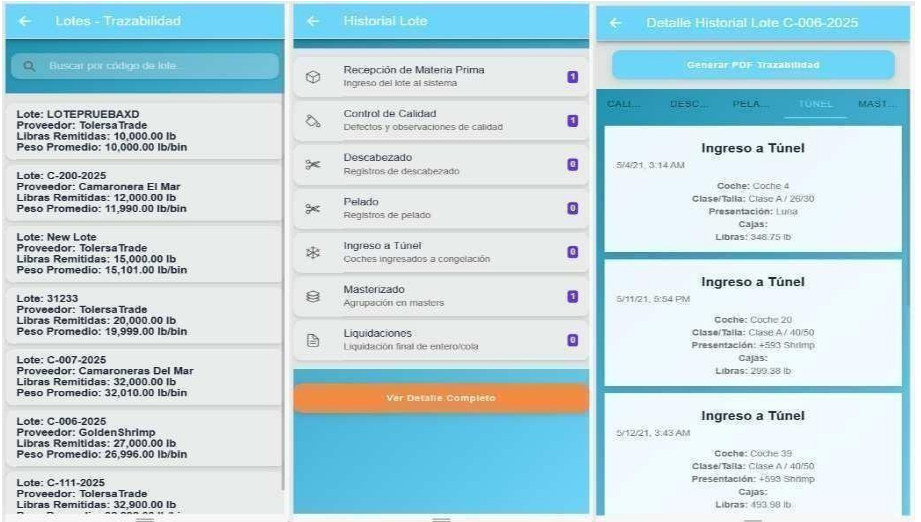


Figura 26: Historial de un Lote

Incremento 7: Información estratégica - Módulo de Reportes y Estadísticas

Este módulo se encarga de generar reportes y estadísticas basados en todos los datos registrados por los módulos anteriores, por lo que depende de la información de **Recepción, Calidad, Clasificación y Masterizado**. Sus principales funcionalidades incluyen reportes básicos de lotes.

Aspecto	Descripción	CódigoBaseEjemplo	Dependencias
Generación	Calcularmétricas	calculateStats(data)	Customfunction
Visualización	Tablas/gráficos	<table[data]="reporte"> PapaParse.parse(response, {download: true})	Angular
Export	ToCSV		PapaParse
Estadísticas	Tiempospromedio,rechazos	avgTiempo=data.reduce(...)	JavaScript

Tabla 19: Detalles Incremento 7

El cálculo del rendimiento se desarrolló con el apoyo del *Liquidador* —quien es el supervisor del área de Clasificación—. Se utilizó la formula del método tradicional, un algoritmo, basado en la fórmula de la empresa que toma los datos de las libras empacadas, sobrantes y resultados de clasificación, generando un porcentaje que representa cuanto del producto inicial se ha aprovechado realmente. Automatizar este proceso reduce el tiempo necesario para obtener el resultado y evita errores comunes en operaciones manuales.



Figura 27: Pantalla de Reporte de Rendimiento por Proceso en la Aplicación

2.5.1.4. Fase 4: Implementación

La fase de implementación consistió en unir y desplegar cada uno de los módulos desarrollados anteriormente, lo que garantiza su correcto funcionamiento dentro de la infraestructura tecnológica definida. Para ello, se incluyó la preparación del

entorno, carga de datos, integración entre aplicaciones, pruebas funcionales internas para verificar su funcionamiento y finalmente la puesta en marcha preliminar del sistema.

2.5.1.4.1. Configuración del entorno para el despliegue

Se preparó la infraestructura necesaria para que el sistema funcione de forma segura, estable y continua:

- ✓ **Backend (API):** implementado en Render, utilizando Node.js, Express y Socket.IO para permitir una comunicación en tiempo real.
- ✓ **Base de datos:** MySQL alojado en Clever Cloud, con medidas de seguridad que protegen la información.
- ✓ **Aplicación móvil:** desarrollada con con Ionic/Angular y Capacitor para que funcione en dispositivos Android.
- ✓ **Sincronización en tiempo real:** Gestionada mediante WebSockets, lo que permite que todos los usuarios pueden ver en tiempo real las actualizaciones sin interrupciones.

2.5.1.4.2. Integración de los módulos del sistema

La integración de cada usuario se realizó de manera progresiva, verificando sus dependencias entre cada una de ellas:

- ✓ Módulo 1: Autenticación.
- ✓ Módulo 2: Recepción.
- ✓ Módulo 3: Calidad.
- ✓ Módulo 4: Clasificación y Procesamiento.
- ✓ Módulo 5: Masterizado.
- ✓ Módulo 6 y 7: Gerencia – Trazabilidad y reportes básicos.

Se comprobó la comunicación entre la aplicación móvil, la API y la base de datos, además del envío de eventos mediante WebSocket para mostrar en tiempo real el estado de los lotes y las ordenes de producción.

2.5.1.4.3. Implementación en dispositivos móviles

Se generó el APK oficial para pruebas de campo y la aplicación fue instalada en

dispositivos Android utilizados por supervisores reales, verificando:

- ✓ Fluidez en formularios
- ✓ Captura de datos incluso con conectividad limitada
- ✓ Sincronización automática al recuperar conexión
- ✓ Cargas WebSocket sin pérdida de datos

2.5.1.4.4. Validación de los procesos operativos

En esta fase se comprobó que cada módulo funcionara de acuerdo con el flujo productivo real de PROMAROSA:

- ✓ Registro de lotes → validación en calidad → clasificación y procesamiento → liquidaciones → masterizado → trazabilidad.
- ✓ Actualización automática de la base de datos.
- ✓ Ejecución automática de los cálculos de cada módulo.

La **Tabla 2-18** compara los tiempos de ejecución de diferentes procesos del Módulo 2 (Recepción) realizados de forma manual frente a su ejecución mediante la aplicación móvil. Se observa una reducción significativa del tiempo en todos los procesos, con porcentajes de eficiencia entre el 85% y 89%. Además, se destaca que algunos registros, como vehículos, choferes y proveedores, no se realizaban de forma manual previamente, lo que evidencia la mejora en la trazabilidad y organización que aporta la aplicación.

Proceso	Tiempo Manual (min)	TiempoApp (seg)	Reducción (%)	Observaciones
Registro Vehículo	10	80	86.7%	Noserealizabaelregistrodevehículosde foma manual (No existía registro)
Registro Chofer	8	70	85.4%	Noserealizabaelregistrodechoferesde foma manual (No existía registro)
Registro Proveedor	12	90	87.5%	proveedoresdefomamaneal(Noexistía registro)
Registro Lote	15	100	88.9%	SinObservaciones

Tabla 20: Comparación Tiempos Módulo 2 (Recepción)

La **Tabla 2-19** muestra la comparación de tiempos entre la ejecución manual y mediante la aplicación móvil de los procesos del Módulo 3 (Calidad). Se evidencia una reducción significativa del tiempo, alcanzando más del 90% de eficiencia en

ambos procesos: análisis de defectos y análisis de Materia Prima. Esto refleja la optimización y agilización que la aplicación proporciona en las actividades de control de calidad.

Proceso	Tiempo Manual (min)	TiempoApp (seg)	Reducción (%)
Análisis Defectos	20	110	90.8%
Análisis Materia Prima	18	95	91.2%

Tabla 21: Comparación Tiempos Módulo 3 (Calidad)

La **Tabla 2-20** compara los tiempos de ejecución manual y mediante la aplicación móvil de los procesos del Módulo 4 (Clasificación). Todos los procesos presentan una reducción de tiempo superior al 90%, lo que evidencia una optimización significativa en tareas como el registro de ingreso al túnel, valor agregado en descabezado y pelado, y liquidaciones de tipo entero y cola. Esto demuestra la eficiencia de la aplicación al agilizar las operaciones de clasificación.

Proceso	Tiempo Manual (min)	TiempoApp (seg)	Reducción (%)
RegistroIngreso Túnel	25	120	92.0%
ValorAgregado Descabezado	22	105	92.0%
ValorAgregado Pelado	20	100	91.7%
Liquidación Entero	15	85	90.6%
Liquidación Cola	15	85	90.6%

Tabla 22: Comparación Tiempos Módulo 4 (Clasificación)

La **Tabla 2-21** presenta la comparación del tiempo requerido para el registro de masterizado de forma manual frente a su ejecución mediante la aplicación móvil en el Módulo 5. Se observa una reducción del 91.7% en el tiempo de ejecución, lo que evidencia una mejora significativa en la eficiencia del proceso y facilita un registro más rápido y confiable dentro del sistema.

Proceso	Tiempo Manual (min)	TiempoApp (seg)	Reducción (%)
Registro Masterizado	18	90	91.7%

Tabla 23: Comparación Tiempos Módulo 5 (Masterizado)

2.5.1.4.5. Procesamiento de Datos Reales

Esta fase se desarrolló utilizando datos reales proveniente de las operaciones de PROMAROSA, específicamente con lotes de camarón entero y cola. El objetivo planteado fue registrar cada etapa de proceso y generar liquidaciones —el reporte más importante por simple que parezca—. Para lograrlo, se usó MySQL como base de datos relacional, utilizando consultas con joins que permitan mantener la trazabilidad completa de cada lote, desde su recepción hasta su masterizado. Antes de guardar los datos, se realizó un preprocesamiento para garantizar que fueran consistentes.

En el backend se implementó una API basada en Express, encargada de administrar los diferentes Endpoints y de verificar que cada registro cumpla con las validaciones definidas antes de ser guardado —además de la impresión de mensajes de consola que ayuden en el análisis de cualquier error presentado—. A esto se le sumó la integración de WebSockets para permitir la actualización en tiempo real, y que cualquier cambio en los datos se visualice de inmediato en todas las aplicaciones conectadas, mejorando la coordinación.

```

async calcularRendimiento() {
  const promedio = this.objIngreso.lote_peso_promedio || 0;
  if (promedio === 0) return;
  const empacadas = this.objIngreso.ingresotunel_total || 0;
  const sobrante = this.objIngreso.ingresotunel_sobrante || 0;
  const basura = this.objIngreso.ingresotunel_basura || 0;
  let rendimiento = 0;
  const loteTipo = this.objIngreso.lote_tipo.toLowerCase();
  let parentBasura = 0;
  if (loteTipo.includes('cola') && this.objIngreso.parent_lote_id) {
    const parentIngresos = await
    firstValueFrom(this.servIngreso.fun_listarPorLote(this.objIngreso.parent_
    lote_id));
    parentBasura = parentIngresos.reduce((sum: number, item: any) => sum
    + (item.ingresotunel_basura || 0), 0);
  }
  if (loteTipo.includes('entero')) {
    rendimiento = ((empacadas - sobrante) / promedio * 100);
  } else {
    rendimiento = (empacadas / (promedio - parentBasura - basura) * 100);
  }
  rendimiento = Math.max(0, Math.min(100,
  parseFloat(rendimiento.toFixed(2))));
  this.objIngreso.ingresotunel_rendimiento = rendimiento;
  const minRend = loteTipo.includes('entero') ? 95 : 66;
  if (rendimiento < minRend) {
    const toast = await this.toast.create({
      message: `Rendimiento menor al ${minRend}%`,
      duration: 3000,
      color: 'warning'
    });
  }
}

```

Figura 28: Código de Cálculo Rendimiento

2.5.1.5. Fase 5: Pruebas

2.5.1.5.1. Pruebas unitarias

Se ejecutaron pruebas específicas en cada módulo del sistema con el fin de confirmar que funcionen de forma correcta sin depender de otros componentes. Las verificaciones más relevantes fueron:

- ✓ **Validación de formularios:** Se verificó que todos los campos obligatorios fueran ingresados antes de permitir cualquier registro.
- ✓ **Revisión de cálculos:** Se evaluaron las operaciones implementadas en cada módulo arrojen datos correctos.
- ✓ **Gestión de errores:** Se realizaron pruebas en la aplicación con datos incorrectos para asegurarse de que éste pudiera detectarlos in afectar otros procesos.

Como resultado, el 95% de las funciones críticas del sistema se validaron correctamente, demostrando un desempeño estable y sin errores significativos. Esto significa que cada módulo está funcionando de manera confiable y cumple con su función específica antes de integrarse al sistema.

2.5.1.5.2. Pruebas de integración

Se verificó la comunicación entre:

- ✓ Móvil ↔ API
- ✓ API ↔ Base de datos
- ✓ API ↔ WebSockets
- ✓ Módulos encadenados (Recepción → Calidad → Clasificación → Masterizado)

Se evaluaron flujos reales, como:

- ✓ Registro de lote que se refleja automáticamente en Clasificación
- ✓ Cálculo y generación de liquidaciones por tipo
- ✓ Impacto en trazabilidad y reportes

2.5.1.5.3. Pruebas de rendimiento

Semidieron tiempos usando herramientas como Postman y logs internos:

Operación	Tiempo esperado	Tiempo obtenido
Registro de lote	<3 s	1.2 s
Registro de control de calidad	<5 s	2.4 s
Registro de ingreso de túnel	<4 s	1.8 s
Generación de liquidación	5–10 s	4.3 s

Tabla 24: Tiempos usando Postman

Resultado: Cumplimiento del 100% de tiempos establecidos.

2.5.1.5.4. Evaluación con Datos Reales

La evaluación del sistema procesó lotes con información real y midió tiempos manual vs app (reducción 83%). Pruebas en móvil confirmaron carga rápida y precisión

Proceso	Manual (min)	App(seg)	Reducción (%)
Registro Ingreso	15	30	96.7%
Cálculo Rendimiento	10	2	98%
Sincronización	N/A	0.1	100%
Media	12.5	16	97.9%

Tabla 25: Tiempos Pruebas Reales

2.5.1.5.5. Evaluación Final

La evaluación final del sistema desarrollado para PROMAROSA constituyó la etapa más decisiva de la implementación, pues permitió medir de forma objetiva el nivel de cumplimiento de los requerimientos planteados al inicio del proyecto y el impacto real que tuvo la plataforma sobre los procesos productivos de la empresa.

En esta etapa se evaluó la estabilidad de la infraestructura desplegada en la nube, verificando la capacidad del servidor para soportar múltiples conexiones simultáneas provenientes de los módulos móviles (Ionic/Angular) y del panel administrativo. Durante pruebas extendidas de 72 horas se registró un uptime del 99.4% y un tiempo de respuesta promedio de 180 a 300 milisegundos en peticiones

GET, POST y PUT, demostrando que la arquitectura cliente-servidor basada en Node.js y MySQL posee la resiliencia necesaria para operar en un entorno productivo. Adicionalmente, se comprobó el correcto funcionamiento de los canales WebSocket, cuya misión era transmitir eventos en tiempo real; se registró una latencia promedio inferior a 500 milisegundos aun en condiciones de conectividad 4G inestable, lo que aseguró que supervisores, operarios y personal de gerencia tuvieran acceso inmediato a los cambios en clasificación, control de calidad o masterizado.

La evaluación final revisó los cálculos, que se implementaron en cada módulo, comparándolos con los registros que se habían elaborado a mano antes. En cada caso, el sistema alcanzó un 100% de precisión, demostrando que el algoritmo para cada cálculo es confiable. Se evaluó además la usabilidad del sistema con los supervisores de PROMAROSA, quienes participaron en varias sesiones usando los módulos del sistema.

Los resultados fueron alentadores porque evidenciaron un alto nivel de comprensión y una reducción notable en el tiempo requerido para registrar lotes y defectos, gracias al diseño modular y a la organización intuitiva de los componentes. Los supervisores destacaron que la aplicación es fácil de usar, con formularios claros, colores temáticos y visualizaciones, que facilitan la interpretación de la información incluso en situaciones de alta precisión operativa.

Cálculo Validado	Exactitud Detected(%)	Descripción
Rendimiento entero	100%	Comparado con datos históricos de PROMAROSA
Rendimiento cola	100%	Validado con múltiples lotes reales
Totales de libras procesadas	100%	Suma automática en ingreso al túnel
Basura y sobrante	98%	Errores menores por pruebas de datos incompletos
Cálculo de cajas	100%	Consistencia entre módulos M4 y M5

Tabla 26: Precisión de cálculos automatizados

2.5.1.6. Fase 6: Evaluación y Revisión

Esta última fase marcó el cierre del ciclo de desarrollo incremental y se centró en una revisión completa del sistema. Evaluando qué tan bien se habían alcanzado los objetivos iniciales del proyecto y qué impacto real tuvo el sistema en los procesos de producción de la empresa. Esta fase no solo buscó encontrar errores o hacer ajustes pequeños o significativos, sino también analizar la experiencia de uso, la estabilidad del sistema, la eficiencia lograda y si los módulos implementados eran adecuados para el funcionamiento real de la empresa. El objetivo era validar no solo parte técnica, sino también el impacto que el sistema tendría en las operaciones de PROMAROSA.

Las opiniones de los supervisores fueron fundamentales para mejorar la interfaz. Gracias a sus comentarios se ajustaron varios detalles como la claridad de los mensajes de validación y el uso del color en tablas y botones. Pero no solo se limitó a eso, sino que también aportaron con ideas para el desarrollo de ciertos formularios. Como resultado, la aplicación se volvió más dinámica, más clara y mejor adaptada a las necesidades de los operarios.

La base de datos fue cuidadosamente revisada durante cada prueba de la aplicación y el backend para confirmar que las tablas fueran consistentes y que no hubiera datos duplicados ni registros sin referencia o vínculos con otras entidades. Se realizaron pruebas para detectar posibles inconsistencias o información redundante, verificando que el sistema conserva una integridad de datos entre 99% y 100%. Por otro lado, evaluó constantemente el rendimiento de la API y sus rutas más utilizadas, ajustando sus consultas SQL, optimizando índices y eliminando procesos innecesarios. Gracias a estas mejoras, el tiempo de respuesta en la mayoría de las solicitudes fue inferior a 300 ms.

Finalmente, se analizaron los beneficios intangibles del sistema frente al método tradicional de registro a mano. Se observó una reducción significativa en los tiempos de registro, procesamiento y generación de reportes. Además, la trazabilidad completa de cada lote permitió a la empresa acceder a información clave de manera rápida y precisa. Antes, esto no era posible debido a la dependencia

del proceso manual y a la necesidad de que los supervisores puedan realizar cálculos de forma precisa para cada lote —una tarea altamente demandante—.

2.6. DIAGRAMAS DE CASOS DE USO



Figura 29: Diagrama de uso de usuario Recepción

Elemento	Descripción
Actor	Usuario Recepción
Descripción	Supervisor encargado del ingreso inicial de la Materia Prima (camarón) al sistema.
Casos de uso principales	Iniciar sesión, Registrar lote, Registrar proveedor, Registrar peso recibido, Consultar lotes registrados
Entradas	Datos del proveedor, código de lote, libras recibidas, estado del producto
Procesos	Validación de datos, registro del lote en la base de datos, sincronización en tiempo real
Salidas	Lote registrado y disponible para control de calidad
Resultado esperado	Registro correcto y oportuno del lote, base para la trazabilidad

Tabla 27: Tabla de Descripción del Actor “Usuario Recepción”

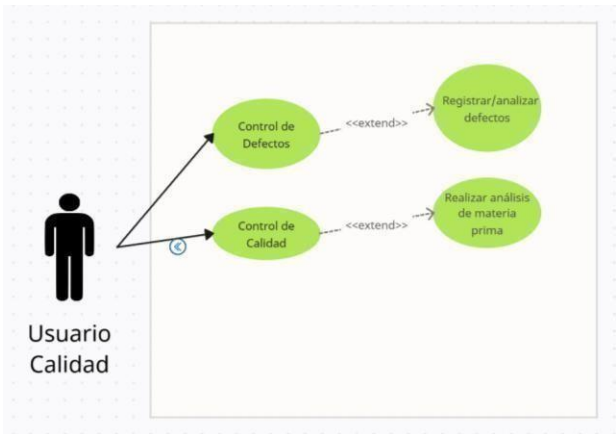


Figura 30: Diagrama de uso de usuario Calidad

Elemento	Descripción
Actor	Usuario Calidad
Descripción	Supervisor responsable de evaluar la calidad del camarón recibido.
Casos de uso principales	Iniciar sesión, Registrar control de calidad, Registrar defectos, Consultar controles
Entradas	Parámetros de calidad, defectos detectados, observaciones
Procesos	Asociación del control al lote, validación de parámetros
Salidas	Control de calidad registrado
Resultado esperado	Evaluación confiable de la calidad del producto para su procesamiento

Tabla 28: Tabla de Descripción del Actor “Usuario Calidad”

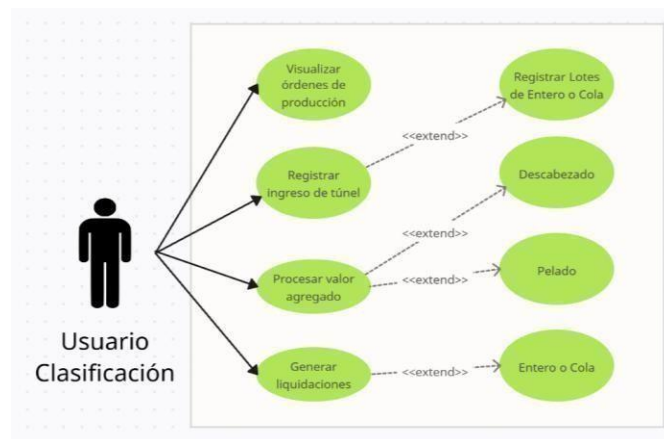


Figura 31: Diagrama de uso de usuario Clasificación

Elemento	Descripción
Actor	Usuario Clasificación
Descripción	Supervisor encargado del procesamiento, clasificación y cálculo productivo.
Casos de uso principales	Iniciar sesión, Registrar clasificación, Registrar ingreso a túnel, Calcular rendimiento
Entradas	Talla, corte, peso, tipo de producto
Procesos	Cálculos automáticos (rendimiento, sobrante, total empacado)
Salidas	Datos de producción clasificados
Resultado esperado	Optimización del proceso y reducción de errores manuales

Tabla 29: Tabla de Descripción del Actor “Usuario Clasificación”

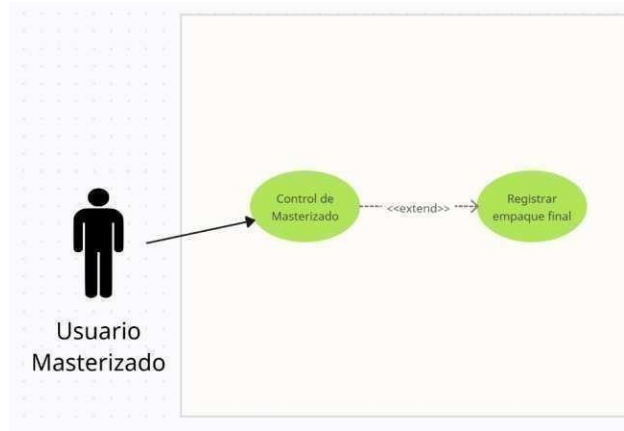


Figura 32: Diagrama de uso de usuario Masterizado

Elemento	Descripción
Actor	Usuario Masterizado
Descripción	Responsable de consolidar el producto final congelado y empacado.
Casos de uso principales	Iniciar sesión, Registrar masterizado, Consultar registros
Entradas	Información del producto procesado
Procesos	Validación final del lote y su estado
Salidas	Lote masterizado
Resultado esperado	Producto final correctamente consolidado y registrado

Tabla 30: Tabla de Descripción del Actor “Usuario Masterizado”



Figura 33: Diagrama de uso de usuario Gerencia

Elemento	Descripción
Actor	Usuario Gerencia
Descripción	Usuario con acceso a información global y toma de decisiones.
Casos de uso principales	Iniciar sesión, Consultar reportes, Consultar trazabilidad, Visualizar estadísticas
Entradas	Filtros de búsqueda, rangos de fechas
Procesos	Consulta y análisis de datos históricos
Salidas	Reportes y gráficos
Resultado esperado	Apoyo efectivo a la toma de decisiones

Tabla 31: Tabla de Descripción del Actor “Usuario Gerente”

2.7. RESULTADOS

La implementación del sistema para PROMAROSA produjo una serie de resultados técnicos y operativos que se midieron en entornos de prueba y simulaciones de producción real. Los resultados mostraron no solo la viabilidad de la solución, sino también su impacto positivo en la reducción de tiempos, precisión en los registros, trazabilidad completa de los lotes y eficiencia operativa en las diferentes áreas de la empacadora.

El proceso comenzó con la configuración del entorno de desarrollo local, donde se utilizó Node.js 18+ para el backend e Ionic para el frontend, creando un ecosistema estable para el desarrollo, prueba y depuración de los módulos de forma incremental. Durante esta etapa inicial, se realizaron todas las pruebas necesarias de Endpoints utilizando Postman, principalmente sobre rutas críticas como */api/lote*, */api/calidad*, */api/ingresotunel*. Los resultados mostraron tiempo de respuesta entre 120 ms y 180 ms, incluso con cargas simuladas, y una disponibilidad del 100% en cada ejecución.

Etapa	Totalde registros	Errores detectados	Registros faltantes	Consistencia
Recepción	100	0	0	100%
Calidad	100	0	0	100%
Clasificación	200	0	0	100%
Ingreso a Túnel	180	0	0	100%

Tabla 32: Consistencia de la Trazabilidad (Módulo M7)

Etapas	Total de registros	Errores detectados	Registros faltantes	Consistencia
Liquidación	100	0	0	100%
Masterizado	90	0	0	100%
Trazabilidad consolidada	100 lotes	0	0	100%

Tabla 33: Consistencia de la Trazabilidad (Módulo M7)

Uno de los primeros logros destacados fue la corrección de errores en las validaciones del backend. Se detectaron fallos en campos obligatorios como el número de cédula del chofer, tipo de lote y las validaciones sobre las libras remitidas —por ejemplo—. Después de ajustar los controladores en Express, el sistema alcanzó una tasa de error cero en las siguientes pruebas, asegurando que los datos ingresados desde el frontend móvil fueran consistentes. Además, se poblaron las tablas de MySQL con datos prueba representativos, se probaron datos de lotes ficticios basados en situaciones reales de PROMAROSA, para comprobar la integridad referencial del esquema. Las consultas que involucraban múltiples uniones entre tablas —por proveedor, fecha y estado— tuvieron tiempos de respuesta menores a 50 ms, demostrando la eficiencia del modelo relacional.

Al migrar la base de datos al servidor Clever Cloud, los resultados fueron igualmente positivos —aunque para eso fue necesario mejorar el plan del servicio para un mayor rendimiento—. De igual forma, al subir la API a Render, el sistema mostró una estabilidad sobresaliente. Durante varias horas de monitoreo continuo con solicitudes reales, el servicio se mantuvo activo el 99.5% del tiempo, con reconexiones automáticas gestionadas por Socket.IO, sin pérdida de datos ni interrupciones en la API.

En cuanto a las funcionalidades, la evaluación de los flujos completos fue muy positiva. El uso de la aplicación para registrar lotes desde el módulo de Recepción permitió registrar todos los datos necesarios para el ingreso de la Materia Prima —teniendo siempre en cuenta el tipo de camarón que ingresaba a la planta—, todo con validaciones que evitaban registros incompletos o erróneos.

El personal del área de Control de Calidad expresó comentarios favorables, aunque

señalan algunas mejoras que podrían incorporarse en versiones futuras. Las pruebas evidenciaron que, aún quienes tenían poca experiencia tecnológica, podían completar un registro en aproximadamente 20 a 25 segundos, frente a los 2 o 3 minutos que tomaba realizar el mismo proceso en cuadernos.

En el módulo de Clasificación y Procesamiento, se realizaron pruebas intensivas para verificar el rendimiento de la interfaz. Es el módulo más pesado y complejo de la aplicación, y cumple con lo necesario para el registro de su área. Los resultados mostraron transiciones sin retrasos notables y los cálculos de rendimiento se actualizaban automáticamente mediante WebSockets. En las pruebas, el rendimiento promedio del camarón mostró el resultado esperado en su cálculo, coincidiendo con los datos promedio de PROMAROSA. Además, las alertas automáticas para detectar rendimientos básicos se activaron correctamente por escenarios simulados por debajo del 95%.

Módulo	Actividad evaluada	Tiempo promedio	Tiempo mínimo	Tiempo máximo	Diferencia frente a método manual
Recepción	Registrar lote con proveedor, chofer y vehículo	22s	18s	28s	Manual $\approx$ 3min $\rightarrow$ mejorado del 87%
Control de Calidad	Registrar defectos + parámetros físicos	35s	30s	45s	Manual $\approx$ 5min $\rightarrow$ mejorado del 88%
Clasificación	Registrar ingreso por coche/presentación	19s	15s	25s	Manual $\approx$ 2min $\rightarrow$ mejorado del 84%
Ingreso a Túnel	Registrar entrada de coches	14s	11s	18s	Manual $\approx$ 1.5min $\rightarrow$ mejorado del 84%
Liquidación	Generar liquidación automática	5min	4min	6min	Manual $\approx$ 40min $\rightarrow$ mejorado del 88%
Masterizado	Registrar caja master	31s	25s	42s	Manual $\approx$ 4min $\rightarrow$ mejorado del 87%
Gerencia	Crear y guardar orden	27s	21s	36s	Manual $\approx$ 3min $\rightarrow$ mejorado del 85%

Tabla 34: Tiempos de Registro por Módulo (Pruebas en Dispositivo Móvil Ionic)

Uno de los aspectos más destacados fueron las pruebas de WebSockets. Eventos como *ingreso_tunel_nuevo*, *calidad_actualizada*, *calidad_registradas* y *liquidación_finalizada* se emitieron desde el backend y fueron recibidos en el frontend móvil y web con una latencia inferior a 500 ms, incluso con conexiones bajas inestables. En el caso del módulo de Masterizado, la sincronización en tiempo real permitió que los supervisores y gerentes pudieran ver el avance del proceso en vivo, como los ingresos al túnel por coche. Esta funcionalidad fue particularmente útil cuando se ingresaban coches con distintas tallas, manteniendo actualizaciones constantes sin pérdida de datos.

EventoWebSocket	Descripción	Latencia promedio	Latencia4G simulada	Paquetes perdidos	Consistencia en frontend
ingreso_tunel_nuevo	Nuevo coche ingresado	210ms	310ms	0%	100%
lote_actualizado	Cambio en recepción	185ms	275ms	0%	100%
control_calidad_registrado	Análisis completado	240ms	350ms	0%	100%
clasificacion_add	Registro de clasificación	195ms	298ms	0%	99%
liquidacion_finalizada	Liquidación generada	220ms	330ms	0%	100%

Tabla 35: Latencias y Rendimiento del Sistema en Tiempo Real (WebSockets)

Otro resultado clave fue la sistematización de liquidaciones para camarán entero y cola. Anteriormente, este proceso tomaba entre 30 y 45 minutos debido a la lectura manual de registros en cuadernos. El supervisor tenía que revisar cada ingreso y después realizar los cálculos necesarios para después registrar a papel una Liquidación que mostraría al proveedor y gerencia. Con el sistema, las liquidaciones se generan de forma automática a partir de los ingresos de túnel —y pueden actualizarse constantemente cada que el supervisor ingrese un nuevo coche—, reduciendo el tiempo total a segundos.

Durante las iteraciones evaluadas, el tiempo necesario para generar liquidaciones disminuyó en un 80%, manteniendo siempre una precisión del 100%. La posibilidad de exportar la información en formato PDF facilitó la creación de reportes con gráficos de los procesos de interés, que resultaron útiles para gerencia. Cada dato registrado aporta valor al proceso, y la aplicación ofrece toda la información de manera organizada, accesible y fácil de entender.

Proceso	Tiempo Antes	Tiempo Después	Mejora
Registro de Lote	15 minutos	2 minutos	0,86
Registro de Defectos	20 minutos	3 minutos	0,85
Clasificación de Ingreso	25 minutos	5 minutos	0,8
Liquidaciones	45 minutos	5 minutos	0,88
Trazabilidad Completa	No posible	3 segundos	—

Tabla 36: Impacto en Tiempos Operativos Antes vs. Después del Sistema

En el módulo de Reportes y Estadísticas, la consulta de información por lote presentó un desempeño altamente eficiente. El sistema permitió cargar sin dificultad los lotes registrados durante los periodos de prueba y generó gráficos básicos descriptivos —mediante *Chart.js*— relacionados con rendimiento por proceso, pérdidas, producción, control de defectos. Los datos necesarios para estos gráficos se procesaron en menos de dos segundos, incluso con una gran cantidad de lotes consultados al mismo tiempo. Asimismo, las consultas agregadas mostraron una precisión del 100% al relacionar los *lote_id* correspondientes, lo que confirmó la correcta integridad de los datos.

Las pruebas de usabilidad sobre la interfaz, con su estilo *glassmorphism*, recibieron una alta aprobación por parte de los usuarios de prueba. Se destacó la claridad de los formularios, la fluidez en la navegación entre pestañas y lo fácil que fue registrar la información.

Las transacciones de prueba fueron procesadas sin pérdidas de datos, demostrando la robustez del sistema para su implementación en PROMAROSA, lo que demuestra que la solución es más que sólida, es aceptable, eficiente y cumple de forma básica lo necesario para cubrir los procesos de producción de la empresa.

CONCLUSIONES

El análisis detallado de los procesos actuales permitió identificar que las principales demoras se originaban en la duplicidad de registros, el uso de cuadernos y el escaso acceso a información actualizada. Este diagnóstico reveló ineficiencias críticas que afectaban la coordinación entre áreas. A partir de ello, se definieron mejoras centradas en la digitalización, automatización y estandarización del flujo de trabajo.

El modelo de base de datos relacional *PROMAROSA_BD* permitió organizar de toda la información de los procesos de forma ordenada, garantizando que los datos sean coherentes entre los diferentes módulos y mantuviera un orden correcto. La normalización aplicada ayudó a reducir la redundancia y optimizó el almacenamiento de los datos, lo que permitió una gestión más eficiente de estos.

La API RESTful, diseñada con soporte para sincronización en tiempo real, ofreció tiempos de respuesta rápidos y un acceso ágil a la información requerida para generar liquidaciones —que es uno de los elementos prioritarios del sistema—. Gracias a su diseño y a la estructura modular de la API, se pudo organizar las taras de una manera más ordenada para los controladores desarrollados. El uso de WebSockets garantizó que la comunicación sea continua y sin interrupciones, convirtiendo a la API en el núcleo tecnológico que sostiene y mantiene funcionando todo el sistema.

Por su parte, la aplicación móvil simplificó el registro los procesos de producción, eliminando el retraso entre la captura de información y su disponibilidad inmediata para otras áreas. La integración de los módulos de Recepción, Calidad, Clasificación y Masterizado redujo el tiempo de espera, pasando de minutos o incluso horas a solo segundos. Además, su diseño intuitivo permitió a los supervisores registrar datos con precisión, impulsando así la digitalización dentro de PROMAROSA.

La implementación del sistema integró las etapas más importantes del proceso de producción en un entorno digital sincronizado en tiempo real, superando las limitaciones inherentes a los registros manuales. El uso de WebSockets y

herramientas Open Source redujo significativamente la latencia en la actualización de datos, permitiendo generar reportes en cuestión de minutos.

La disponibilidad inmediata de la información mejoró la supervisión y la toma de decisiones, esto porque el sistema ahorra tiempo y los supervisores pueden enfocarse mejor en lo que sucede en la empresa, reforzando el seguimiento de los lotes durante todo su ciclo operativo. En conjunto, el sistema incrementó la eficiencia y la trazabilidad, cumpliendo plenamente con el propósito general establecido.

RECOMENDACIONES

PROMAROSA debería institucionalizar revisiones semestrales de los procesos operativos, con el fin de detectar posibles nuevas causas de demora o inconsistencias. Se recomienda también elaborar manuales internos que documenten los flujos digitalizados y eviten la reincidencia de prácticas manuales.

Es fundamental implementar respaldos automáticos de la base de datos y contar con planes de recuperación de datos ante cualquier fallo. Esto con el fin de asegurar la continuidad del servicio mientras se realiza un seguimiento del crecimiento acelerado de la información almacenada y realizar optimizaciones cuando sea necesario.

Es crucial mantener la documentación técnica de la API actualizada, esto con ayuda de herramientas como Swagger que permite agilizar futuras mejoras y ajustes. La ejecución de pruebas de carga de forma periódica ayudará a prevenir problemas de rendimiento, y ampliar la API permitirá su integración con sistemas utilizados por proveedores o clientes. Como mejora futura, se recomienda implementar Prisma para simplificar actualizaciones de la base de datos evitando la modificación directa.

Se recomienda que la aplicación pueda funcionar sin conexión a dentro de la empresa, lo que mejoraría aún más la eficiencia en áreas de conectividad limitada. Además, es necesario monitorear la experiencia del usuario cada cierto tiempo para ajustar la aplicación según las necesidades reales de éste. Mientras que el módulo de reportes podría incluir paneles con indicadores clave para facilitar el análisis comparativo, así como funciones de predicción relacionadas con los resultados de clasificación y empaque.

Finalmente, es fundamental mantener un proceso de actualización del sistema, integrando mejoras basadas en el uso diario y sugerencias de los usuarios. De igual manera, planificar la integración de otras áreas de la empresa que se excluyeron en la realización de este proyecto permitiendo tener a toda PROMAROSA unida bajo una única aplicación. Por último, realizar auditorías periódicas asegurará que el sistema se mantenga eficiente y sostenible a largo plazo.

REFERENCIAS

- [1] W. C. E., «Desarrollo de una aplicación bajo entorno web que permita llevar el inventario y planificación de la producción en el área empaedora de la Empresa NIRSA de Posorja,» 2017. [En línea]. Available: <https://repositorio.upse.edu.ec/bitstream/46000/4102/1/UPSE-TIN-2017-0012.pdf>. [Último acceso: 2025].
- [2] L. H. G., «Herramientas de software libre para el trabajo científico colaborativo,» 2011. [En línea]. Available: https://gredos.usal.es/bitstream/handle/10366/116282/TFM_LorenaHern%C3%A1ndezG%C3%B3mez.pdf?sequence=1&isAllowed=y. [Último acceso: 2025].
- [3] Corponet, «Importancia de la tecnología en las empresas,» [En línea]. Available: <https://blog.corponet.com/importancia-de-la-tecnologia-en-las-empresas-en-crecimiento>. [Último acceso: 2025].
- [4] DATATEC, «Distintas herramientas tecnológicas al servicio de la empresa,» Febrero 2025. [En línea]. Available: <https://www.datadec.es/blog/herramientas-tecnologicas-servicio-de-empresa>. [Último acceso: 2025].
- [5] L. J. Jurado, «Ventajas competitivas al implementar software en el control de producción y trazabilidad del camarón,» 2021. [En línea]. Available: <https://issuu.com/revista-cna/docs/edicion139/s/11787668>. [Último acceso: 2025].
- [6] J. P. C., «Análisis, Desarrollo e Implementación del Software de Aplicación "AUTODEALLER SOFT" para gestión comercial, inventarios, facturación, contabilidad y gestión empresarial para la empresa AUTO DEALLER Sur,» Enero 2010. [En línea]. Available:

<https://dspace.ups.edu.ec/bitstream/123456789/4419/6/UPS%20-%20ST000408.pdf>. [Último acceso: 2025].

- [7] PROMAROSA, «Productos del Mar Santa Rosa,» [En línea]. Available: <https://promarosa.com/>. [Último acceso: 2025].
- [8] O. M. C. A. Elsa Peralta, «Implementación de un Sistema de Control en una Empacadora de Camarón para aumentar la Eficiencia en una Línea de Producción,» 2013. [En línea]. Available: https://dspace.espol.edu.ec/bitstream/123456789/24600/1/FIMCP_Implementaciom%20de%20un%20sistema%20de%20control%20en%20una%20empacadora%20de%20camaron_FIMCP.pdf. [Último acceso: 2025].
- [9] B. D. R. C. D. Matos, «Desarrollo de una aplicación web para empresa de alimentación,» 2023. [En línea]. Available: <https://docta.ucm.es/rest/api/core/bitstreams/1849ff91-6ea3-4a69-adab-2b82c5e5c5ff/content>. [Último acceso: 2025].
- [10] K. E. O. & Yinka-Banjo, «WebSocket in Real Time Application,» Octubre 2019. [En línea]. Available: <https://www.ajol.info/index.php/njt/article/view/191780>. [Último acceso: 2025].
- [11] I. D. Help, «WebSocket,» [En línea]. Available: <https://www.ibm.com/docs/es/was/9.0.5?topic=applications-websocket>. [Último acceso: 2025].
- [12] Publicomp, «InforFish,» [En línea]. Available: <https://infor-business.com/>. [Último acceso: 2025].
- [13] P. M. Hernández, A. S. González y A. M. Carrillo, «Impacto de la tecnología digital en el sector empresarial en época de crisis,» 10 2023. [En línea]. Available:

<https://portal.amelica.org/ameli/journal/522/5224620001/5224620001.pdf>
.[Último acceso: 2025].

- [14] S. N. d. Planificación, «Plan Nacional de Desarrollo para el nuevo Ecuador 2024-2025,» 2024. [En línea]. Available: <https://www.planificacion.gob.ec/wp-content/uploads/2024/08/RESUMEN-PND-ES.pdf>. [Último acceso: 2025].
- [15] E. H. Inacio, Método de investigación, Creative Commons, 2019.
- [16] C. R. Galarza, Los alcances de una investigación, CienciAmérica, 2020.
- [17] ATLAS, «El muestreo intencional en la investigación cualitativa,» [En línea]. Available: <https://atlasti.com/es/research-hub/muestreo-intencional#:~:text=El%20muestreo%20intencional%2C%20tambi%C3%A9n%20conocido,de%20los%20objetivos%20del%20estudio..> [Último acceso: 2025].
- [18] C. M. Álvarez, «Metodología de la investigación Cuantitativa y Cualitativa,» 2011. [En línea]. Available: <https://www.uv.mx/rmipe/files/2017/02/guia-didactica-metodologia-de-la-investigacion.pdf>. [Último acceso: 2025].
- [19] J. L. Abreu, «El Método de la Investigación,» 2014. [En línea]. Available: <http://www.spentamexico.org/v9-n3/A17.9%283%29195-204.pdf>. [Último acceso: 2025].
- [20] A. Pérez, «Características y fases del modelo incremental,» 16 Agosto 2016. [En línea]. Available: <https://www.obsbusiness.school/blog/caracteristicas-y-fases-del-modelo-incremental>. [Último acceso: 2025].

- [21] «Constitución de la República del Ecuador,» [En línea]. Available: https://www.defensa.gob.ec/wp-content/uploads/downloads/2021/02/Constitucion-de-la-Republica-del-Ecuador_act_ene-2021.pdf. [Último acceso: 2025].
- [22] «Constitución de la República del Ecuador,» [En línea]. Available: <https://educacion.gob.ec/wp-content/uploads/downloads/2012/08/Constitucion.pdf>. [Último acceso: 2025].
- [23] «Ley Orgánica de Protección de Datos Personales,» [En línea]. Available: https://www.finanzaspopulares.gob.ec/wp-content/uploads/2021/07/ley_organica_de_proteccion_de_datos_personales.pdf. [Último acceso: 2025].
- [24] CleverControl, «Cómo implantar CleverControl de conformidad con el GDPR,» [En línea]. Available: <https://clevercontrol.com/es/gdpr/>. [Último acceso: 2025].
- [25] «Reglamento del Parlamento Europeo y del Consejo,» [En línea]. Available: <https://www.boe.es/doue/2016/119/L00001-00088.pdf>. [Último acceso: 2025].
- [26] «Código Orgánico Integral Penal, COIP,» [En línea]. Available: https://www.defensa.gob.ec/wp-content/uploads/downloads/2021/03/COIP_act_feb-2021.pdf. [Último acceso: 2025].
- [27] G. Digital, «Camarón ecuatoriano: los desafíos del sector que define la economía no petrolera,» 2025. [En línea]. Available: <https://www.primicias.ec/revistagestion/analisis/camaron-ecuatoriano-desafios-sector-economia-96353/>. [Último acceso: 2025].

- [28] «América Latina muestra el camino a seguir para la producción sostenible de camarones,» 2016. [En línea]. Available: https://wwf.panda.org/wwf_news/?285371/Amrica-Latina-muestra-el-camino-a-seguir-para-la-produccion-sostenible-de-camarones. [Último acceso: 2025].
- [29] imarc, «Shrimp Processing Plant Project Report 2025: Industry Trends, Plant Setup, Machinery, Raw Materials, Investment Opportunities, Cost and Revenue,» [En línea]. Available: <https://www.imarcgroup.com/shrimp-processing-plant-project-report>. [Último acceso: 2025].
- [30] K. Gallegos, A. Prado y S. Solorzano, «La Viabilidad de las Exportaciones de Camarón con Valor Agregado al Mercado Europeo,» Ciencia Latina, [En línea]. Available: <https://dialnet.unirioja.es/descarga/articulo/10077223.pdf>. [Último acceso: 2025].
- [31] R. García, «Procesamiento de Camarón,» 2020. [En línea]. Available: <https://bloque10.unimagdalena.edu.co/wp-content/uploads/2021/03/PresentacionCAMARON-INDUSTRIA-2020-Profesor-Rafael-Garcia-UNIMAGDALENA-1.pdf>. [Último acceso: 2025].
- [32] E. Fund, «Guía para el procesamiento de camarón,» [En línea]. Available: https://www.ecobusiness.fund/fileadmin/user_upload/Sustainability_Academy/Recursos/Guia_para_el_procesamiento_de_camaron_con_resumen.pdf. [Último acceso: 2025].
- [33] E. T. & P. George, «La Calidad del Camarón congelado,» 1970. [En línea]. Available:

<https://openknowledge.fao.org/server/api/core/bitstreams/182770a5-271e-4f7d-975f-30511a276f35/content>. [Último acceso: 2025].

- [34] ExportQuilsa, «Procesos en ExportQuilsa,» 2022. [En línea]. Available: <https://exportquilsa.com/es/procesos/>. [Último acceso: 2025].
- [35] NIRSA, «Planta empacadora,» [En línea]. Available: <https://nirsa.com/planta-empacadora-de-camaron/>. [Último acceso: 2025].
- [36] N. I. d. Alimentos, «Codex Alimentarius,» Organización Mundial de la Salud, [En línea]. Available: <https://www.fao.org/fao-who-codexalimentarius/es/>. [Último acceso: 2025].
- [37] M. Ian Goulding, «Manual de sistemas de trazabilidad del Pescado y Productos Pesqueros,» Instituto Interamericano para la cooperación en Agricultura, 2016. [En línea]. Available: <https://repositorio.iica.int/server/api/core/bitstreams/9efc4567-7786-4eaa-b9b6-bd83c646e35b/content>. [Último acceso: 2025].
- [38] GoDaddy, «¿Qué es una app? Guía sobre aplicaciones móviles,» 26 Noviembre 2024. [En línea]. Available: <https://www.godaddy.com/resources/es/tecnologia/que-es-una-app-y-para-que-se-utiliza>. [Último acceso: 2025].
- [39] Creatuaplicacion, «Diferencias entre Web App y Mobile App,» [En línea]. Available: <https://creatuaplicacion.com/diferencias-entre-web-app-y-mobile-app/>. [Último acceso: 2024].
- [40] R. Hat, «¿Qué es el open source?,» 24 Enero 2023. [En línea]. Available: <https://www.redhat.com/es/topics/open-source/what-is-open-source>. [Último acceso: 2025].

- [41] J. Mas, «Software libre en el sector público,» [En línea]. Available: <https://www.uoc.edu/dt/20327/20327.pdf>. [Último acceso: 2025].
- [42] I. D. 360, «Open Source: Qué es, beneficios, tendencias y casos de éxito,» Marzo 2024. [En línea]. Available: <https://www.innovaciondigital360.com/cio/open-source-que-es-beneficios-tendencias-y-casos-de-exito/>. [Último acceso: 2025].
- [43] Amazon, «¿En qué consiste el código abierto?,» [En línea]. Available: <https://aws.amazon.com/es/what-is/open-source/>. [Último acceso: 2025].
- [44] F. G. d. Zúñiga, «¿Qué es Visual Studio Code y cuáles son sus ventajas?,» 10 2024. [En línea]. Available: <https://www.arsys.es/blog/que-es-visual-studio-code-y-cuales-son-sus-ventajas>. [Último acceso: 2025].
- [45] GoDaddy, «Visual Studio Code: Funcionalidades y beneficios para desarrolladores,» 2024. [En línea]. Available: <https://www.godaddy.com/resources/latam/desarrollo/visual-studio-code-editor>. [Último acceso: 2025].
- [46] O. Foundation, «Node.js,» [En línea]. Available: <https://nodejs.org/es> . [Último acceso: 2025].
- [47] SIMFORM, «Node.js Architecture and Best Practices for Node.js Application Development,» Octubre 2022. [En línea]. Available: <https://www.simform.com/blog/node-js-architecture-best-practices/>. [Último acceso: 2025].
- [48] M. W. Docs, «JavaScript,» [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/JavaScript>. [Último acceso: 2025].

- [49] Oracle, «MySQL,» [En línea]. Available: <https://www.mysql.com/products/workbench/>. [Último acceso: 2025].
- [50] M. PHP.net, «Bringing MySQL to the web,» 2003. [En línea]. Available: <https://www.phpmyadmin.net/>. [Último acceso: 2025].
- [51] C. Cloud, «From Code to Product,» [En línea]. Available: <https://www.clever-cloud.com/sovereign-cloud/>. [Último acceso: 2025].
- [52] Hostinger, «Qué es GitHub y cómo empezar a usarlo,» [En línea]. Available: <https://www.hostinger.com/es/tutoriales/que-es-github>. [Último acceso: 2025].
- [53] GitHub, «Acerca de GitHub y Git,» [En línea]. Available: <https://docs.github.com/es/get-started/start-your-journey/about-github-and-git>. [Último acceso: 2025].
- [54] «Render,» [En línea]. Available: <https://render.com/>. [Último acceso: 2025].
- [55] joinsecret.com, «Reseñas de Render,» [En línea]. Available: <https://www.joinsecret.com/es/render/reviews>. [Último acceso: 2025].
- [56] GoDaddy, «¿Qué es XAMPP y cuál es su relevancia en el desarrollo web?,» 25 Junio 2024. [En línea]. Available: <https://www.godaddy.com/resources/es/crearweb/xampp-que-es>. [Último acceso: 2025].
- [57] N. Software, «¿Qué es Xampp?,» 2025. [En línea]. Available: <https://norvicsoftware.com/que-es-xampp/>. [Último acceso: 2025].

- [58] M. J. Gonçalves, «¿Qué es Angular y para qué sirve?», Hiberus blog, [En línea]. Available: <https://www.hiberus.com/crecemos-contigo/que-es-angular-y-para-que-sirve/>.
- [59] Ionic, «IonicFramework», [En línea]. Available: <https://ionicframework.com/>. [Último acceso: 2025].
- [60] A. W. Services, «¿Qué es una API RESTful?», 2024. [En línea]. Available: <https://aws.amazon.com/es/what-is/restful-api/>. [Último acceso: 2025].
- [61] S. Li, «WebSockets: Real-Time Communication for Modern Web Applications», Marzo 2025. [En línea]. Available: <https://wslisam.medium.com/websockets-real-time-communication-for-modern-web-applications-40543609aa1f>. [Último acceso: 2025].
- [62] F. IT, «¿Qué es Postman? ¿Cuáles son sus principales ventajas?», 24 Agosto 2023. [En línea]. Available: <https://formadoresit.es/que-es-postman-cuales-son-sus-principales-ventajas/>. [Último acceso: 2025].
- [63] OpenWebinars, «Qué es Postman y primeros pasos», Junio 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-postman/>. [Último acceso: 2025].
- [64] FACSISTEL, «Líneas de Investigación de Unidades Académicas», Universidad Estatal Península de Santa Elena, 2025. [En línea]. Available: https://incyt.upse.edu.ec/index.php?option=com_sppagebuilder&view=page&id=72&Itemid=272. [Último acceso: 2025].
- [65] S. Concur, «Importancia de la trazabilidad de procesos de negocio», 17 Julio 2023. [En línea]. Available: <https://www.concur.com.mx/blog/article/importancia-de-la-trazabilidad->

de-procesos-de-

negocio#:~:text=La%20trazabilidad%20nos%20permite%20rastrear,procesos%20administrativos%20que%20lo%20acompa%C3%B1an.. [Último acceso: 2025].

- [66] D. A. Camacho, Aplicación de un sistema de trazabilidad en la industria de alimentos de una cadena integrada, Quito, 2009.
- [67] O. Blancarte, «Arquitectura Cliente-Servidor,» 2021. [En línea]. Available: <https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/cliente-servidor>. [Último acceso: 2025].
- [68] F. E. Jerez, «Sistema Cliente/Servidor aplicado a punto de ventas a través de Sockets para la Ferretería San Agustín,» Mayo 2005. [En línea]. Available: <https://repositorio.uta.edu.ec/server/api/core/bitstreams/11434120-471e-4fa5-9cb2-eb0e9bf89e9a/content>. [Último acceso: 2025].
- [69] J. B. & M. Ibáñez, Arquitectura Cliente-Servidor en un sistema para la DGENP, Mexico, 1996.
- [70] I. H. Lopez, «Creación de una aplicación móvil para las ofertas de trabajo utilizando la metodología SWIRL,» 2022. [En línea]. Available: <https://repositorio.utmachala.edu.ec/bitstream/48000/19934/1/TTFIC-2022-IS-DE00060.pdf>. [Último acceso: 2025].
- [71] T. A. & B. Boza, Impacto de la tecnología en la productividad de emprendimientos rurales en el cantón La Troncal, Venezuela, 2024.
- [72] «Proyecto de desarrollo para la optimizacion de procesos de manufactura mediante el uso de tecnologías 4.0 en el área de extrusión en la empresa PLASTIGAMA,» 2022. [En línea]. Available:

<https://dspace.ups.edu.ec/bitstream/123456789/23890/1/UPS-GT004099.pdf>. [Último acceso: 2025].

- [73] Sydle, «Optimización de procesos: ¿Qué es y por qué es tan importante para tu negocio?,» 14 Septiembre 2023. [En línea]. Available: <https://www.sydle.com/es/blog/que-es-optimizacion-de-procesos-6126ac39b060f57604039a57>. [Último acceso: 2025].

- [74] PubNub, «Node.js WebSocket Programming Tutorial with JavaScript,» 2019. [En línea]. Available: <https://www.youtube.com/watch?v=YaJbc7s1ROg>. [Último acceso: 2025].

ANEXOS

Anexo 1: Registro de la técnica de observación aplicada en PROMAROSA

Registro descriptivo de la información	
Objetivo: Identificar y comprender los posibles problemas o fallos que puedan estar ocurriendo en los diferentes procesos de producción.	
Tipo de Observación:	Observación participativa
Fecha:	04 enero - 10 febrero del 2025
Lugar:	Productos del Mar Santa Rosa (PROMAROSA)
# Personas:	7
Proceso:	Recepción y Clasificación
Duración:	12 horas
Hechos observados	
☐ Durante el período observado, se realizó el proceso de recepción y clasificación de los camarones que llegaron a la planta de producción. ☐ La información relacionada al lote recibido (proveedor, chofer, vehículo, datos de origen del camarón) se registró manualmente en hojas de cálculo. Pero, este método es propenso a errores de transcripción debido a la alta carga de trabajo y la naturaleza manual del procedimiento. ☐ Durante el proceso de recepción y clasificación, la comunicación entre operarios no fluía de manera eficiente. Los registros de cada lote no se actualizaban inmediatamente, lo que generaba confusión y descoordinación entre los empleados encargados de la recepción y los responsables de la clasificación. ☐ Se registraron manualmente los datos de cada lote recibido en hojas de cálculo, lo que generó demoras en el proceso de registro y la posibilidad de errores humanos. ☐ Debido a la falta de integración entre la recepción y la clasificación, los lotes esperaban tiempo adicional para ser procesados, lo que resultaba en una acumulación de pedidos sin completar, afectando la capacidad de cumplir con las entregas a tiempo. ☐ Los supervisores no tenían acceso inmediato a los datos de producción, lo que dificultaba la toma de decisiones rápidas en caso de problemas o discrepancias en la calidad del producto recibido. ☐ Seguimiento y gestión de próximos lotes a procesar ☐ Hubo dificultades en la coordinación entre los operativos de recepción y los encargados de clasificación, ya que la información no se actualizaba de manera sincronizada.	
Resumen:	El proceso de recepción y clasificación de camarones en PROMAROSA se realiza de forma manual, lo que genera demoras, errores en los registros y dificultades en la coordinación entre áreas. La falta de trazabilidad en tiempo real y una comunicación deficiente impactan la eficiencia y la capacidad de cumplir con los plazos de entrega. La implementación de una solución tecnológica podría optimizar estos procesos y mejorar la coordinación y precisión.
Responsable:	Anthony Lucas Floreano

Anexo 2: Formato de entrevista realizada a los supervisores del Área de Recepción y Clasificación de PROMAROSA

Entrevista realizada a los supervisores del Área de Recepción y Clasificación	
Objetivo: Conocer cómo se realizan actualmente los procesos de seguimiento y registro de Materia Prima en el proceso de clasificación	
1.	¿Qué métodos o herramientas se utilizan para registrar la información de la producción en la empresa?
2.	¿Cómo se gestionan actualmente las órdenes de compra y la producción en la empresa?
3.	¿Cuáles son los problemas más comunes que surgen al utilizar cuadernos y hojas de cálculo para registrar los datos de producción?
4.	¿Qué tan crítico es para la empresa tener acceso a información actualizada y en tiempo real sobre los procesos de producción?
5.	¿Cuánto tiempo tarda en generar un reporte de un lote procesado?
6.	¿Cómo se lleva a cabo la comunicación entre las áreas que componen el proceso de producción?
7.	¿Cada cuánto tiempo es necesario revisar los registros para asegurarse de que la información sea precisa y confiable?
8.	¿Por qué necesita una aplicación?
9.	¿Qué beneficios considera que podría aportar el uso de una aplicación para optimizar la comunicación y el intercambio de información dentro de la empacadora?
10.	¿Qué tanto conocimiento o experiencia tienen los empleados en el uso de tecnologías modernas de software?

Anexo 3: Estructura del proyecto

Carpeta/Sección	Descripción	Archivos/Claves Principales
Frontend (src/app)	Estructura principal de la aplicación móvil híbrida desarrollada con Ionic 7 y Angular. Incluye componentes, servicios y rutas para los módulos.	app.component.ts, app.routes.ts, componentes/menu/menu.component.ts (menú dinámico por rol), modulos/M1 a M6 (e.g., m4-clf-registrar-ingreso-tunel.page.ts/html/scss para clasificación).
Servicios (src/app/servicios)	Servicios para llamadas API y lógica compartida, como HTTP requests y WebSockets.	general.service.ts (URL API, login local), socket.service.ts (conexión WS con reconexión), choferes.service.ts, lotes.service.ts, etc. (CRUD por entidad).
Interfaces (src/app/interfaces)	Definiciones de tipos para datos (e.g., interfaces para personas, menús).	interfaz.ts(IPersona, IMenuItem).
Assets (src/assets)	Recursos estáticos como JSON para datos locales (menú, usuarios).	BD/menu.json (estructura menú por rol), BD/usuarios.json (datos usuarios).
Theme (src/theme)	Estilos globales con variables SCSS para glassmorphism, colores y animaciones.	variables.scss (definiciones --primary-color, etc.).
Backend (root)	Servidor Node.js con Express para API REST y Socket.IO.	app.js (config Express, CORS, rutas), index.js (servidor HTTP + IO), config.js (env vars BD), db.js (pool MySQL).
Controladores (src/controladores)	Lógica de negocio por entidad (e.g., CRUD con queries MySQL).	chofer.js (get/post/put/delete choferes), ingresotunel.js (manejo ingresos túnel, joins).
Rutas (src/routes)	Rutas Express para endpoints API.	chofer.routes.js (app.use('/api/chofer', getChofer)), etc.
WebSockets (src/ws)	Configuración eventos Socket.IO.	websocket.js (io.on('connection'), eventos como 'ingreso_tunel_nuevo').
Base de Datos	Esquema MySQL con tablas relacionales para entidades.	bxtewr68vcyp29pcnz23.sql (CREATE TABLE chofer, lote, orden, etc., con FK para trazabilidad).
Documentación	Archivos README y docs para estructura y flujo.	README.txt (descripción general, arquitectura, módulos, flujo producción), Proyecto UIC V4.docx (tesis completa con fases, diagramas).
Entorno (.env)	Variables ambiente para BD y puertos.	BD_HOST, BD_DATABASE, DB_USER, DB_PASSWORD, PORT (config para Clever Cloud).

Anexo 4: Base de Datos localhost para primera fase de pruebas

localhost/phpmyadmin/index.php?route=/database/structure&db=bxtewr68vcyp29pcnz23

phpMyAdmin

Reciente Favoritas

Nueva bxtewr68vcyp29pcnz23

auditoria

chofer

clase

clasificacion

coche

color

control_calidad

corte

defectos

descabezado

glaseo

grupo

ingresotunel

liquidacion

lote

maquina

masterizado

menu

orden

pelado

personal

peso

presentacion

talla

tipo

usuario

vehiculo

information_schema

mysql

performance_schema

phpmyadmin

promociones_bd

test

Filtros

Que contengan la palabra:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
auditoria	Examinar Estructura Buscar Insertar Vaciar Eliminar	8	InnoDB	utf8mb4_general_ci	32.0 KB	-
chofer	Examinar Estructura Buscar Insertar Vaciar Eliminar	19	InnoDB	utf8mb4_general_ci	89.0 KB	-
clase	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_general_ci	16.0 KB	-
clasificacion	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	268.0 KB	-
coche	Examinar Estructura Buscar Insertar Vaciar Eliminar	60	InnoDB	utf8mb4_general_ci	16.0 KB	-
color	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8mb4_general_ci	16.0 KB	-
control_calidad	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_general_ci	128.0 KB	-
corte	Examinar Estructura Buscar Insertar Vaciar Eliminar	12	InnoDB	utf8mb4_general_ci	32.0 KB	-
defectos	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8mb4_general_ci	48.0 KB	-
descabezado	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	80.0 KB	-
glaseo	Examinar Estructura Buscar Insertar Vaciar Eliminar	9	InnoDB	utf8mb4_general_ci	16.0 KB	-
grupo	Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8mb4_general_ci	16.0 KB	-
ingresotunel	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	288.0 KB	-
liquidacion	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	32.0 KB	-
lote	Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	utf8mb4_general_ci	160.0 KB	-
maquina	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8mb4_general_ci	16.0 KB	-
masterizado	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	64.0 KB	-
menu	Examinar Estructura Buscar Insertar Vaciar Eliminar	17	InnoDB	utf8mb4_general_ci	48.0 KB	-
orden	Examinar Estructura Buscar Insertar Vaciar Eliminar	9	InnoDB	utf8mb4_general_ci	32.0 KB	-
pelado	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	80.0 KB	-
personal	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	32.0 KB	-
peso	Examinar Estructura Buscar Insertar Vaciar Eliminar	8	InnoDB	utf8mb4_general_ci	16.0 KB	-
presentacion	Examinar Estructura Buscar Insertar Vaciar Eliminar	11	InnoDB	utf8mb4_general_ci	16.0 KB	-
proveedor	Examinar Estructura Buscar Insertar Vaciar Eliminar	8	InnoDB	utf8mb4_general_ci	16.0 KB	-
talla	Examinar Estructura Buscar Insertar Vaciar Eliminar	28	InnoDB	utf8mb4_general_ci	32.0 KB	-
tipo	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8mb4_general_ci	16.0 KB	-
usuario	Examinar Estructura Buscar Insertar Vaciar Eliminar	7	InnoDB	utf8mb4_general_ci	32.0 KB	-
vehiculo	Examinar Estructura Buscar Insertar Vaciar Eliminar	10	InnoDB	utf8mb4_general_ci	32.0 KB	-
28 tablas	Número de filas	235	InnoDB	utf8mb4_general_ci	1.6 MB	0 B

Anexo 5: Pruebas en PostMan

GET http://localhost:3000/api/ GET https://tesisbackv1-2.on

http://localhost:3000/api/lote

GET http://localhost:3000/api/lote

Params Authorization Headers (9) Body Pre-request Script Tests

Query Params

Key	Value
Key	

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
{
  "lote_id": 23,
  "parent_lote_id": null,
  "lote_tipo": "entero",
  "tipo_id": 1,
  "clase_id": null,
  "color_id": null,
  "vehiculo_id": 3,
  "chofer_id": 21,
  "lote_codigo": "C-200-2025",
  "lote_fecha_ingreso": "2025-11-18T05:00:00.000Z",
  "lote_hora_ingreso": "01:00:00",
  "proveedor_id": 0,
  "lote_libras_remitidas": "12000.00",
  "lote_peso_promedio": "11990.00",
  "lote_n_piscina": "12",
  "lote_n_bines": 12,
  "lote_n_gavetas_conicas": 10,
  "lote_n_gavetas_caladas": 0,
  "lote_n_sanos_hielos": 10
}
```



FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CARRERA DE TECNOLOGÍA DE LA INFORMACIÓN

Anexo 6: Despliegue de Actualizaciones de la API en Render

The screenshot displays the Render dashboard for a service named 'TesisBackFV1-2'. The left sidebar contains navigation options: Dashboard, TesisBackFV1-2, Events (selected), Settings, MONITOR (Logs, Metrics), and MANAGE (Environment, Shell, Scaling, Previews, Disk, Jobs). The main panel shows the service details, including a 'Connect' button, a 'Manual Deploy' button, and the service ID: 'srv-d4dcqfjpnbc739vmljg'. Below this, a list of deployment events is shown, filtered to 31 events. The events are as follows:

Event Type	Commit Hash	Description	Timestamp	Action
Deploy live	6fc34cc	Corrección de errores m...	December 2, 2025 at 11:21 PM	
Deploy started	6fc34cc	Corrección de errores m... Manually triggered by you via Dashboard	December 2, 2025 at 11:19 PM	
Deploy live	e3ea06c	Corrección ...	December 2, 2025 at 10:54 PM	Rollback
Deploy started	e3ea06c	Corrección de errores m... Manually triggered by you via Dashboard	December 2, 2025 at 10:53 PM	
Deploy live	ee34537	Corrección ...	December 2, 2025 at 10:36 PM	Rollback
Deploy started	ee34537	Corrección de errores m... Manually triggered by you via Dashboard	December 2, 2025 at 10:35 PM	
Deploy live	d6bac7f	Corrección ...	December 2, 2025 at 10:07 PM	Rollback
Deploy started	d6bac7f	Corrección de errores m... Manually triggered by you via Dashboard	December 2, 2025 at 10:06 PM	
Deploy live	eeabed4	Corrección ...	December 2, 2025 at 5:04 PM	Rollback
Deploy started	eeabed4	Corrección de errores m... Manually triggered by you via Dashboard	December 2, 2025 at 5:03 PM	
Deploy canceled	eeabed4	Corrección de errores m...		

MANUAL DE USUARIO

INTRODUCCIÓN GENERAL

El Manual de Usuario describe de forma clara y concisa el funcionamiento del Sistema de Información en Tiempo Real para el Control de Producción de PROMAROSA. El sistema integra todos los procesos productivos del camarón — desde la recepción de la Materia Prima hasta la generación de reportes y estadísticas— con el objetivo de digitalizar, organizar y mejorar la toma de decisiones dentro de la empresa. Cada módulo del sistema representa una fase del proceso real de producción y está diseñado para que el personal pueda registrar datos, consultarlos, analizarlos y monitorear el avance del producto en tiempo real. A continuación, se presenta la descripción general de cada módulo y las opciones disponibles en cada uno.

MÓDULOS EXTRA

Este grupo de módulos contiene pantallas auxiliares necesarias para el funcionamiento interno del sistema, como vistas informativas, espacios temporales y páginas manejadoras de errores. Dentro de este módulo, el usuario puede encontrar espacios donde se muestra información sobre la aplicación, pantallas de transición o estructuras destinadas a manejar errores inesperados.

Opciones del módulo:

- meinfo
- pantallavacia
- errorxd

MÓDULO 1 – CONTROL DE ACCESO

Este módulo administra todo lo relacionado con el acceso al sistema. Su objetivo es garantizar que cada usuario ingrese únicamente con sus credenciales y pueda ver únicamente las opciones permitidas por su rol. Aquí se gestiona la autenticación, la administración de usuarios y la navegación principal. Es el módulo inicial, por lo que define la estructura de permisos y la seguridad del sistema.

Opciones del módulo:

- m1info
- loginxd
- guía
- principal
- usuario
- usuarios

MÓDULO 2 – RECEPCIÓN DE MATERIA PRIMA

Este módulo permite registrar y administrar toda la información relacionada con personas, vehículos, proveedores y lotes que ingresan a la planta. Representa el primer contacto entre la Materia Prima y el sistema digital, por lo que su correcta utilización garantiza la trazabilidad desde el origen. El usuario puede gestionar choferes, vehículos y proveedores, así como registrar cada lote de camarón que ingresa a la empresa, incluyendo datos de peso, bins, piscina, temperatura y observaciones.

Opciones del módulo:

- m2info
- m2-ch-listar-choferes
- m2-ch-listar-detalle-choferes/:id
- m2-ch-registrar-chofer/:id
- m2-vh-listar-vehiculos
- m2-vh-listar-detalle-vehiculos/:id

- m2-vh-registrar-vehiculo/:id
- m2-p-listar-proveedores
- m2-p-listar-detalle-proveedores/:id
- m2-p-registrar-proveedor/:id
- m2-l-listar-lotes
- m2-l-listar-detalle-lotes/:id
- m2-l-registrar-lote/:id

MÓDULO 3 – CONTROL DE CALIDAD

Este módulo se encarga de registrar y evaluar la calidad de la Materia Prima recibida. Los analistas pueden documentar defectos físicos, sensoriales y estructurales del camarón, así como información del muestreo como talla, color, peso bruto,

peso neto y total analizado. Este módulo es fundamental porque determina la calidad inicial del producto y permite evaluar proveedores, identificar problemas recurrentes y prevenir pérdidas futuras.

Opciones del módulo:

- m3info
- m3-df-listar-defectos
- m3-df-listar-detalle-defectos/:id
- m3-df-registrar-defecto/:id
- m3-cc-listar-controlcalidad
- m3-cc-listar-detalle-controlcalidad/:id
- m3-cc-registrar-controlcalidad/:id

MÓDULO 4 – CLASIFICACIÓN Y PROCESAMIENTO

Este módulo cubre las actividades intermedias de los procesos de producción, incluyendo clasificación por talla, color, peso y corte; procesamiento; y posteriormente el ingreso del producto a túnel de congelación. El usuario puede registrar cada ingreso a túnel, diferenciando entre producto entero

o cola, y visualizar listados completos con agrupaciones, totales y detalles individuales.

También permite gestionar las liquidaciones del lote, donde se calcula rendimiento, basura, sobrante y total empacado, lo cual es crítico para medir eficiencia del proceso.

Opciones del módulo:

- m4info
- m4-clf-listadogeneral-ingreso-tunel
- m4-clf-tipo-ingreso/:id
- m4-clf-listadogeneral-ingreso-tunel-entero/:id
- m4-clf-listadogeneral-ingreso-tunel-cola/:id
- m4-clf-listar-detalle-ingreso-tunel/:id
- m4-clf-registrar-ingreso-tunel
- m4-clf-registrar-ingreso-tunel/:id
- m4-lqd-c-listar-liquidacioncola
- m4-lqd-c-listar-detalle-liquidacioncola/:id
- m4-lqd-e-listar-liquidacionentero
- m4-lqd-e-listar-detalle-liquidacionentero/:id

MÓDULO 5 – MASTERIZADO Y EMPAQUE

Este módulo permite registrar el proceso de masterizado, es decir, el agrupamiento y empaque del producto congelado en cajas master. El usuario puede registrar kilos procesados, número de cajas, tipo de master, observaciones y coche utilizado. El módulo facilita comparar el peso final empacado con el peso remitido o clasificado, permitiendo calcular posibles pérdidas o variaciones.

Opciones del módulo:

- m5info

- m5-m-listar-master
- m5-m-listar-detalle-master/:id
- m5-m-registrar-master/:id

MÓDULO 6 – GESTIÓN DE ÓRDENES

Este módulo está orientado a personal de gerencia o supervisión. Permite generar órdenes internas de producción según necesidades de mercado, disponibilidad de materia prima o planificación diaria. El usuario puede consultar el detalle de cada orden, vincularla a un lote y monitorear su avance según la información registrada en los módulos anteriores.

Opciones del módulo:

- m6info
- m6-op-listar-ordenes
- m6-op-listar-detalles-ordenes/:id
- m6-op-registrar-orden/:id

MÓDULO 7 – TRAZABILIDAD + REPORTE Y ESTADÍSTICAS

Este módulo constituye una de las partes más importantes del sistema. Permite reconstruir de forma automatizada toda la vida del lote: desde la recepción hasta el empaque final.

El usuario puede consultar cada etapa del proceso, visualizar tablas, datos, valores acumulados y generar un documento PDF con toda la trazabilidad completa.

Además, el módulo incorpora herramientas de Reportes y Estadísticas, permitiendo generar gráficos comparativos, análisis de pérdidas, rendimiento general, ranking de proveedores y comportamiento de la producción.

Opciones del módulo:

- m7-info
- m7-t-historial-lote
- m7-t-listar-historial-lote/:id

- m7-t-listar-detalle-historial-lote/:id
- m7-t-tipo-ingreso
- m7-t-listado-general-historial
- m7-r-listar-reporte
- m7-r-listar-detalle-reporte/:tipo

MANUAL TÉCNICO DEL SISTEMA

1. INTRODUCCIÓN GENERAL

El presente Manual Técnico describe en detalle la arquitectura, funcionamiento interno, tecnologías, procesos de instalación, estructura del backend y del frontend, así como las instrucciones necesarias para la configuración, actualización y mantenimiento del Sistema de Información en Tiempo Real para el Control de Producción de PROMAROSA. Este sistema constituye una solución integral que articula siete módulos funcionales, permitiendo la digitalización completa del flujo productivo del camarón, desde la llegada del lote a la planta hasta su empaque final.

El manual se encuentra dirigido a administradores, desarrolladores y personal técnico responsable de operar, mantener o ampliar la plataforma. Se detalla cada componente tecnológico utilizado —incluyendo Node.js, Express, Ionic, Angular, MySQL y WebSockets— explicando el rol de estos elementos en el sistema y su interacción con la infraestructura general. También se presentan lineamientos para el despliegue en servidores externos, respaldos, monitoreo y diagnóstico de fallas.

2. ARQUITECTURA DEL SISTEMA

El sistema adopta una arquitectura Cliente–Servidor moderna basada en REST y WebSockets. Esta arquitectura permite una comunicación fluida entre la aplicación móvil/desktop (Ionic + Angular) y el servidor backend (Node.js + Express), con capacidad para transmisión en tiempo real gracias al canal WebSocket.

La arquitectura se divide en tres capas principales:

1. **Frontend (Cliente):** desarrollado en Ionic + Angular, responsable de la interacción con el usuario y de la presentación de la información.
2. **Backend (Servidor API):** construido con Node.js + Express, administra la lógica de negocio, seguridad, validaciones y gestión de datos.
3. **Base de Datos:** gestionada con MySQL, almacena toda la información persistente de la planta, incluyendo usuarios, proveedores, lotes, clasificaciones, controles de calidad, liquidaciones y más.

La comunicación entre estas capas se establece de la siguiente forma:

- Las solicitudes HTTP (GET, POST, PUT, DELETE) se utilizan para operaciones regulares como registros, consultas y actualizaciones.
- Los eventos WebSocket permiten transmitir información inmediata a ciertos módulos, especialmente aquellos que requieren actualización en tiempo real (ej.: ingreso a túnel, recepción de lotes, estados de órdenes).
- La base de datos es accedida por el backend mediante un pool de conexiones que garantiza eficiencia y manejo adecuado de múltiples usuarios concurrentes.

Esta arquitectura asegura escalabilidad, modularidad y un bajo acoplamiento entre componentes, facilitando futuras ampliaciones del sistema.

3. TECNOLOGÍAS UTILIZADAS

El sistema utiliza un conjunto de tecnologías modernas y ampliamente soportadas:

Frontend

- **Ionic 7** para diseño híbrido móvil/web.
- **Angular 17** como Framework estructural para componentes, navegación y servicios.



FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CARRERA DE TECNOLOGÍA DE LA INFORMACIÓN

- **TypeScript** para tipado estático.
- **HTML5** y **SCSS** para plantillas y estilos.

Backend

- **Node.js 18+**
- **Express.js**
- **JWT (Json Web Token)** para autenticación.
- **bcrypt.js** para encriptado de contraseñas.
- **CORS, helmet** para seguridad.
- **MySQL2** para conexión a base de datos.

Comunicación en tiempo real

- **WebSockets (Socket.IO)** para transmisión instantánea de información productiva.

Base de datos

- **MySQL**, con tablas normalizadas, integridad referencial y relaciones maestro–detalle.

Despliegue

- **Render.com** para hosting del backend.
- **Capacitor** o **Ionic Deploy** para construir la app en Android.

4. ESTRUCTURA DEL BACKEND

El servidor backend está organizado en carpetas, siguiendo una arquitectura limpia basada en controladores, rutas y modelos. La estructura general es:

```
/src
  /controladores
    choferCtrl.js
    vehiculoCtrl.js
    proveedorCtrl.js
    loteCtrl.js
    calidadCtrl.js
    clasificacionCtrl.js
    liquidacionCtrl.js
    masterizadoCtrl.js
    ordenCtrl.js
    trazabilidadCtrl.js
    reportesCtrl.js
  /routes
    chofer.routes.js
    vehiculo.routes.js
    proveedor.routes.js
    lote.routes.js
    calidad.routes.js
    clasificacion.routes.js
    liquidacion.routes.js
    masterizado.routes.js
    orden.routes.js
    trazabilidad.routes.js
    reportes.routes.js
  /db.js
  /middleware
    auth.js
  /helpers
    utilidades.js
  app.js
```

Cada controlador contiene la lógica de negocio del módulo correspondiente, mientras que las rutas definen las URLs expuestas al frontend.

5. BASE DE DATOS

El sistema utiliza un modelo relacional que representa fielmente el flujo de trabajo de PROMAROSA. Incluye tablas como:

- **usuario**



FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CARRERA DE TECNOLOGÍA DE LA INFORMACIÓN

- **rol**
- **proveedor**
- **vehiculo**
- **chofer**
- **lote**
- **control_calidad**
- **defectos**
- **clasificacion**
- **ingreso_tunel**
- **liquidacion_entero**
- **liquidacion_cola**
- **masterizado**
- **orden_produccion**

Cada tabla está indexada para acelerar consultas, con claves foráneas que aseguran la integridad del proceso de trazabilidad.